IMAGE CHARACTERISATION OF MARINE EXPLOSIVE CYCLOGENESIS

by

Barry Pierce

B.E.S., The University of Waterloo, 1993

THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

ENVIRONMENTAL SCIENCE

THE UNIVERSITY OF NORTHERN BRITISH COLUMBIA

August 2002

0-612-80676-6

APPROVAL

Name: Barry Pierce

Degree: Master of Science

Thesis Title: IMAGE CHARACTERISATION OF MARINE EXPLOSIVE CYCLOGENESIS

Examining Committee:

Chair: Dr. Robert W. Tait
Dean of Graduate Studies
UNBC

Supervisor: Dr. Peter Jackson
Associate Professor, Environmental Studies Program
UNBC

R.D. Wheate

Committee Member: Dr. Roger Wheate
Associate Professor, Geography Program
UNBC

Committee Member: Dr. Josef Ackerman
Associate Professor, Environmental Studies Program
UNBC

External Examiner: Dr. Phil Austin
Associate Professor, Earth & Ocean Science
Atmospheric Science Program
University of British Columbia

Date Approved: August 31, 2002

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Image Characterisation of Marine Explosive Cyclogenesis

Original work was undertaken to define and explore the theoretical concept of *Image Characterisation* as an extension of the remote sensing concept of *signatures* of physical materials. *Character* represents a higher–level synthesis of the overall meaning of the image scene or sequence, rather than a description of constituent elements. An applied problem in meteorology was addressed by exploring the character of marine explosive cyclogenesis from satellite image sequences. The work followed methods developed by Roger Weldon and by the Meteorological Service of Canada in their *PROBE* operational procedure. Image enhancements and methods for measuring both radiometric and geometric changes in the imagery were developed, and applied to five cases. Statistical treatment of the results of these methods, focused around a four–phase physical model of development, revealed promising correlations. The length of the Phase II to Phase IV interval correlated strongly with both the 24 hour Bergeron value and the minimum recoded case pressure ($R^2 = 0.79$, $p = 0.04$ and $R^2 = 0.87$, $p = 0.07$ respectively). Three significant findings related the difference in variables as measured at the Phase III and Phase II intervals: storm bearing and jet tilt from imagery predicted the 24 hour Bergeron value ($R^2 = 0.92$, $p = 0.01$ and $R^2 = 0.86$, $p = 0.07$ respectively); and the maximum image gradient strength of the storm edge predicted minimum case pressure ($R^2 = 0.77$, $p = 0.05$). The results are especially promising given that these measures could be derived on intervals occuring 20 to 30 hours prior to maximum storm strength. Other results suggest that the 24 hour interval used to measure storm intensity, currently part of the standard definition, may be too long to capture the essential character of these systems. These

findings partially validated the concept of character, while the methods developed may lead to more effective and cost efficient predictions.

1. INTRODUCTION

1.1 Description and hazard

Computer analysis of environmental phenomena has opened a wide range of potential methods for description and prediction of environmental hazard. In the field of meteorology, a persistent and important problem is the need to provide timely and accurate prediction of high–impact storms. While some powerful systems, such as hurricanes, develop over an extended period of time and can be easily tracked, there exists a class of winter cyclones, colloquially referred to as *Marine Bombs* [39] which deepen explosively and which has historically confounded numerical prediction methods [39], [4], [13], [34].

Detailed numerical models are hindered by the sparseness of available meteorological data over the deep ocean waters where Bombs typically develop, and by the unique dynamics that drive the explosive development of these cyclones (see Chapter 2). The problem of prediction, while difficult, is nonetheless very significant. Yearly economic benefit from hazard avoidance derived from first and second day meteorological predictions has been estimated to be 2 and 2.8 billion dollars Canadian respectively [28]. In cases of explosive development, hazard is magnified, since coupled with their quick development, Bombs typically become strong and long lived cyclonic systems [14]. At their peak of development, they can span several thousand kilometres horizontally. They move farther and travel faster than other systems [38], and can have an intense low pressure core of over 200 kilometres in diameter [31].

The most often cited technical definition for explosive cyclogenesis is a drop in central pressure of at least 24 hectopascals (hPa) over a 24 hour period [39]. While a deepening of 24 hPa describes a fairly strong storm, it should be noted that this is the **minimum** requirement for explosive development. Many *Bombs* continue to develop much deeper than a drop of 24 hPa, sometimes with total pressure falls greater than 60 hPa. In order to illustrate the nature of a *Marine Bomb*, a term colloquially adopted by Sanders and Gyakum [39], the characteristics of some well known storms will be described.

For example, the notorious Queen Elizabeth II storm of 1978 saw a deepening of 60 hPa/24 h along the American eastern seaboard resulting in the battering of the *Queen Elizabeth II* and loss of the dragger *Captain Cosmo* [12], while the often studied President's Day storm of 1979 resulted in record–breaking snowfalls in the mideastern United States [4]. Another cyclone analysed by Neiman and Shapiro [31] similarly deepened 60 hPa/24 h, and was approximately 5000 km across, with a 250 km wide comma shaped cloud region (comma head) around its eye. It was found to be one of the most intense cyclonic systems ever recorded [31]. Surface winds reaching 45 m/s were accompanied by 10 m ocean swells. Put on a human scale, one can imagine a series of three–story tall swells of water rushing at us backed by over 160 km/h winds – a truly frightening prospect. By way of comparison, circumstance changed hazard into tragedy when a significantly weaker storm led to the sinking of the *Ocean Ranger* off Newfoundland and the destruction of the Soviet *Mekhanik Tarasov*, resulting in the loss of 116 lives [38].

1.2 Operational problems

Early warning, even as soon as twelve hours prior to maximum intensification, would help ships reach port or prepare for an incipient storm. Currently, a potentially dangerous situation exists where real-time meteorological data are unavailable to initialise numerical models. Data made available to weather forecast centres hours after maximum intensification can only be useful in a post hoc analysis, and cannot help mariners prepare for or avoid the hazards of explosive development (cf. [4]). The situation remains the same today, as suggested by Pacific Weather Centre (PWC) staff (Personal communication with Laurie Neil and Neil McLennan, Pacific Weather Centre, Meteorological Service of Canada, 1998. See Section 3.1).

Further, the irregular nature of transient ship reports makes accurate predictions unlikely, even where the data are available in good time. Several authors have noted poor data availability as a severe limit on predictive model performance. Sanders and Gyakum [4] noted sparse data over the Pacific, while Sanders [38] found some events in an Atlantic study were completely missed due to data sparsity. Kuo and Low-Nam [20] suggested better data coverage would improve model performance. Kuo and Low-Nam [20] and Kuo et al. [21] also indicate that Bombs are more sensitive to initial conditions than other mesoscale systems: an indication that missing data can be expected to have a marked impact on predictive model performance. Neiman et al. [21] identify a need for sensors that provide high spatial and temporal resolution. They further suggested that remote sensing systems are the most likely operational system to meet such requirements.

1.3 Images and processing

The goal of this work is to investigate the uses of remote sensing as an alternative and complementary data source for prediction and description of Bombs. Some basic principles Remote Sensing follow; these concepts are common knowledge within the field of Remote Sensing, but are offered here to provide background information relevant to the work presented here. For a more detailed discussion, cf. [23], [17].

Every year new earth resource satellites are proposed or launched, resulting in a global network of commercial systems that is quite large. Fundamentally, there are two types of remote sensing satellite systems: near polar orbiters, and geostationary orbiters. The former have a ground track slightly oblique to the poles of the Earth to avoid the influence of its magnetic field. They provide coverage over all points within a few degrees away from the poles from once to a few times a day depending on orbital characteristics. The rotation of the Earth below the satellite allows complete coverage, providing at nadir (straight down) viewing for any point.

The latter (geostationary satellites) orbit around the equator at a rate that is calculated to keep them above the same point on Earth at all times. As with polar orbiters, the field of view depends on the optics involved in specific systems and the altitude at which the platform orbits, although geostationary orbits require a much greater elevation, usually allowing for a view of the complete disc of the Earth. The main advantage of geostationary systems is that they provide continuous coverage of the same frame of view. They share with polar orbiters the property that objects at the extremes of the frame will be obliquely sensed and will therefore appear stretched; however, unlike polar orbiters, the nadir point on geostationary systems is fixed, so there is no way for these platforms to provide true geometry for other points on Earth in subsequent frames. For this reason, geostationary orbiters provide poor geometric coverage of arctic regions. Geometric post–corrections can be used with some success to restore proper geometry to regions within the visible frame.

Both polar and geostationary platforms can carry a variety of sensors. Most systems use either a rotating mirror to focus light across the entire swath of the scene below onto a photoelectric cell (eg. Landsat), or have optics that simultaneously focus the entire swath onto an array of cells called a charge–coupled device (eg. SPOT). Polar orbiters can take advantage of the motion of the platform relative to the Earth to advance the swath band: a technique called "push–broom" scanning. The Geostationary Operational Environmental Satellite (GOES) series, one of the most commonly used geostationary satellites, scans the image swath by rotation of the entire satellite parallel to the Earth's equator. Subsequent scan lines are filled by tilting an internal mirror in the optical path.

Most common satellite sensors, with the exception of radar, passively record reflected or emitted radiation. The "bands" of a sensor are determined by which wavelengths of radiation each sensor element records. A digital intensity level is averaged for each band over a spatial range (the scan resolution), resulting in a series of discrete points within the swath, then swaths are combined to produce a full frame. The choice of bands depends on the intended uses of the digital product. Each band may reveal the same scene in differing ways depending on the energy interactions occurring with the object being sensed and with any interfering factors, such as atmospheric scattering and absorption.

Meteorologists rely heavily on geostationary scenes, recording in the visible and thermal infrared spectra. Meteorological satellites are designed with relatively coarse spatial resolution, since the phenomena they are designed to sense are large with respect to the image frame. Coarse resolution allows the amount of data needed to fill a frame to be significantly smaller, allowing a short turn–over period for transmission of each scene. The GOES series, for example, provides a full scene each half hour: a significant advantage for tracking developing weather systems. The GOES system was chosen as the data platform for the current study.

Visible band images have a higher spatial resolution than thermal because the intensity of reflected visible light is much higher than that of emitted thermal energy. (See Chapter 3.) The target area being sensed at any given time is known as the instantaneous field of view (IFOV) of the sensor, and corresponds to the final resolution of each picture element (pixel) in the image scene. Higher energy in the visible band allows the sensor to be saturated when averaging over a smaller target area, resulting in a smaller IFOV. For the older GOES–7 satellite series, the difference was between 0.78 kilometre pixels per pixel side for visible and 7 kilometres per pixel side for thermal bands. In addition, the visible band was quantised to 8 bits versus 6 for thermal: a difference of $2^8=256$ and $2^6=64$ discrete digital levels. (Quantization is the process of assigning a discreet integer value to digitally represent the signal amplitude. cf. [11], pp. 31–40 for a general reference.) Newer images on the GOES–9 series are 1 km per side in the visible and 4 km per side in the thermal bands. Quantization is to 10 bits; a significant performance improvement (for further details of the imagery used in this work, see Section 3.1). High spatial and spectral visible band accuracy (i.e. small pixel size and sensitivity to radiometric change) is balanced by the advantages of night–capable imaging, and the correspondence between vertical development and cloud top temperature in a developing system, which can be estimated from the thermal sensor response values.

Operational meteorology draws on satellite information primarily as a source of manual scene interpretation. A highly trained human expert, knowledgeable of the physics used to describe meteorological systems and with many hours of experience viewing image loops of system development, can discern details from imagery that permit a remarkably accurate subjective analysis–for example, results from 111 cyclones analysed with the PROBE method (described below) had an overall *Probability of Detection* of Bombs of 95%, with a *False Alarm Rate* (incorrect assessment of a cyclone as a *Bomb*) of only 8% [27]. Coupled with near real–time physical measurements, such as wind velocity, temperature profiles and radio soundings, imagery allows the expert to make a reasonable assessment of current conditions and predict possible future conditions. Even in the data–sparse situation that plague real–time analysis of rapid deepeners, with data not always available in real–time, human experts and numerical weather models have managed to produce useful predictions.

1.4 Automation

In 1986, the Pacific Weather Centre (PWC) of the Meteorological Service of Canada (MSC) developed and implemented a procedure for manual detection of rapid deepeners, which they called PROBE (PROcedure for Bomb Evaluation). The system was implemented in the autumn of 1988, presented to the World Meteorological Organization in 1989, and was used operationally for several years (cf. [26], [27], [30]). While results were encouraging, the system was found to be too operator intensive to maintain. Funding cuts and turn–over of experienced staff led to the programme being shelved.

MSC attempted to salvage something from the fruitful results of their work on PROBE by proposing an automated system which would reduce the number of operator intensive steps, provide an objective procedure and assist in training new staff. The proposal was to be designed around the "expert system" concept, and would incorporate the key features of PROBE. Due to further funding difficulties the system was never developed. While not directly related to PROBE or MSC's proposed expert system, the current work addresses similar problems and has been developed with support and advice from the staff at the PWC in Vancouver.

1.5 Characterisation

In order to assist in *Bomb* prediction, this work attempts to formalise a method for computer–based analysis of image sequences. The framework for this method, and the remote sensing concepts it draws on, can be described from the perspective of the philosophy of logic. In formal logic, an *object* cannot be directly known, but is represented by a set of *signs*. Signs could be the colour of reflected light, the shape, the texture etc., all of which build an impression or *construct* in our minds of what the specific object is. In remote sensing, we are limited to non–tactile signs, which are usually restricted to reflected or emitted electromagnetic radiation. The goal of many remote sensing studies can then be described as the determination of a set of unique signs (or electromagnetic signals, patterns etc.) that "speak for" the object under consideration. In logic, the requirement for a definition of an object through its signs is that they be necessary and sufficient to the description: each is a property that **must** belong to the object, and their collection separates that object from all others (cf. [32], [18]).

It should be noted that in the case of satellite sensing of explosive cyclogenesis, the "object" being studied is in fact a human construct; that is to say, a cyclone does not have an obvious, differentiated physical form with rigorously definable temporal and physical boundaries. Its exact definition and qualities are very much matters of interpretation. Sensor response patterns recorded in satellite images are being used to track clouds that are associated with the interaction of cloud microphysical processes and winds blowing through the system (cf. [46]. Even over a short time–frame, the images depict an energy pattern rather than a concrete object (i.e. electromagnetic waves which are "signs" for heat energy, which represent cloud moisture and temperature). There is little difficulty for the human investigator to determine which images in a sequence belong to the set that comprise a specific event; however, the imprecision of the human "definition" would make it very difficult for a machine algorithm to make the same assessment. Further, the distinction between common cyclonic systems and Bombs is sometimes ambiguous, even to meteorologists (as discussed in the literature chapter below). Ideally, a set of satellite image–based signs would exclusively identify the character of the construct under investigation, but realistically the design goal of this work is to find significant correlations between signs and events. The logical requirements of necessity and sufficiency in the collection of signs that defines an object will be viewed as the "ideal" case.

Fundamental to this study is the development and testing of the concept of "Characterisation" from image sequences. For many remote sensing studies, the intended final product is a suitable classification of scene elements, which often become thematic layers for input to a geographic information system (GIS) for further analysis. While classification attempts to find isolated electromagnetic patterns that represent landscape themes, these themes only represent basic elements of the landscape itself. We could call a landscape *well–ordered* (the author's term) if the intermixing of its constituent elements is low. The order of a scene can be described quantitatively using Landscape Ecology metrics. For example, using the *Fragstats* spatial analysis package, measures of *Contagion* and *Interspersion* yield descriptions of the complexity of elements in the landscape [25]. For a well–ordered, time–stable landscape, such as an agricultural field, the collection of all themes describes the total *character* of the individual scene; however, such a collection does not *necessarily* say anything about fields, the particular type of field, or agricultural activity in a generalised sense.

The *object* of analysis in the example above is a specific agricultural field. Using traditional remote sensing methods, the analyst attempts to find spectral patterns, or *signatures*, whose distribution over a localised spatial domain is unique to the feature being viewed (cf. [47]). The classifier takes these signature patterns and attempts to generate a best–probability surface that discriminates between feature classes in the overall scene. The *character* of the overall scene can be effectively captured by descriptive metrics for the classes found in the scene, since the object of analysis (the agricultural field) is well–ordered and, with the exception of long-term studies, geometrically time-invariant.

Remote sensing has been a useful tool for agriculture, forestry, geology, and similar applications that share the property of geometric invariance; however, for applications where there is significant geometric deformation in the scene over time, traditional methods become less applicable cf. [2], [48]. Research in the image processing literature has tended towards: pattern recognition, where specific geometric relationships are important; object recognition, where pattern is used as a signature for specific, and possibly moving, objects; machine vision, where objects are distinguished and discerned as figure and tracked relative to ground elements in a scene; and image understanding, a term loosely used to suggest an attempt at human-like comprehension of scene features from the above elements (cf. [11]–Chapter 9, [19]–pp. 241-242, [24] and the introductory notes to Proceedings in [9]).

In 1992, MSC proposed automation of their manual, subjective PROBE methodology for early *Bomb* detection [27]. Their proposal identified the use of computer expert–systems as a means to capture the expertise of a human operator without need to rigorously define all the steps that go into the subjective manual assessment of overall character. The object of study was explosive cyclogenesis; the features that were being analysed in the manual procedure were satellite cloud patterns and traditional meteorological analyses that corresponded to variables describing the physical meteorology such as: baroclinicity, vorticity, translation speed, and deformation. The system would be "trained" by a human operator to arrive at the same predictions.

The concept of characterisation is an attempt to move away from the generic "image understanding" paradigm, by extending the idea of a "signature" to a geometrically variable object. Thus, character is a general case of the signature concept which incorporates geometric change. The concept also incorporates a wider range of generalisability than does "signature": an object's character should converge with the object's logical definition in the ideal case. The term "characterisation" was chosen because the colloquial meaning of the word suggests an appropriate interpretation, and because it is already frequently informally used in the image processing literature.

1.6 Research paradigm

Given the wide range of disciplines that contribute significantly to the remote sensing and image processing literature, and the complexity of the task of developing computer–based methods to represent human expertise, discussion of the research methods to be used is presented below. The initial impulse for the work came from a purely theoretical consideration: an informal survey of the image processing literature alluded to the notion of *character* as a generalised case of the remote sensing concept of a *signature.* The applied case of explosive cyclogenesis provided a real–world example of a problem which might be described in terms of character, but not in terms of signatures alone. Given the impossibility of experimental controls, or strict reproducibility of ephemeral atmospheric events, the idealised Popperian hypothetico-deductive approach was not applicable. An empirical, exploratory approach was used instead.

The real issue to consider was the information content of the data themselves. Human interpreters have demonstrated that a significant degree of information can be extracted from the satellite image sequences of explosive events. The goal then was to develop a means of capitalising on the available information content in the imagery, using a computer–based analysis. Causal claims are avoided in this work, while confounding factors are addressed along with predictive ones in an inductive, exploratory approach. In bounding the work in this manner, the author hopes to avoid confusion that could result from the assumption of a controlled design, and hence the automatic demand for an experimental hypothesis. In the absence of a formal hypothesis, the experimental expectations were as follows:

> Given human interpreters' significant success in subjectively inferring the character of explosive cyclogenesis from satellite image loops, it is asserted that:
>
> 1. Computer–based systems should be able to use the same information to produce similar predictions

2. Rapid deepeners **do** possess identifiable characteristics that can be discerned on satellite images and that separate them from conventional cyclonic systems
3. Those characteristics can be isolated and used to come up with an alternative "definition" for rapid deepeners
4. The concept of *characterisation* from images is a valid and useful extension of the existing *spectral signature* concept in image processing

In proposing and exploring image characterisation, both elucidation and testing of the concept in a semi–operational environment are attempted. The problem of *Bomb* prediction is addressed and used to illustrate and direct the research. The study seeks to develop new image–based methods to capture measures that reveal the "character" of explosive deepeners and provide guidance for future image–based early warning or prediction of explosive events.

Due to the multidisciplinary nature of the work, pure meteorological and image processing literature are reviewed in separate Chapters. Chapter 2 discusses the meteorological foundation of the research, including the presentation of two models from the literature that taken together, tie the physical effects with what is observed in image loops. In Chapter 3, the imagery as a data source is explored, including its accuracy and information content. Chapter 4 develops novel methods for tracking geometric and radiometric changes in the developing system and includes reference to the image processing literature in meteorology. These methods are then applied to five distinct cases in Chapter 5. A synthesis of statistical treatment of the results is presented in Chapter 6, while overall conclusions and future considerations are presented in Chapter 7.

2. METEOROLOGICAL REVIEW

In attempting to find a suitable "definition" for explosive cyclogenesis that is based on satellite information, a reasonable first step is to look at what criteria define this process from a traditional meteorological perspective. Explosive cyclogensis refers to a process of cyclonic development that is meteorologically unusual compared to the manner in which cyclones typically form. The rapid deepener as a class of events must incorporate a large absolute drop in central pressure coupled with unusually fast development. In the literature, the context of writing usually makes it clear whether an author is referring to the maximum pressure drop over the lifetime of a system or the rate at which pressure changes. To be precise, a convention has been adopted for this work to refer to maximum absolute pressure falls as the system *strength*, and to the rate of pressure change as system *intensity*. Where appropriate, these terms have been applied when referring to the findings of other authors where the distinction is not explicitly made. Any definition of rapid deepeners should account for both strength and intensity relative to the population of all cyclonic systems.

The most commonly used definition for rapid deepeners comes from the work of Sanders and Gyakum [39]. They define Bombs as cyclones whose central pressure drops a minimum of 24 millibars (or hPa) over a 24 hour period. Further, they define a unit called the "Bergeron" to represent this development rate, crediting Tor Bergeron, one of the fathers of synoptic meteorology, with the original concept. Since pressure falls are affected by the latitude of the system centre, they also require that central pressures be mathematically adjusted to 60 degrees north, reasoning that Bergeron's description of a cyclone whose pressure fell one millibar (or hPa) per hour for an entire day likely described an event at the 60 degrees north latitude of Bergen where Bergeron lived and worked.

While many authors acknowledge Sanders and Gyakum's definition, they also often choose different rates and standard latitudes as criteria for determining which events in their studies qualify as Bombs (cf. the list that follows). In their own work, Sanders and Gyakum [39] use 12 hPa/12 h at $45°N$ as an operational standard. Gyakum et al. [14], [15] also use 12 hPa/12 h at $45°N$ while Rogers and Bosart [37] use 12 hPa/24 h irrespective of latitude. These "definitions", or standards appear to be loosely fit to the specific datasets these authors were studying. Roebber [36] suggests that the intent of all such definitions is to separate extreme from moderate events. In this regard he seems to stress the requirement for system intensity over its final strength (as defined above), i.e. that rates of deepening be in the extreme upper range relative to the statistical population.

Roebber's [36] statistical treatment of cyclone intensity showed that when no minimum intensity criteria was used, the frequency distribution was significantly skewed. When an intensity threshold was chosen to split the dataset in two, the resulting datasets were both normally distributed (Gaussian). Roebber took these results to suggest that under the reasonable assumption that cyclone intensity is normally distributed, Bombs and non–Bombs belong to separate statistical populations. In addition, Gyakum et al. [14] found that Bombs last longer – up to 4 days, and comprise the strongest winter cyclones, while Sanders [38] found they travel faster, at an average rate of 18 metres per second versus 13 metres per second for normal cyclones. Roebber [36] further determined that Bombs have a mean deepening period of 45 hours versus 24 hours for all cyclones in his study. Such evidence may lead to speculation that Bombs are the result of a physically distinct mechanism, hence validating the concept, if not the specific definition of Bombs.

Hints to the actual physics behind Bombs can also be inferred from climatological information. First, Bombs are almost exclusively marine phenomena – nearly all explosive deepening occurs over ocean waters [38]. They develop over a wide range of sea surface temperatures (SSTs), from 0–23 degrees Celsius, and are thus much less sensitive to SST than other cyclones [39]; however, Bombs occur preferentially along warm ocean currents such as the Kuroshio in the Pacific and the Atlantic Gulf Stream [14], [36]. It would appear that Bombs requiring a strong horizontal temperature *gradient* rather than a warm absolute temperature for development [39], [4].

A second hint is that Bombs are primarily cold–season events, with a nearly Gaussian temporal frequency distribution with a January mean (in the northern hemisphere) and tails extending to September and to April [39]. Spatially, Pacific events are extratropical, occurring between 30 and 50 degrees north and dissipating north of 50 degrees [14]. Atlantic events are more complex, occurring through a large range of latitudes. Roebber [36] found the average latitude of Pacific events to be 42.5 degrees north. A pattern emerges when we consider that strong cyclones develop in areas of strong meridional upper level flow [38] – areas where the jet stream exhibits pronounced sinusoidal oscillations. Strong thermal differences between the equatorial and polar regions in winter lead to strong meridional flow in the mid–latitude jet stream. Thus, there is good climatological evidence to link Bombs to upper level flow patterns.

The suggestion of upper level influences in *Bomb* development is also supported by the inability of the classic Norwegian cyclone model, as described in Bjerknes [3], to explain the rate of deepening characterises the *Bomb* [5]. Kinematic motions, relating to the momentum and pressure forces that move an air parcel, operate at a time scale that is far too long to explain explosive development. What is needed is an external agent that can coincide with, and amplify, kinematic development; *dynamic forcing* is a process whereby energy from an upper level jet is transferred to a coincident lower level system, and is a process that can explain rapid development while permitting observed cyclonic development patterns.

Petterssen and Smebye's [33] dichotomous labelling of "Type A" and "Type B" cyclones can be thought of as separating kinematic from dynamic effects. Type A systems take energy from a baroclinic lower atmosphere. (Baroclinicity can be thought of as occurring where there is a temperature gradient along a surface of constant pressure, and can result in significant thermal advection.) Type B cyclones are driven by imported energy, primarily from the jet stream. Since dynamic forcing can provide the energy needed to explain deepening rates, it is tempting to offer a definition of Bombs as, "cyclones that are dynamically forced"; however, Type B dynamic effects alone are not sufficient to explain long–term development.

Since Bombs have a central pressure core moving at approximately 65 km/h [38], and the phase translation of the shortwave jet oscillation can be much faster, the period of upper and lower level coupling is brief. Thus dynamic forcing does not appear to account for the extremely long duration of many Bombs [14]. Average relative vorticity advection (the horizontal movement of relative air rotation) which is greatly influenced by dynamic forcing, has been tied to average deepening. Sanders [38], found a direct relationship with $r = 0.872$, and with removal of two outliers, a stunning $r = .924$. An explanation for how such a brief effect as dynamic forcing can have long–term impacts can be found by considering the *preconditioning* of the surface low.

There is good reason to consider the preconditioning by Type A, kinematic influences on explosive development. From numerical simulations, Kuo and Low–Nam [20] found that Bombs are much more sensitive to initial conditions than other mesoscale systems. In a study of nine explosive cyclones they concluded that 60% of the observed development could be accounted for by dry baroclinic processes. In fact, maximum system strength for Bombs has been correlated with strength before the onset of dynamic forcing at 39% [15]. As suggested before, most often cited preconditioning is the presence of strong sea surface temperature (SST) gradients before the onset of rapid deepening (or the beginning of dynamic forcing, denoted t0 herein).

Sanders and Gyakum [39] reason that the contrast between cold air and warm ocean waters results in strong latent and sensible heat fluxes. Latent heat refers to energy required in a phase change of water (e.g. from evaporation or condensation), and sensible heat refers to energy resulting in an air temperature change. Warm SSTs destabilise the boundary layer (the region in the atmosphere of sea–atmosphere interface) by heating it from below, and enhance cumulus convection and frontogenesis [21]; vertical convection in turn provides a means for lower level convergence (air flowing horizontally towards a common point), vertical ascent, and subsequent upper level divergence (air flowing horizontally away from a common point), which along with frontal zone formation are key features of the classic description of cyclonic development (cf. Bjerknes [3]).

Reed and Albright [34] assert that large latent heat, small static (vertical) stability and a strong baroclinic boundary layer are all needed for rapid deepening. Significantly, Kuo et al. [21] found from numerical simulations that while sensible and latent heat contributed equally to early development, stored latent heat released by condensation continued to drive deepening long after t0. Thus, if dynamic forcing is thought of as the starter motor in a vehicle, latent heat is the fuel mixture that keeps the engine going after the initial impetus is removed.

Jet level influences can also be important even before t0. Uccellini et al. [45], [44] identify three distinct jet flows: the polar, subtropical and lower–level jets. The jet stream results in convergent and divergent air flow patterns at the jet stream level. From conservation of mass, upper–level divergence results in upward motion, which is cyclogenic. As the jet stream curves cyclonically (counter–clockwise) around a trough (the southerly bend in the oscillation) there is convergence aloft as the air enters the trough and divergence aloft as the air accelerates as it exits the trough. When the jet accelerates, reaching a maximum value, it results in a *jet streak*. As the air flows through a jet streak, there is divergence aloft (and upward motion) at the right entrance and left exit regions of the jet streak maxima. Upper level divergence from the subtropical jet can initiate convection in low static stability areas by displacing air aloft, while temperature and moisture advection associated with the lower–level jet can increase cyclonic potential. Dynamic forcing occurs when the frontal system becomes phase matched with an upper level shortware [4] or more specifically when the polar jet streak enters the upper level trough [45], [44].

The relationship between the jet pattern and the lower level cyclone can be predicted fairly well. Sanders and Gyakum [39] found that Bombs tend to develop 500 nautical miles downstream (eastward) of a jet trough at the 500 hPa pressure level. A preferred phase match of one–quarter wavelength from the trough also helps identify the critical region for the onset of dynamic forcing. The upper level vorticity maximum, which is to the left of the jet streak, exists 36 or more hours prior to the mid–point of the 24 hour period of maximum deepening; however, the surface vorticity centre only appears 12 to 24 hours before this point [38], and for north Pacific events, maximum deepening appears to occur within 12 hours of surface cyclone formation [14]. The significance of these development times for cyclone prediction is that the surface cyclone will tend to be visible **at most** 12 hours before the maximum 24 hour deepening period, and thus lead time from predictions based on surface measures will be short. It appears that the best hope for long–term prediction would be to track upper level vorticity maxima, rather than surface pressure changes.

A final physical influence on *Bomb* development is the effect of stratospheric air. It is not uncommon for the troposphere to become depressed, or folded, resulting in the extrusion of dry stratospheric air to lower levels. Tropopause folding is typically concurrent with rapid development, and there is evidence such folding is important to the formation of upper level fronts [44]. Further, extrusions of stratospheric air are an important mechanism for vorticity advection, which augments jet level forcing and results in more spectacular deepening.

Figure 2.1 shows a schematic for the structure of a developing cyclone. Three distinct airstreams are depicted that are important in cyclonic development: the warm conveyor belt; the cold conveyor belt; and a combination of upper tropospheric and lower stratospheric air behind the cold front. The warm conveyor transports warm, moist air from the tropical side along the equatorward side of the cold front, which then rises over the warm front. Condensation from this airstream creates the visible comma tail and body. The cold conveyor crosses under the warm front below the warm conveyor, then rises over the surface low and merges with the warm conveyor belt at the jet level downwind. The resulting cloud pattern west of the comma tail

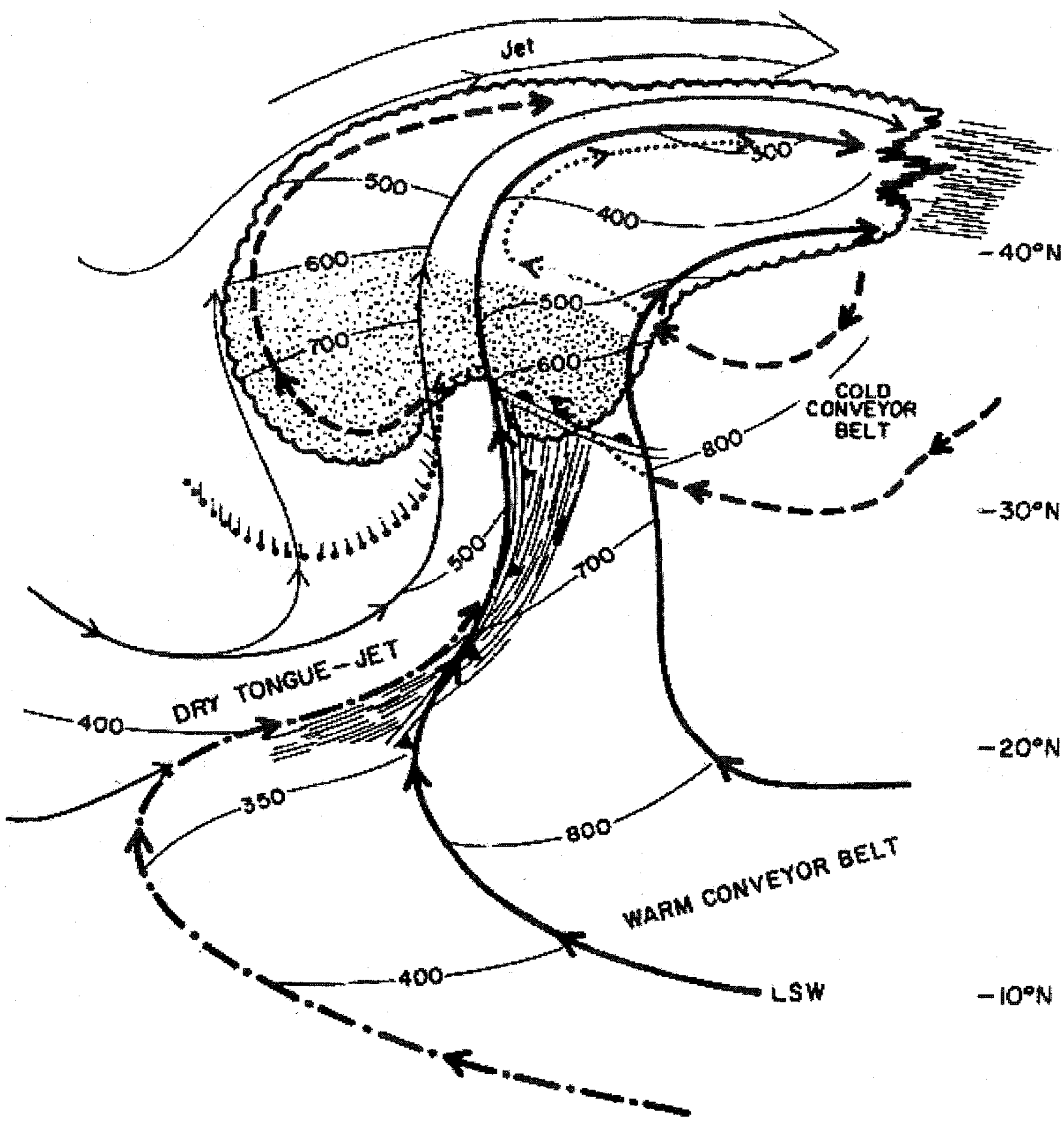


Fig. 2.1. Airflow through a mid–latitude cyclone. From [6]. Solid streamlines show airflow at the top of the warm conveyor. Dashed lines show the cold conveyor. Dot–dashed lines show middle level tropical air. LSW is the limiting streamline of the warm conveyor. Thin solid streamlines show upper level dry air originating west of the trough. Stippling and scalloping show areas of sustained precipitation and mid to upper level cloud respectively.

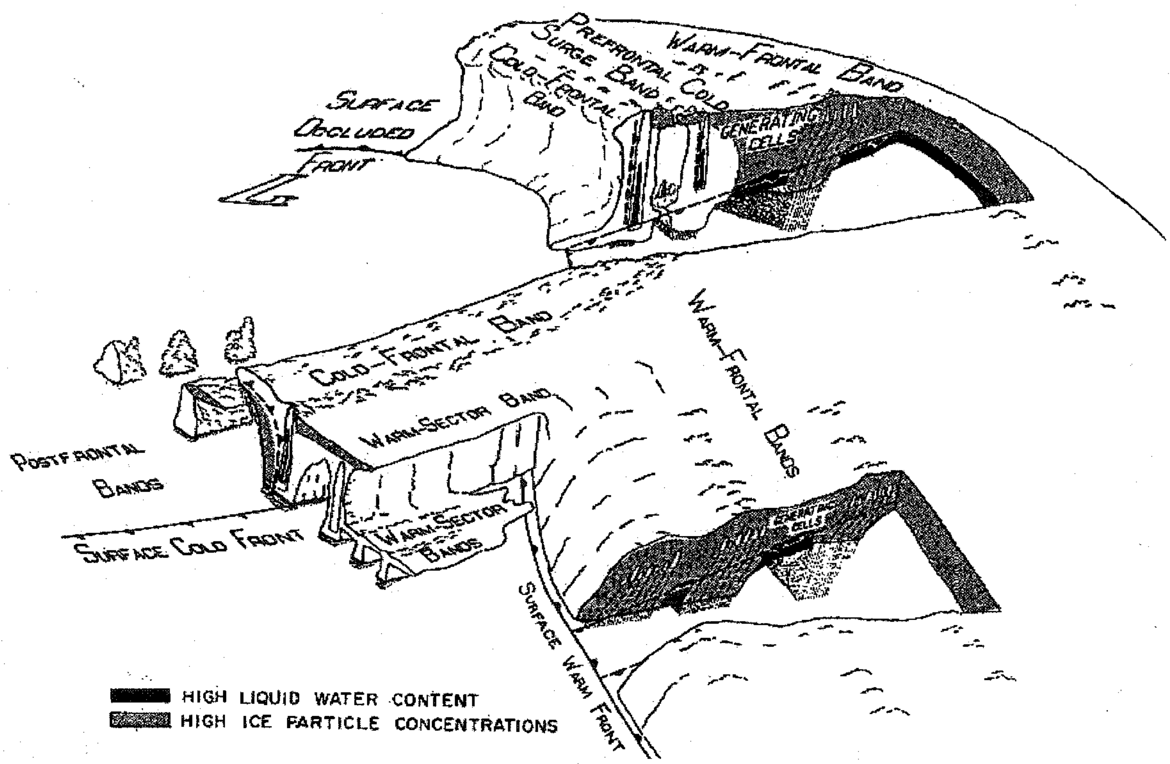


Fig. 2.2. Cyclonic cloud patterns. From [16]

defines the visible comma head, which is sharply defined by interaction with the upper level airstream (as per Carlson [6]).

Figure 2.2 shows cyclone cloud patterns in more detail. The low pressure core that develops about the frontal triple point, where the advancing cold front meets the slower moving bent back warm front, can be seen in thermal and visible band images as a prominent cloud–free area [31]. Here the low is seen in the upper left portion of the figure, since the figure depicts a mature system (the low moves westward of the triple point as it develops). A thick cloud mass spirals inward towards the eye, forming the comma head of the cyclone. There is typically deep convection, visible as cumulus towers, preceding the advancing cold front [39], while a rapidly expanding cloud–top field about the surface low indicates the prominence of latent heat release [34]. Scattered convective cells are also often visible behind the cold front in the form of cumulus puffs.

Upper level influences are also apparent from satellite images. At the jet level, the co–location of a broad cirrus shield about the subtropical jet and a lesser area of cloud near the exit region of the polar jet are visible. Further, a band of very dry air is visible in thermal infra–red images where the two jets merge [44]. The GOES water vapour channel can reveal stratospheric extrusions as low as the 700 hPa level. This channel is a thermal infra–red product whose response depends on the moisture content and temperature of the entire water column represented by each pixel (see Chapter 3 for more details). It is useful for tracing moisture advection and vertical motions [44]. A dry air tongue behind the advancing cold front indicates extrusion of stratospheric air, and is visible on ozone composites [34]. The presence of ozone, a product of upper atmospheric radiation, is a strong indicator of stratospheric air, and can be measured with the Total Ozone Mapping Spectrometer (TOMS) aboard the Nimbus satellite series [44].

Ultimately, the utility of satellite observation of cyclogenesis comes from matching cyclone development to patterns that can be imaged from space. Weldon's early work provides the classic description of satellite imagery and cyclonic development, as published in "Cloud Patterns and the Upper Air Wind Field" [46]. In 1990, Shapiro and Keyser [43] proposed a frontal cyclone development model that was distinct from and complementary to the classic Norwegian cyclone model (cf. [3]). Building on both works, Neiman and Shapiro [31] proposed an integrated frontal cyclone model that is compatible with physical descriptions of development while describing the relevant coincident cloud motions. A physical description as presented by Rogers and Bosart [37] will be denoted model **A** and presented in tandem with the Neiman and Shapiro model, denoted model **B**. Both models describe four stages/phases of development:

Phase I: **A** *Incipient Stage*: before the period of most rapid deepening.

B *Incipient Broad–Baroclinic Phase*: characterised by the development of a baroclinic leaf structure in the cloud field (a leaf–like cloud shape).

Phase II:

A *Explosive Stage*: a period of at least 12 millibar (or hectopascal) deepening in a 12 hour period.

B *Frontal Fracture*: the distinct development of a comma shaped cloud formation about the surface low.

Phase III:

A *Mature Stage*: central pressure and storm area are quasi–steady.

B *Bent–Back Front and Frontal T–Bone*: development of a cirrus shield comma head from the frontal triple point westward along the bent–back front (the bent–back, or *occluded* front occurs when a faster cold front overtakes and lifts a slower warm front).

Phase IV:

A *Decaying Stage*: central pressure does not decrease and there is a decrease in storm intensity.

B *Warm–Core Frontal Seclusion*: the bent–back front and comma head spiral inward toward the cyclone centre, encircling the warm secluded air about the low pressure "eye". The only cold (usually cirrus) cloud visible on the frontal edge is within 250 kilometres of the frontal triple point.

The Neiman and Shapiro [31] model is illustrated in Figure 2.3. While temporal correspondence between physical and cloud field events cannot be pinned down for certain from the models presented, it appears that a strong connection can be inferred between them. In *Phase I*, dynamic forcing has not yet begun, so cloud field expansion is driven primarily by convective and/or kinematic mechanisms. The baroclinic "leaf" cloud structure indicates a baroclinic zone (frontal zone). In *Phase II*, explosive deepening has been initiated by upper level dynamic forcing (positive vorticity advection, resulting in divergence aloft and upward motion), coinciding with the appearance of the developing comma–cloud structure. The *Phase III* *T*–bone

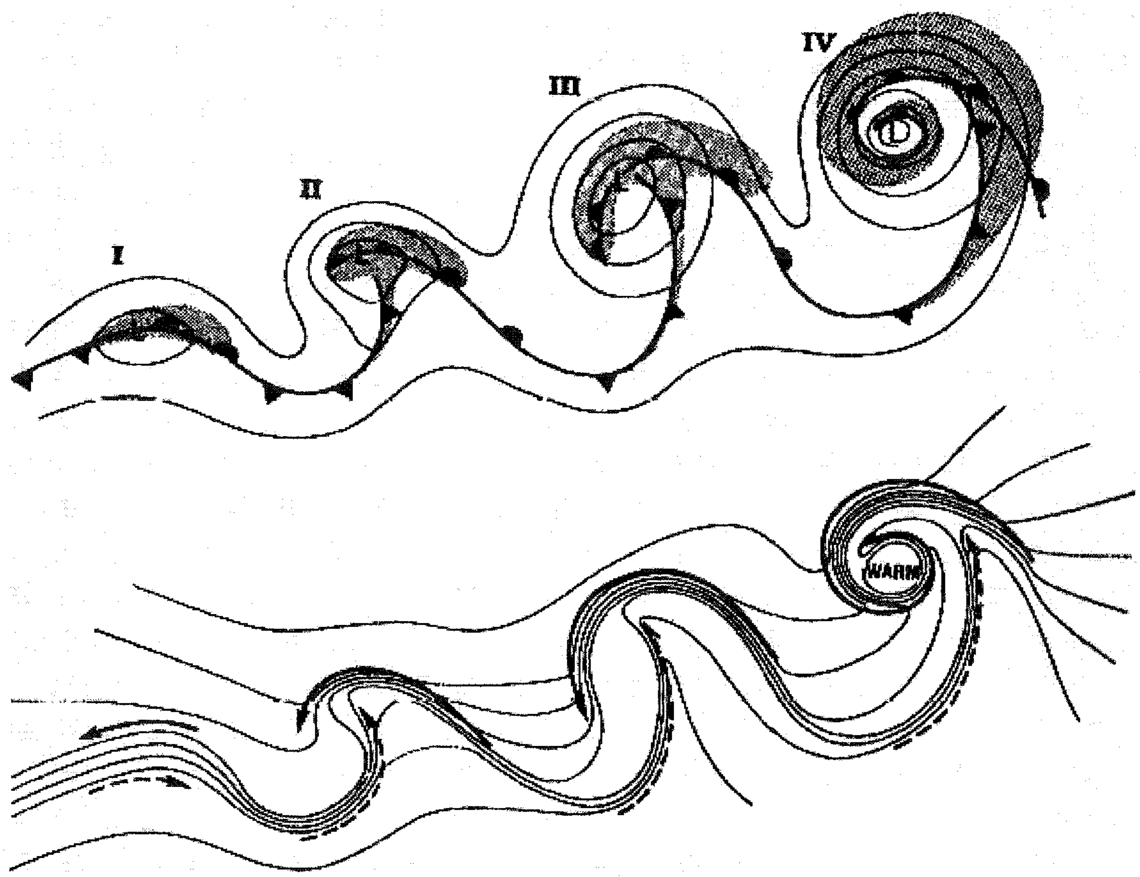


Fig. 2.3. Four stages of the *Neiman and Shapiro* [31] model of explosive development. Top figure–pressure and frontal lines with shaded cloud fields; bottom figure–isotherms with cold (solid) and warm (dashed) currents.

shape may indicate convection along the occluded front and continuing lower level convergence. *Phase IV* visually indicates that the storm is "drawing in" on itself as the frontal structures begin to decay, explaining why cold cloud is geographically limited in this phase to the proximity of the surface low, which lies beneath an upper level low.

Based on the Neiman and Shapiro [31] model, and other cloud–field evidence presented above, it would appear that the best option for surface level based early warning would be to watch for the critical change from *Phase I* to *Phase II.* Specifically, as the "leaf" pattern changes into a "comma" pattern, it may be possible to detect the onset of rapid deepening. While it would be preferable to offer mariners warnings *before* t0, the knowledge that a developing storm will explosively deepen may help save lives (L. Neil, pers. comm., 1998. See Section 3.1). As methods were developed for this work, the temporal period around and just before t0 was seen as the critical starting window for detection. Subsequently pushing back the prediction window must be seen as the ultimate goal of operationalising image based prediction methods.

3. DATASET DEVELOPMENT

Having framed the nature of the inquiry for this study, and explored the relevant meteorological literature, the problem of suitable methods needs to be addressed. A useful starting point and directing influence was to look at the quality and limitations of available data. By the nature of the project topic and the limitations on ocean-based meteorological sources, finding suitable time series of image data was essential. Several online sources, such as NOAA (National Oceanic and Atmospheric Administration, USA) and NASA (National Aeronautics and Space Administration, USA) web sites were considered, but rejected because the continuity of image sequences was poor. Subsequently, a variety of CDROM sources were considered. A promising source was the ERICA (Experiment on Rapidly Intensifying Cyclones over the Atlantic) project, conducted on the eastern seaboard from December 1988 through February 1989 ([35]). The ERICA dataset was created specifically to study the behaviour of explosive deepeners. Unfortunately, from written requests it appears that the raw images were never distributed with the ERICA dataset.

Since ERICA data turned out to be unsuitable, focusing generally on Pacific cases seemed reasonable. A desired minimum requirement was specified as a representative set of imagery which expressed the full range of eastern Pacific weather over at least a single season. Since the climatological review suggested a Gaussian frequency distribution centered on the month of January [39], imagery from late fall through early winter were expected to be especially valuable.

3.1 Data sources and database management

Mert Horita, (Manager, Environment Canada Applications and Services) at Vancouver's Pacific Weather Centre (PWC), a branch of the Meteorological Service of

Canada (MSC) introduced the author to Laurie Neil (Head, Environment Canada Meteorological Research and Development), a meteorologist at the Centre who has been involved with explosive cyclone studies for several years. Laurie arranged that a continuous feed of images from GOES–9 West be established, using PWC's *MetPC* software. *MetPC* offers new imagery every half–hour. A half–hour frequency shows all major development and is considered quite adequate for tracking large systems.

MetPC's raw imagery is standardised to a 1024 pixel per side raw frame quantised to 8–bit depth ($2^8 = 256$ possible brightness values), and projected into a polar stereographic system with origin at $(140°W, 60°N)$. Real–time images are transferred automatically via internet file transfer protocol in compressed form at 512 pixels per side at 6–bit depth (2^6=64 values) using the *JPEG* (Joint Photographic Experts Group) image format. *JPEG* is a "lossy" format, meaning that the reconstructed uncompressed images are nearly but not exactly identical to the originals.

GOES–9 covers the north–eastern Pacific region. The extent of the image frames provided by PWC is shown in Figure 3.1. The system offers five bands (on the Imager, the primary instrument), as shown in Table 3.1. Pacific Weather Centre provided three of these bands covering the Visible (VIS), thermal–infra red (TIR1 or TIR from herein) and "water vapour" (WV) regions of the electromagnetic spectrum. The WV channel falls in the thermal–infra red range, and is especially sensitive to thermal re–emission from water in the vapour phase. It roughly corresponds to the amount of water vapour in the air column directly below the sensor element, but saturates quickly with depth for clouded areas (L. Neil, Pers. Comm. See above.) Thus, unlike the TIR channel, WV is not only a representation of cloud–top but can represent the entire depth of the atmosphere below the sensor.

The dataset for this study consists of images collected over the period from December 1996 through August 1998. These images were compiled initially on DAT (digital audio) tape and later sorted and re–archived on CDROM. The CDROM dataset contains 27846 infra–red and 24388 visible band images. It also contains 19141 water vapour images matching certain of the IR and VIS frames. The lower number of

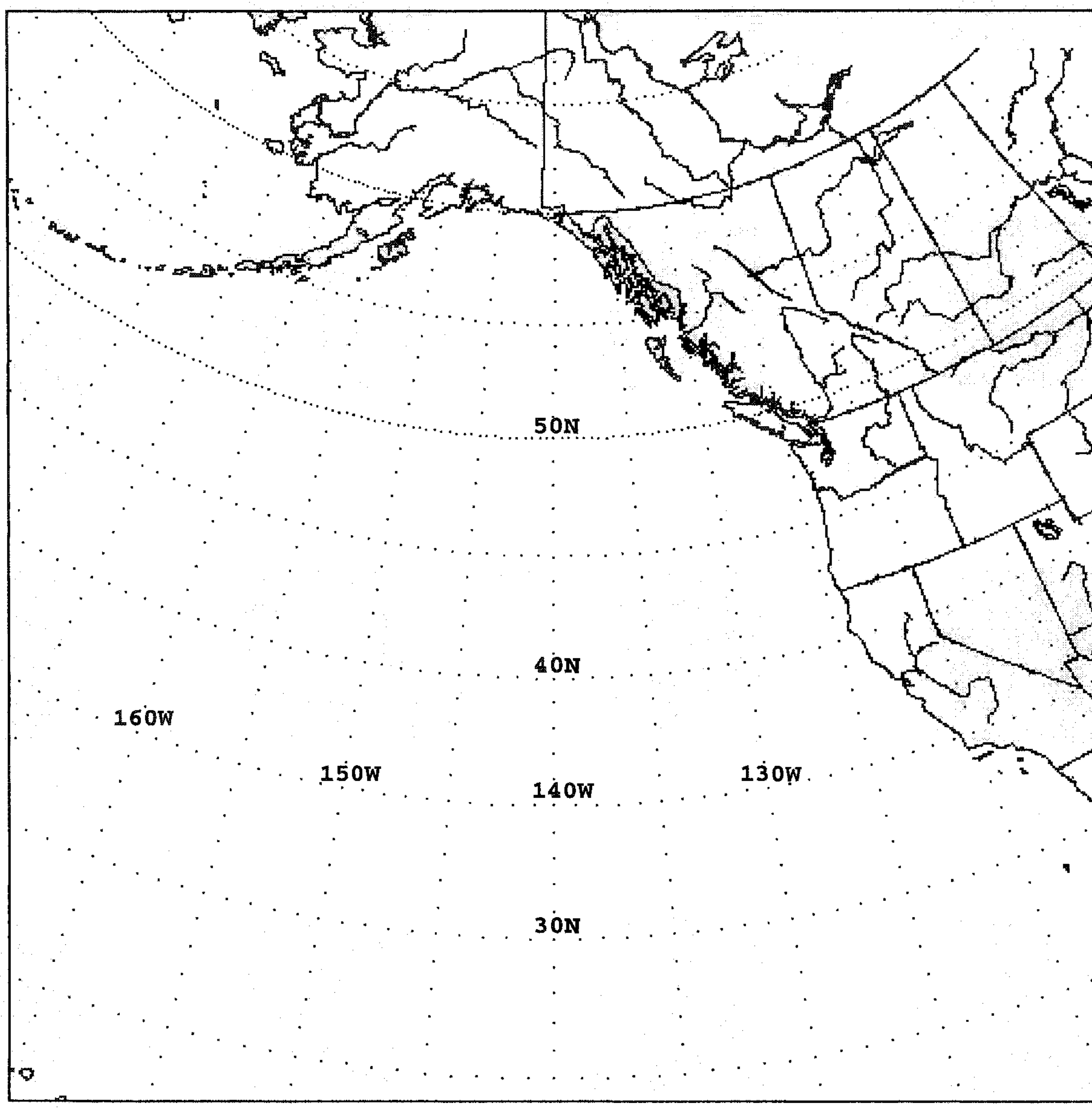


Fig. 3.1. Coverage extents of images from GOES–9, as provided by Pacific Weather Centre, Meteorological Service of Canada. All satellite frames in this work are from this source.

Table 3.1
GOES–9 (WEST) Band Properties [29].

	VIS	NIR	WV	TIR1	TIR2
Lower Bound	$0.55\mu m$	$3.80\mu m$	$6.50\mu m$	$10.2\mu m$	$11.5\mu m$
Upper Bound	$0.75\mu m$	$4.00\mu m$	$7.00\mu m$	$11.2\mu m$	$12.5\mu m$
IFOV	1 km	4 km	8 km	4 km	4 km

collected water vapour images was the result of the late realisation of the potential utility of these images.

Spatial resolution of the GOES system is constrained by the light collecting ability of the sensor for each specific band. The VIS signal corresponds to reflected sunlight from the 6000 K radiant temperature of the sun. Both the TIR and WV channels represent radiation from the earth–atmosphere system radiating at approximately 300 K. Since the *Stefan–Boltzmann Law* expresses total radiant power as proportional to the fourth power of the temperature source over all wavelengths (assuming a blackbody source), far less radiant energy is available from cooler objects radiating predominantly in the longer wavelengths, such as the TIR and WV channels record (cf. [23], pg. 6). The area of collection of radiant energy at the sensor is termed the instantaneous field of view (IFOV), and corresponds to the final resolution of each picture element (pixel) in the image scene. To produce useful returns, the pixels in each image should record values that represent the full dynamic range of the sensor. A specific comparison of available energy at the sensor for a fixed IFOV can be made by application of *Planck's Law*, which relates emissive energy to temperature and wavelength for a given unit area. Figure 3.2 shows the range of four GOES spectral bands and how they relate to energy curves for blackbodies emitting at typical cloud temperatures. Higher energy in the visible band allows the sensor to be saturated when averaging over a smaller target area, allowing for a smaller IFOV for equal band widths. High spatial visible band accuracy is traded off versus the advantages of night–capable imaging, and the correspondence between vertical development and cloud top temperature in a developing convective air column, which can be estimated from thermal response values.

3.2 Data assessment and case selection

The primary software used in exploratory analysis was the *PCI* remote sensing package (©PCI Enterprises Inc., 1997, Version 6.2.2). *PCI* offers a wide range of image processing, format conversion and georeferencing support, as well as a high level scripting language for automation of analysis procedures. Other image processing tasks were performed in ENVI/IDL (©Research Systems, Inc., 2001). Statistical analysis was performed in the *S-PLUS* (©MathSoft, Inc., 1996) environment and with the freely distributable *R* software package (©GNU General Public License, 1998). Both packages are derivatives of AT&T's *S* statistical scripting environment and share a similar syntax and user interface. A variety of *UNIX* tools were also used including *PERL* language (©Larry Wall, 1998) scripts and *C* language code to manipulate datasets, and produce some elementary image statistics.

In order to detect and track weather systems in the dataset, a *PERL* and *C* procedure was created that called image conversion and manipulation routines from the publicly available *ImageMagick* package (©E. I. du Pont de Nemours and Company Inc., 1998). For both TIR and WV bands, available image times were sorted and four images where chosen as close as possible to 0, 6, 12 and 18Z (Zulu, or Coordinated Universal Time). There were occasional gaps in the archive due to transmission or storage problems, causing some approximation in the desired six hour interval. These images were averaged to create daily composites, and *ImageMagick* was called to paste all images for the month together into a single contact sheet. An example is shown in Figure 3.3. Raw values greater than 68 in the TIR and less than 70 in the WV bands were masked out since they imaged areas too warm or too dry to represent cloud cover. Contact sheets formed a useful way to quickly survey development over a long period of time.

Another procedure was performed from the TIR image set to help find strong

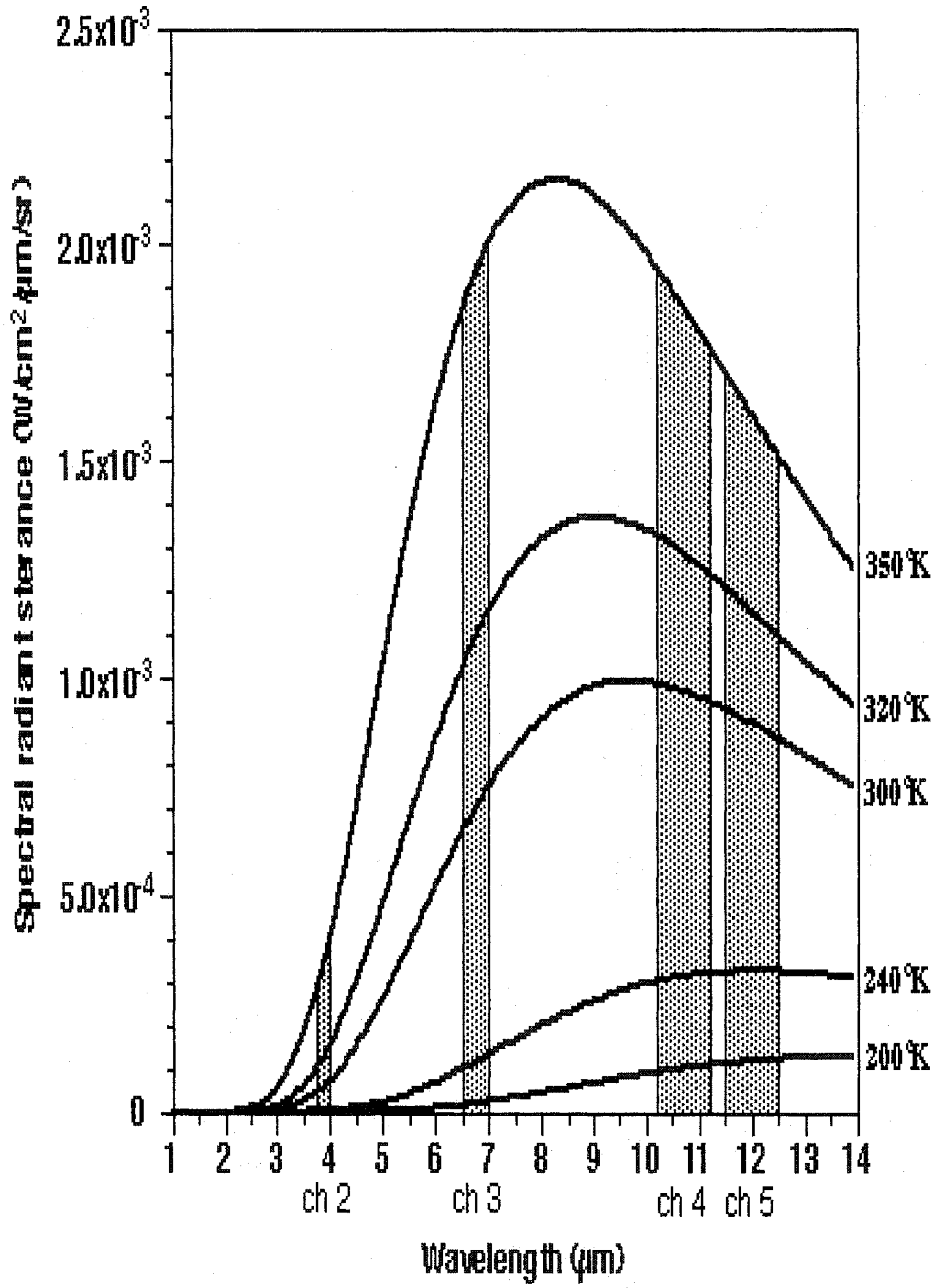


Fig. 3.2. The spectral "width" of GOES–9 channels 2 through 5. Spectral sterance curves are typical for emissions from terrestrial features and cloud bodies in $W/cm^2/\mu m/steridian$. From [7].

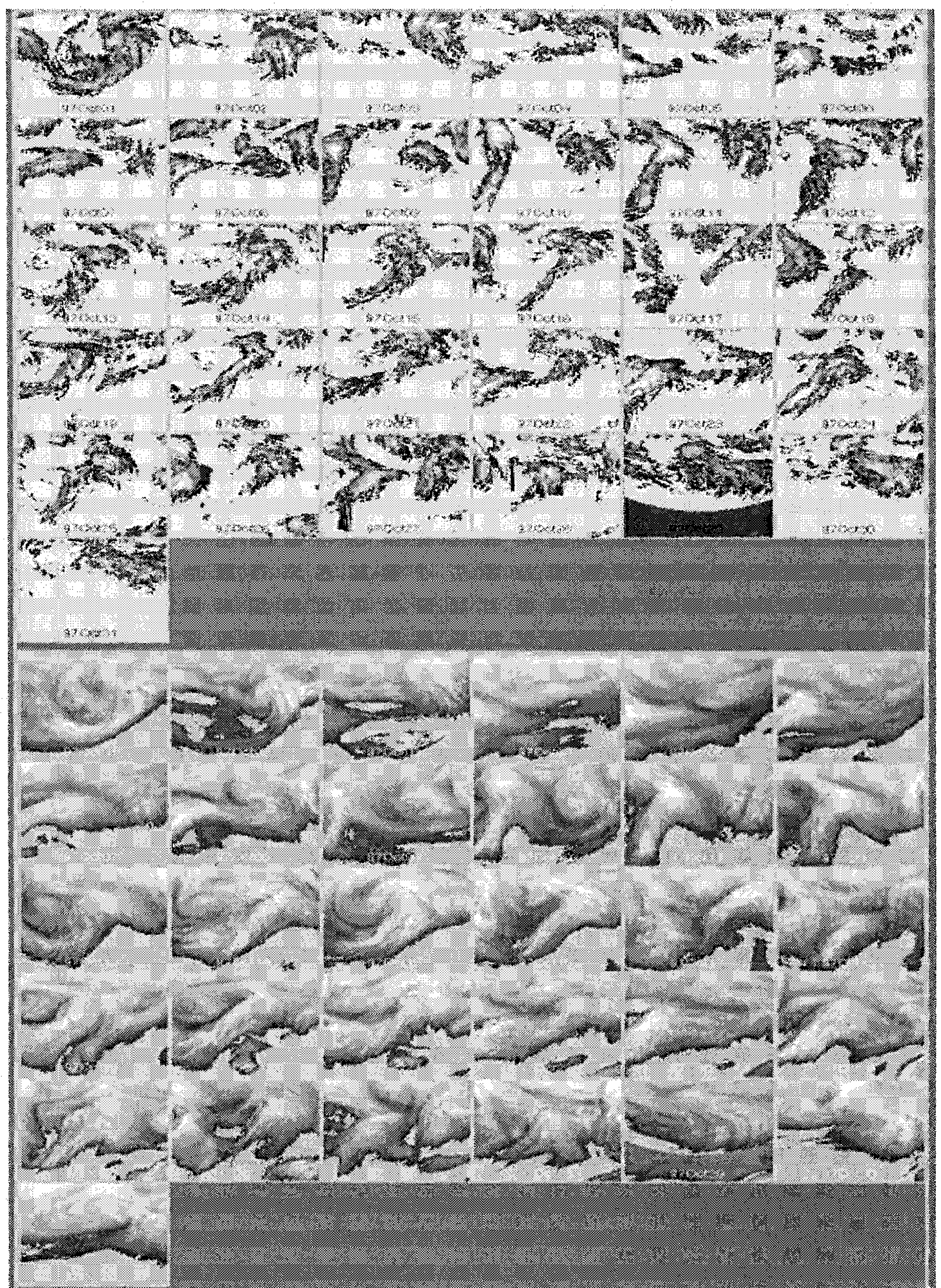


Fig. 3.3. Thermal Infra–Red and Water Vapour contact sheets. Each frame is a daily average of 4 images at 6 hour intervals. Top frames show visible band; bottom frames show Water Vapour band.

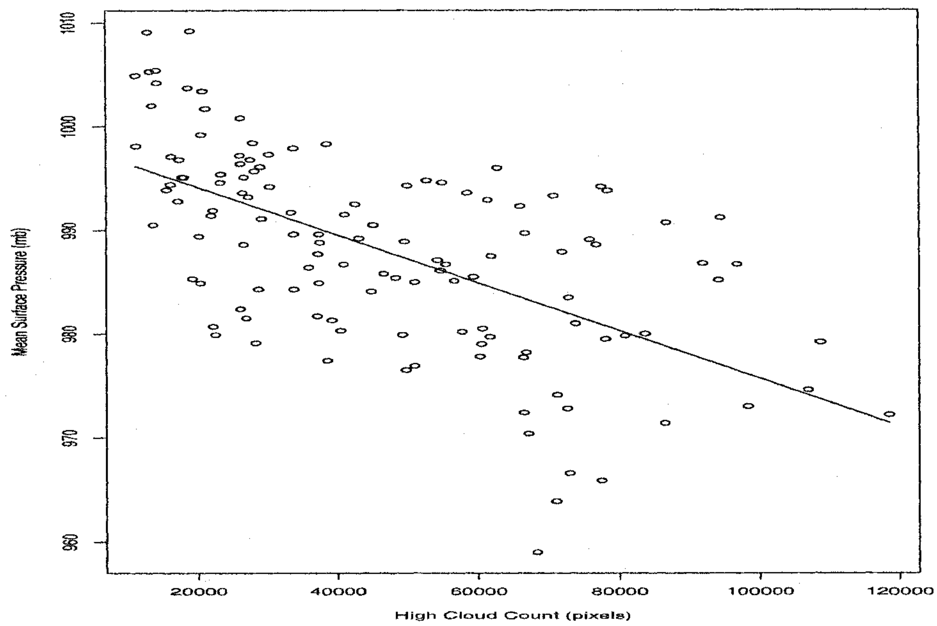


Fig. 3.4. High cloud counts correlate to mean surface pressures with $R^2 = 0.37$.

cyclonic cloud structures. Since the TIR response is proportional to emissive temperature, a threshold could be chosen to separate "high" from "low" cloud in each image (see Section 4.1 for details). Since cyclonic systems are revealed by highly structured high cloud masses, it was thought that the percentage of high cloud pixels in an image might be indicative of "storminess". To partially verify this theory, U.S. government NCEP/NCAR (National Centers for Environmental Prediction/National Center for Atmospheric Research, USA) re–analysis mean sea level pressure records in the period of December 12, 1996 through June 29, 1997 were obtained. A correlation between mean daily high cloud count and mean daily sea surface pressure of $R^2 = 37\%$ was found over the indicated period (Figure 3.4). Since mean pressure falls should be highly influenced by large cyclonic mid–latitude systems, high cloud counts may be a useful way of locating such systems. High cloud totals were computed for each frame in the image set.

Assuming a fairly simplistic geometric representation of a developing system allows

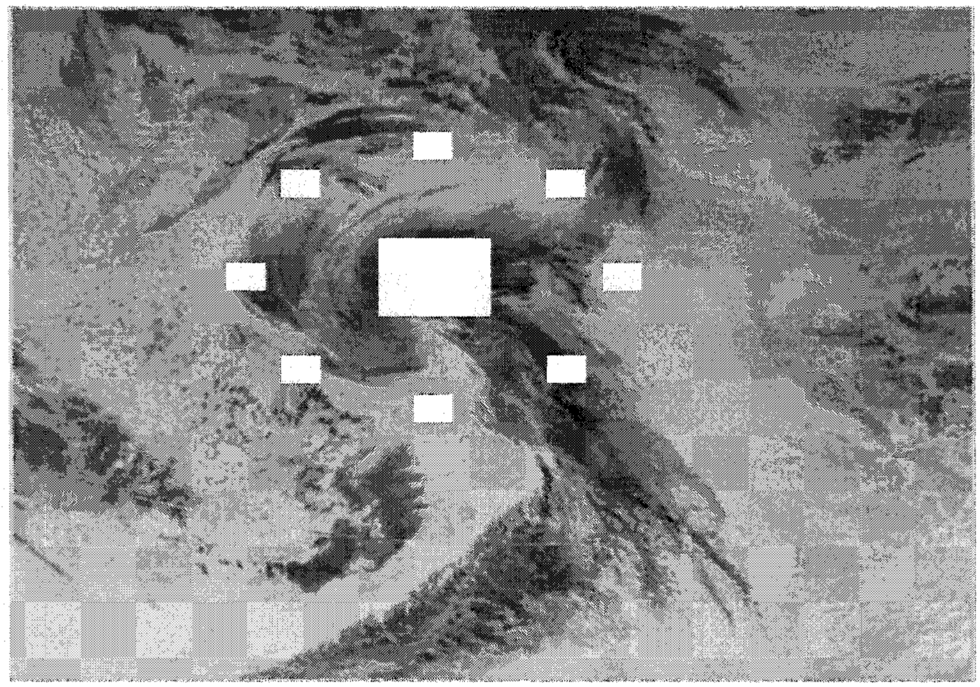


Fig. 3.5. Central square mask of area A and eight surrounding squares B_i where $\sum B_i = A$ superimposed on a mature cyclone (raw TIR image: dark areas are cold and cloudy).

for another system locator algorithm. A mask was developed that compares high cloud counts within a central square polygon with those of eight surrounding squares whose areas total to that of the central polygon (Figure 3.5).

The count of high cloud pixels in area A is divided by high cloud counts from $\overline{B_i} + 1$(to avoid division by zero) from the surrounding squares. The resulting index has a high value when area A is well filled with high cloud and areas B_i are on average free of high cloud. In image processing terms, the mask defined here is fairly complex and requires significant computation time to apply. Therefore, to increase the execution speed of the code, counts from area A below 50% were not considered. Counts from area A above 90% were also ignored to avoid large blocks of high cloud that do not contain at least some mid or low level cloud. These criteria are designed to produce a large index value when the mask is centered over a large, mature cyclonic system, as in Figure 3.5. The size of the final mask (as applied in Chapter 5), was scaled to fit a sample of mature cyclones from the available systems in the image dataset.

3.3 Assessment of Radiometric Information Content

Diurnal loss of coverage during night hours in the spatially and radiometrically superior VIS image product is a serious limitation to using this band as a surrogate for traditional meteorological information. Comparisons between daytime values of the VIS and thermal products helps to suggest the extent to which these 24 hour usable images can be substituted for the VIS information. Method development for actual cases was undertaken using a full resolution uncompressed (at 1024 pixels per side) sequence of an explosive event that occurred April 24-25, 1996. Initial radiometric investigation of the case dataset involved a principal components analysis to explore the inter–band correlations. The April sequence contained 21 frames from each of the three bands. Frames one through 19 are in one hour intervals, while frames 20 and 21 are in six hour intervals. Output from the principal components algorithm yielded covariance and variance values. Dividing the covariances by the product of both variances in a band comparison yields the inter–band correlation. Plots of correlation between the TIR and VIS bands indicate that frames 6 through 14 have the highest values (Figure 3.6). Plots of WV and VIS correlation show nearly identical results (not shown). Frame 6 represents 1700Z (Zulu or Coordinated Universal Time), or 9 a.m. Pacific Standard Time. (Standard notation in Meteorology is to indicate <day>Z<hour>–eg. 12Z24 for 12 hours on the 24th day, Coordinated Universal Time.) For these images, initial high correlation with the visual band corresponds to morning solar ascent. As expected, frame 14 was captured at 0100Z or 5 p.m. Pacific Standard Time, corresponding to solar descent. Frames 20 and 21, at 1200Z and 1800Z April 25, represent increasing solar elevations the following day at 4 a.m. and 10 a.m. Pacific Standard Time. Note also that frame centre ($140°W$) is approximately one time–zone west of Pacific Standard Time, so sunrise is about one hour later.

During peak sunlight hours, the mean correlation between TIR and VIS bands

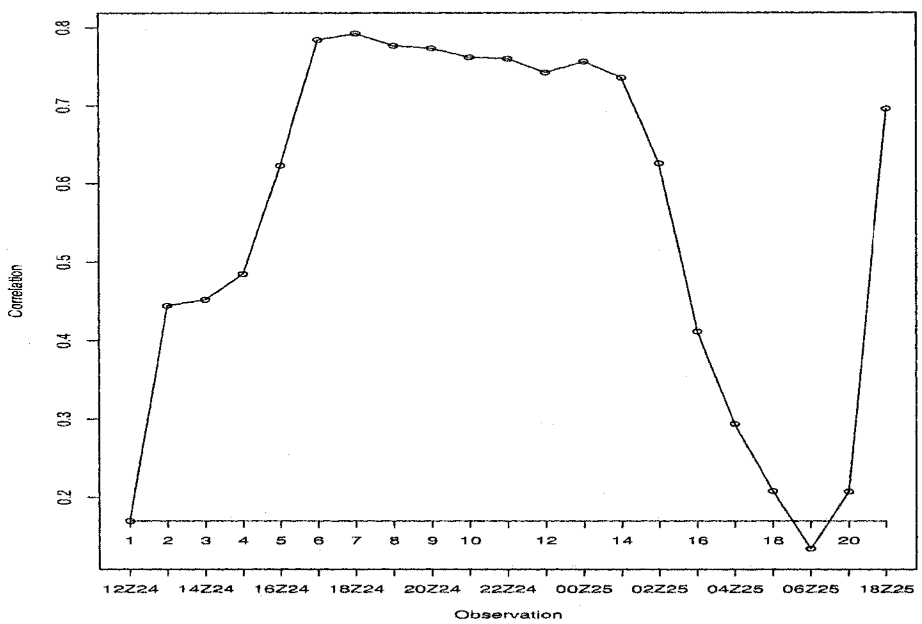


Fig. 3.6. Correlations between TIR and VIS cycle as VIS image comes into frame (during daylight hours). Top horizontal axis: sequential frame number. Bottom horizontal axis: Coordinate Universal Time in standard notation.

was 0.76 with a range decreasing over the day from 0.79 to 0.74. The WV and VIS bands started with a lower correlation, increased slightly and then decreased over the day from 0.67 to 0.61 with a mean peak hours correlation of 0.64. By comparison, the TIR and WV correlation (Figure 3.7) remains high through all the frames, with a range from approximately 0.88 through 0.85 and a mean value of 0.86. The plot of the TIR to WV correlation exhibits a definite sinusoidal shape (Figure 3.7) but correlations between the bands do not vary greatly. The correlations suggest that TIR is a reasonable radiometric substitute for VIS in terms of information content for those frames where VIS is not available; however, reduced spatial resolution for TIR remains a liability. Fairly consistent correlations for TIR and WV suggest that synthetic principal components from these bands should be consistently interpretable. Since the first principal component of a two–band system will contain the greatest possible pooled variance, a synthetic product based on such a procedure may offer a remediation strategy for information loss due to the choice of emitted versus reflected imagery.

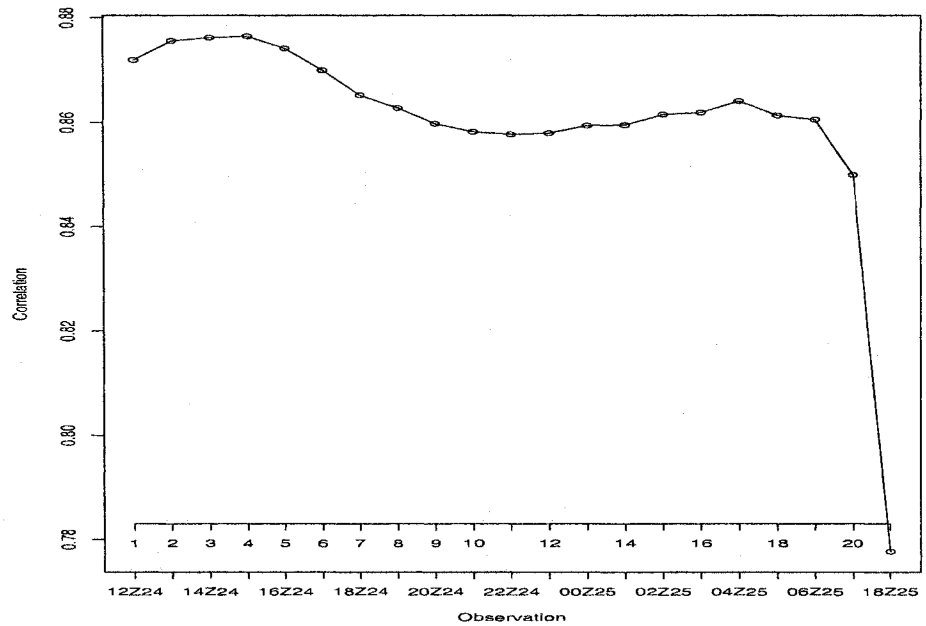


Fig. 3.7. Correlation between TIR and WV frames.

4. IMAGE PROCESSING BACKGROUND AND METHODS DEVELOPMENT

Traditional remote sensing change detection techniques monitor localised spectral changes in geographically static targets. Since a developing cyclone represents changes in a dynamically structured target, traditional change detection methods are not directly applicable (see Chapter 1). In particular, since the interactions between momentum, moisture, airmass density, temperature etc. are continuously evolving over time, and since evolution inherently defines a cyclone, some manner of rectifying geometric changes from frame to frame is required before radiometric changes can be assessed in a spatial manner. A similar problem has been tackled for "deformable templates" such as an expressive human face (e.g. [42]). In this section, image enhancements are presented that assist in identifying developing cloud structures, four geometric approaches to defining storm evolution are discussed, and early radiometric work is presented that resulted from a geometrically controlled sequence of image frames. Finally, methods used to process raw output of radiometric and geometric procedures are presented, and, the statistical treatment of these data is described.

4.1 Image enhancement

A major concern for any image rectification procedure is the identification of suitable control or "tie" points. For such points to be located, visually distinct and persistent features must be identified. When image "targets" are not rigid bodies, the problem can be severe; major works such as [42] (winning PhD thesis in the British "Distinguished Dissertations in Computer Science" series) have been devoted solely to this topic. Since cloud bodies are somewhat amorphous, and continually evolving, pre–enhancement methods were critical to developing a means of identifying such control points.

A survey of the literature suggests image processing work in meteorology has largely focused on the tracking of cloud masses, and separation of cloud decks. Cloud mass tracking allows weather patterns to be discerned from a quasi–automated analysis. Arnaud et. al [1] used a temperature threshold of -40°C to separate high cloud (cloud top temperatures decrease with height; see Table 4.1). Image segments formed from this threshold were labeled and tracked through time, following both merging and splitting of distinct cloud masses. Schmetz et. al [40] used a histogram analysis technique to extract representative identifier points from cloud masses to produce cloud motion vectors. Coakley and Baldwin [8] and Ebert [10] used variations of the histogram–spatial coherence method to extract and use information about cloud deck levels. This method is based on the observation that when the local standard deviation of a small subwindow of pixels is plotted versus the local radiant temperature, an arch shape is defined. Thick clusters of points in the plot at lower standard deviations represent areas with spatially coherent sensor response. In the arch shape, two such "feet" are found; the one that plots on the left at lower temperatures represents cloud cover; that at the right at higher temperatures represents areas without cloud cover.

Perhaps the most immediate and useful information available from satellite images is cloud top temperature. The thermal infra–red (TIR) channel provided in the *MetPC* product was compared in several image frames to the radiant temperatures reported by the software for sample pixels. A simple linear relationship was developed by the regression $T_{rad} = 0.584_{TIR} - 90.1$. Applying the Stefan–Boltzmann equation, we can also compute the radiant energy emitted by clouds as $I = \sigma(0.584_{TIR} - 90.1 + 273.15)^4 [W/m^2]$, where σ is the Stefan–Boltzmann constant. The actual procedure for temperature conversion was later supplied by MSC, and is included in Appendix B-1; however, in the temperature ranges of cloud top radiance the simple regression returns values to within a fraction of a degree of those from the more complex procedure. For simplicity, the regression method was used. Later, it was realised that since GOES calibrations were identical for all frames in the dataset, a simple lookup–table could be used to convert the raw digital numbers (DNs) to their temperature equivalent.

Since cloud temperatures are proportional to cloud top height, it is possible to place any given cloud pixel into a cloud–type group based on its equivalent temperature. A handbook of meteorology ([41]) was consulted for representative cloud temperatures, and those were then divided into low–cloud, mid–cloud and high–cloud classes (Table 4.1). Edge detection and enhancement were performed to help create a bitmap that traces the boundary of mid and upper–level flow as seen in jet–level cirrus in the imagery. For this procedure, low–cloud areas were initially isolated, then a 7×7 mode filter was applied to the resulting image. The mode filter replaces the central value of a moving window with the numerical mode of all pixels in the window. In this case, the window is seven pixels per side. Once a new value is assigned, the moving window advances one pixel and the process is repeated until the entire image has been filtered. The effect of the mode filter is to make similar DN areas more uniform, and thus increases the edge contrast between differing areas. Since the low cloud regions have been masked out, strong contrast exists between the blanked areas and the mid to high–cloud regions. A 3×3 Sobel edge detector was then applied to the mode filtered image. The Sobel mask is sensitive to localised changes in adjacent areas. In raw form, the Sobel filter calculates the gradient of the image vector along the x and y axes, from which the magnitude of the gradient vector can be calculated. PCI returns the Sobel vector magnitude as $|\overrightarrow{G}| = |\overrightarrow{G_x} + \overrightarrow{G_y}|$. Gradient magnitudes greater than a suitable minimum (a gradient value of 200) were retained and stored as a separate bitmap. The high magnitude gradient edges cleanly delineate mid and upper level clouds. For general references to filters in image processing, cf. [17], [11].

It was decided that due to diurnal loss of coverage the methods development stage would ignore the visual band data. An attempt to mitigate the unavoidable data quality loss from not using the richer visual band (see Chapter 3) was made by using a synthetic principal components image extracted from the raw near infra–red and water vapour channels. These images were formed by performing a histogram matching between the bands to ensure equivalent dynamic range, and then a principal

Table 4.1
Representative cloud types and temperature ranges (adapted from [41], pg. 165).

Cloud Type	Base Height (km)	Base Temperature (°C)
Cirrus	5 – 15	-70 – - 30
Cirrostratus	5 – 15	-40 – -25
Cirrocumulus	5 – 12	-40 – -25
Altostratus	3 – 8	-30 – -10
Altocumulus	2 – 8	-30 – -10
Nimbostratus	0.5 – 2	-10 – -20
Stratus	0 – 2	-10 – -20
Stratocumulus	0 – 2	-10 – -20
Cumulus	1 – 4	-5 – 25
Cumulonimbus	1 – 4	-5 – 25

components decorrelation stretch. The resulting images capture the best properties of both bands, containing a higher information content (a necessity of the principal components method) and having sharper spatial detail and contrast (based on visual assessment).

Subsequent investigation led to production of an IR based pseudo–colour table that automatically covers low–cloud areas with a blocking mask, dynamically stretches mid–cloud ranges and applies a coloured enhancement to cold–cloud regions. Linear contrast stretch was performed for temperatures between -5 and -30 degrees Celsius from dark grey through white. The colour enhancement maps temperatures from less than -30 to -70 degrees Celsius or colder as pure red through green to pure blue. When combined with the "flicker" function in *ImageWorks*, to switch between the infra–red enhancement and a linear stretched view of the whole infra–red range, with the addition of the Sobel edge product, this simpler enhancement is nearly as instructive as the more involved principal components method. It has the advantage of not depending on the water vapour image product, but suffers from lower textural information content and less representativeness of upper level flow patterns.

A final visualisation enhancement was produced as a hybrid method between the principal components and IR pseudo–colour methods. The IR product is generated, and the resulting image is transformed from the display capable *Red, Green, Blue* (RGB) colour space to the more human interpretable *Intensity, Hue, Saturation* (IHS) colour space ([22]). Intensity represents the luminance of the product from black through pure white. The hue and saturation values code for pixel colour. By replacing the high information content intensity channel with the principal component product, colours from the IR method are merged with the jet level patterns and higher textural information from the principal component product. A backwards transformation to the RGB colour space allows the final product to be shown on the display. In all instances where the IR and WV channel are at least partially uncorrelated (*i.e.* not identical) the principal component transform ensures that component one of the transform contains more information than the IR channel alone. While both theory and visual assessment suggest this product represents the single best overall representation in the methods discussed, it is computationally intensive, and in most cases not required for a suitable analysis; however, in difficult cases it may yield insights into the developing patterns that might be missed from the other products. An example of these enhancements is shown in Figure 4.1

4.2 Registration Strategies and Geometric Analysis of Development

Discussion with staff at Pacific Weather Centre (cf. 3.1) made it clear that PROBE involved careful observation of image loops of development. Visual enhancements allowed tie or control points to be identified in the cloud field, leading to the exploration of methods to geometrically control, and hence describe, deformation patterns in the developing systems. Comparison between changing meteorological fields and the image sequence helped to determine which features were characteristic of development. Published Information (*e.g.* [46], [31]) assisted in determining analysis strategies. Progressing from basic to more advanced strategies, the following methods

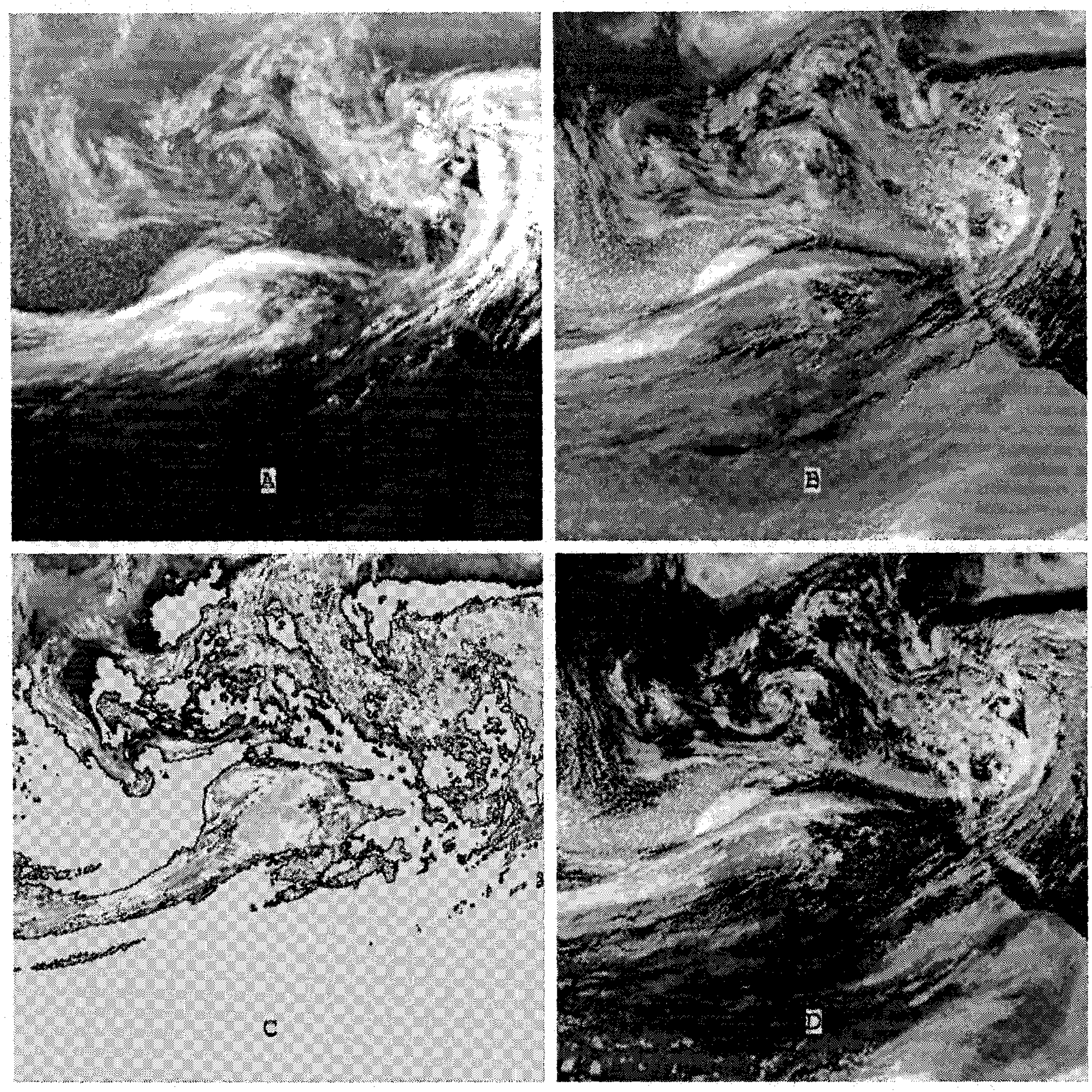


Fig. 4.1. Enhancements used for storm feature detection: *A*–raw IR, range inverted: white=cold cloud; *B*–IR & WV principal component one; *C*–IR pseudo–colour with edge vectors and background mask; *D*–principal component and IR colour fusion.

were developed: Manual Registration, Cloud Frame Rotation, Principal Components on Bitmap Axes, and Idealised Grid Tracking. Based on the development trials, of the four methods, only manual registration appeared incapable of capturing useful information about the system. The last three methods appeared progressively more capable; however, each had its own particular advantages.

4.2.1 Manual registration

The most obvious way to attempt geometric rectification is to apply traditional land–based registration methods. An operator defines a series of "ground control" or "tie" points between frames F_n and F_{n+1} of a series. As points are entered, the root mean square (RMS) error is computed for a given geometric transformation model. RMS error is a resampling measure similar to the variance, which describes how closely the transform model projects tie points of $F_{n+1} \Rightarrow F_n$. We can compute RMS as $\sqrt{(Fx_{n+1} - Fx_n)^2 + (Fy_{n+1} - Fy_{n+1})^2}$ where Fx and Fy give the x and y coordinates in the two frames (*c.f.* [17], pp. 104–105). A low RMS error *can* indicate a good model fit – or it might only indicate that the model fits the control points well. It is advisable to split the points into a "model making" and "model testing" group, to see if the transform also indicates a low RMS error for independently chosen points.

Since deformations along the jet stream and the comma head are dissimilar, a higher than first order (non–affine) non–linear regression of tie points is suggested to capture the essential deformations. Control points are viewed as *estimators* of the global frame deformations, and so should capture the overall nature of those deformations. The higher the model order, the better the model fits local distortions about the control points, although its generalisability may become problematic. Areas not near any control points may tend to produce unrealistic transformation results. Typically for ground–based methods, control points are chosen that have strong edge contrast and are readily identifiable from frame to frame; however, such points have proven to be very difficult to reliably locate in cloud imagery due to both the complex dynamic evolution of the images and the lack of sharply demarcated features.

4.2.2 Cloud frame rotation

Viewed from the perspective of the entire system (comma head and tail) cyclonic development can be seen to proceed with three major deformations: translation, rotation and scaling. Translation can be fairly easily rectified in a frame to frame sequence, since translation relative to the storm centre can be defined by the position of the surface level low pressure centre. The upper level vorticity maximum would be another possible tracking point; however, as it can be readily identified from satellite images. The location of the surface low is not always available on surface charts; however, an approximate position for the surface low can be estimated by looking for the intersection of the comma head and the upper level jet flow (*c.f.* [46], pp. 21–31).

Rotational changes are more difficult to assess, requiring the positioning of an axis in each frame. Further, it becomes apparent that deformation along the jet stream axis, from herein denoted the *principal axis*, is different from that occurring in the expending comma head region, particularly after upper level decoupling of the jet and surface cyclone (see Chapter 2). To completely capture these changes, an affine, first order transform will not suffice. In the method being described only rotation relative to the the principal axis (its tilt) is measured, not the rotation of the comma head relative to the jet stream itself.

To test this simple method a circular mask was applied about the surface chart low centres of five frames, at six hour intervals, from the April 1996 sequence after centering the mask about the surface low (manually controlling for the translation of the low). The mask reduces the influence of extraneous clouds. Since clouds in cyclonic systems tend to rotate counter–clockwise (cyclonically), an algorithm was developed to "unrotate" subsequent frames and compare them to the previous frames ($F_{n+1} \Rightarrow F_n$) by applying a positive rotation. After projecting frame F_{n+1} by a given degree rotation, correlations were computed between those frames. Correlation values should be highest when frames separated by a small time index are rotated such that their principal axes mostly superimpose. Table 4.2 shows correlation values for five raw frames covering the major development period of the April 1996 event in

Table 4.2
Correlations between frames from April 24-25, 1996. Time index is for UTC (Zulu) time.

Time	#	1	2	3	4	5
18Z24	1	1	0.61	0.34	0.29	0.34
00Z25	2	...	1	0.64	0.43	0.49
06Z25	3	...	...	1	0.64	0.51
12Z25	4	...	...	...	1	0.66
18Z25	5	...	...	...	...	1

six hour intervals. Results for separations of one frame after rotation are shown in Figure 4.2. Correlation values drop as frame time separation increases (not shown) due to strongly different geometry between them. The curves also become flatter with a less definite maximum value.

The angle of maximum correlation for the same five frames at all time separations (multiples of six hours in this example) are shown in Table 4.3. In this comparison, a given frame F_n is not only compared to F_{n+1} (*i.e.* to the frame six hours later), but to $F_{n+2...}$ etc. as well, similar to values in the non–adjacent columns of Table 4.2. There does not seem to be any systematic relationship for how correlations change as frame separation increases, although Figure 4.2 does seem to suggest a localised maximum for relatively short separations (6 hours in these examples). It is perhaps not surprising that as frame separations become large, rotating one frame relative to the other does not produce a meaningful superimposition, as the frames would then record greatly different stages of development in the system. A larger sample set, or a choice of a smaller frame to frame time separation would be needed to test the problem further. Such testing is beyond the scope of the current work.

Another similar approach is to generate a bitmap from the segmented mid and high level cloud in each frame, and to rotate one bitmap with respect to the previous one ($B_{i+1} \Rightarrow B_i$). Under bitmaps, textural features are removed, and only the overall

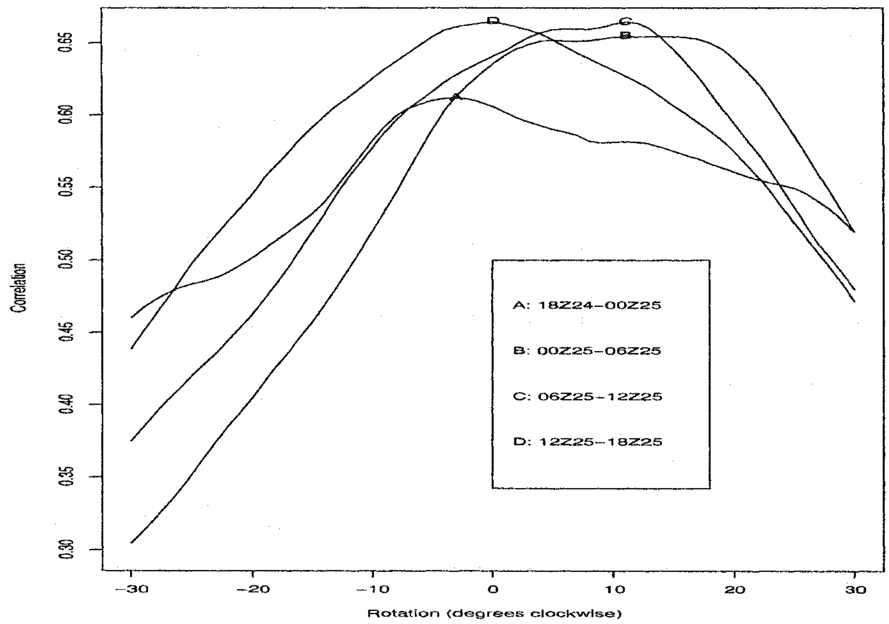


Fig. 4.2. Correlation curves as frames F_{n+1} are rotated against F_n.

Table 4.3
Angle of maximum correlation under rotation of $F_{n+1} \Rightarrow F_n$.

Frame	00Z25	06Z25	12Z25	18Z25
18Z24	-3	30	-15	-12
00Z25	X	11	10	-5
06Z25	X	X	11	-1
12Z25	X	X	X	0

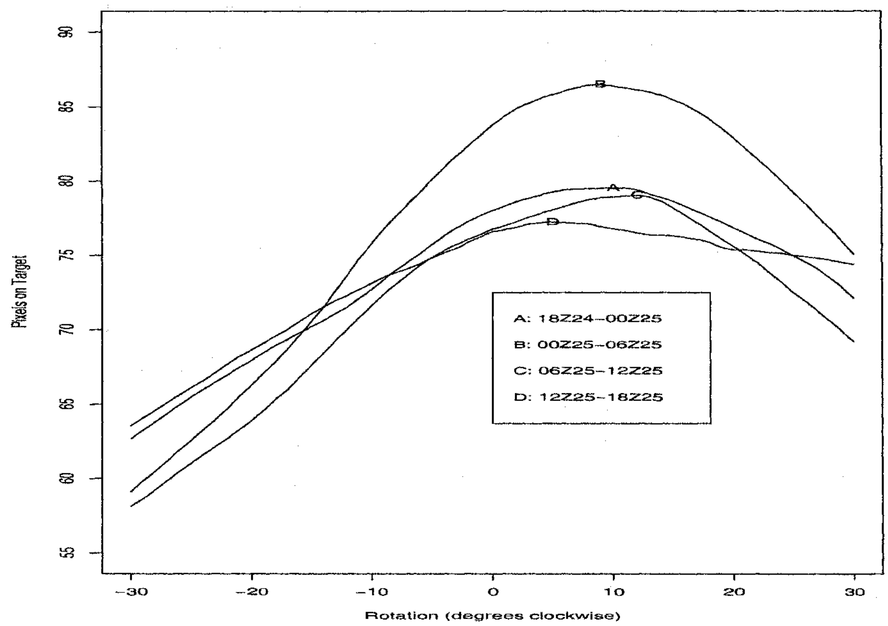


Fig. 4.3. Percent "pixels on target" of B_{i+1} rotated onto B_i.

shape of the cloud–mass is compared between frames. Some unrelated remnant and downstream jet–related cloud are left in the segmented images. If those areas are manually removed in an image editor, a mask area is created that represents the extent and shape of development in that particular image frame. A suitable metric for comparison of these shapes under rotation is the "pixels on target" measure: a count or percentage of on pixels in bitmap two that overlay on pixels in bitmap one – $POT = \frac{\Sigma(B_i * B_{i+1})}{\Sigma B_i} * 100[\%]$, where $B_i * B_{i+1}$ defines a grid multiplication ($B_{i(x,y)} * B_{i+1(x,y)}$ for all (x, y)pairs), and ΣB_i is the count of all on pixels in frame B_i. Since bitmaps are either on (value of one) or off (value of zero) their product defines their logical intersection. Results for the same five frames as before are shown in Figure 4.3. Use of bitmaps seems to produce cleaner curves with more definite maxima.

Correlation between frames and pixels on target of bitmaps offer objective, simplistic measures of development. The only subjective factors in their calculation are the location of the surface low, which can be done with a fair degree of accuracy, and the removal of surrounding cloud masses for bitmaps, which is a straight–forward procedure. Both methods deal with translation and rotation fairly well (if only based on manual placement of the low), although in this sample, results for comparisons B, C and D correspond, while those for comparison A do not (comparing the curves in Figures 4.2 and 4.3). Unfortunately neither method implicitly accounts for stretching along the principal axis.

4.2.3 Principal components on bitmap axes

Once bitmap shapes of flow development are isolated, as described above, these shapes can be automatically aligned according to their principal axes and scaled according to their length, resulting in a fairly good measure of development. Such a procedure is suggested in Gonzalez and Woods ([11], pp. 151–152), and is often used in rotationally invariant object recognition procedures. Given a vector $\vec{V} = \begin{bmatrix} x \\ y \end{bmatrix}$ such that x and y are the (x, y) coordinates of the contour points of the bitmap shapes, then principal components one and two of $\vec{V}$ will form new coordinate axes with the following properties: P_1 will contain the majority of the variance; $P_2 \perp P_1$ and therefore uncorrelated with P_1; P_1 explains more of the pooled variance than any other single axis that could be chosen in the dataspace. In the context of an image bitmap, P_1 will be the axis along which the greatest spread of points will be found if all points are projected onto that axis, which corresponds to the longest axis in the bitmap pattern. The orthogonality constraint ensures that (P_1, P_2) is a suitable substitute coordinate space for (x, y). For early stages of development, the longest visual feature is bound to be the jet stream cloud trace, and thus P_1 should correspond fairly well to the principal axis. Furthermore, the eigenvectors of the components specify the linear axis transform that defines the new components. The transform in turn describes translation, rotation and scaling of the entire bitmap. Such transforms were run on test bitmaps extracted from the mid and high cloud edge detection segmentation (Figure 4.4). The bitmaps represent the contour of the target region. They were converted to vector format for input to SPlus using a raster–

to–vector conversion module. Unfortunately, the principal component transform is strictly linear, and so like the simpler bitmap rotation method cannot account for differential deformation between the jet stream and comma head.

4.2.4 Idealised grid tracking

Characteristic features of a developing cyclone are the baroclinic leaf, expanding comma head and dry slot region behind the surface low (see Chapter 2). All three features are observable to different degrees under the various enhancements presented in Section 4.1. From the onset of rapid deepening until the dry slot becomes sufficiently pronounced to create a pocket of cloud–free air inside the comma head, an idealised wire–figure can be used to represent the system. The "comma grid" used in this work underwent 10 distinct forms before one was found that provided a satisfactory representation of the developing comma head from the April 1996 test sequence. Figure 4.5 shows grid 10; the final version.

The grid was created by specifying geometric relationships between its parts and computing the resulting coordinates in a spreadsheet. Segment $\overrightarrow{AC}$ represents a linear portion of the jet stream from underneath which the comma head emerges, and is denoted the *principal axis.* If $\overrightarrow{AC}$ is one unit length, segment $\overrightarrow{AB}= 3/5$ and $\overrightarrow{BC}= 2/5$ units. There are five "tick" alignment marks along segment $\overrightarrow{AB}$ and five within the comma head between B and C (with an additional two at B and C). The head itself is the segment of a circle of radius R who's centre is located beneath point 8, such that the distance between the centre and point 8 is $2/3R$, where the circle is secant to the principal axis. From these criteria, we find that $R = \frac{\overrightarrow{BC}}{\sqrt{5}}$. Comb–like teeth extend from points 5, 7, 9 and 11 for a length R perpendicular to and below the principal axis. Radial lines from the centre along points 6 through 10 extend beyond those points for a total length of R from the centre. The teeth and radial lines are used to specify the jet–stream width and comma head position relative to the principal axis.

Alignment of the grid to a development frame is guided for enhanced imagery (without use of surface charts) by the following steps:

A Estimation is made of the position of the surface low. Attention is paid to a

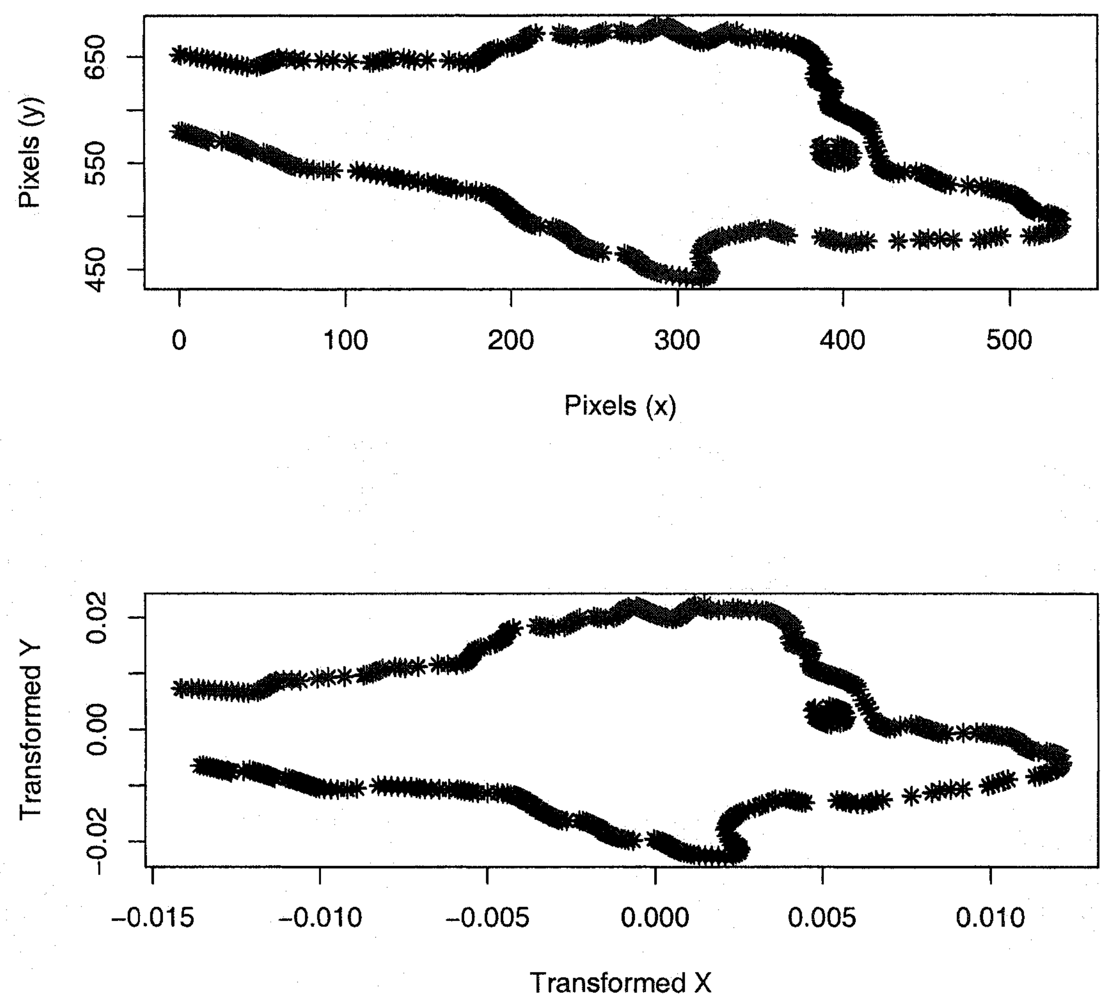


Fig. 4.4. Vector outline of development and its principal components alignment.

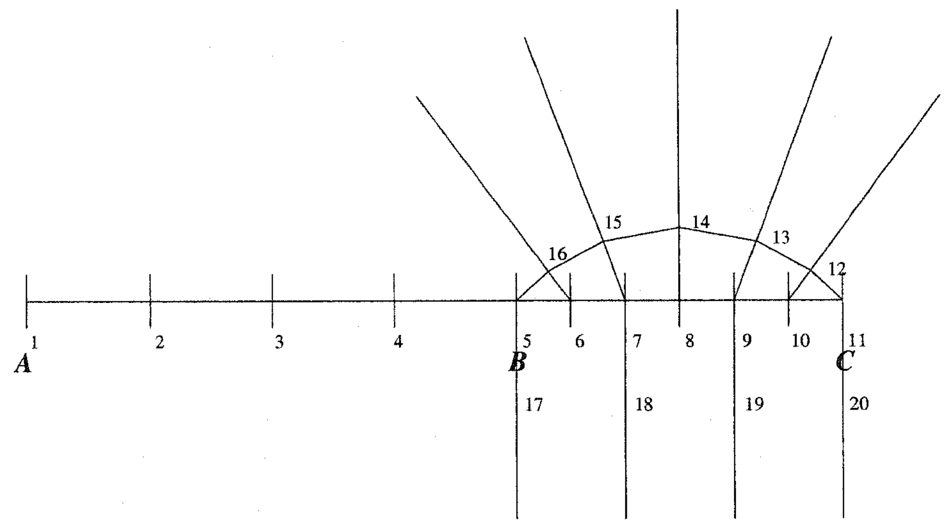


Fig. 4.5. Alignment grid #10 for flow rectification. Numbers in the figure represent tie points used in specifying development of a system when the grid is superimposed on an image frame.

developing dry slot at the south or south–western edge of the comma head. Point #5 of the grid is positioned at this location.

B Using the stretch function, point #11 is grabbed and dragged to the far edge (north or north–east) of the comma head. The principal axis should be fitted as per a regression line through the brightest westernmost extent of the jet stream flow. The entire grid will rotate and stretch to accommodate the line segment between the surface low and edge of the comma head. Awareness is necessary of splitting cloud masses that are entrained by the low or the jet stream but that will split off in subsequent frames. The position of the gradient enhancement bitmap edges can be used for guidance.

C Comma head expansion points are added along the radial lines through grid points 12 through 16. Focus is on the general shape of the expansion, rather than the exact point where the head crosses the radial lines. As for step two, the process is like fitting a non–linear regression or spline through a semi–elliptical region. Also as in step two, awareness of splitting and merging cloud masses and the gradient bitmap location are necessary.

D Points 17 through 20 are used to specify the mean cloud width. Alignment is chosen to correspond to the opposite side of the bright upper–level mass used as the principal axis, and should approximate the mean width of this baroclinic shield cloud feature. From the April 1996 test case, cloud width was a very well defined and easily identifiable feature. In subsequent images, it was found that this feature is not reliable for all development patterns. In the April 1996 test case it does denote usable information, so determination of this feature will have to be done on a case by case basis.

4.3 Radiometric Analysis of Development

Early parts of this work focused on the changes occurring in a system at the radiometric level. In order to examine pixel level differences in sensor response, it was first necessary to control for, or remove, the geometric deformations occurring over the interval being investigated. The grid method presented above was used to co–register the entire sequence of images in the development case. In this manner, similar regions of the system were forced to geometrically overlap, so that pixel changed could be assessed frame to frame. The goal of these methods was to determine if radiometric change in specific parts of the system were indicative of development, and ultimately, if these changes were characteristic of explosive events. Some patterns, such as banding in and around the comma head, were immediately visually apparent. These visual features are also depicted schematically in Figure 2.2.

After image registration was performed, a series of methods were applied to try to quantify distinct visual patterns. Clustering of local textures as measured from grey level co-occurrence matrices was performed (cf. [11], [17]). Areas of like–texture were then analysed using the *Fragstats* suite of landscape ecology software [25], to produce and analyse shape metrics at the patch, class and landscape (or "cloudscape") level. Preliminary results suggested that texture scaling (the size of characteristic brightness changes) was an important problem, even in a series of geometrically controlled images. Some work with multiscale Wavelet and Fractal analysis was done, but the entire section of radiometric work was dropped due to the significant problems scaling posed. Future work in this area may prove fruitful, as important visual information seemed to be contained in the changing textures, but suitable methods of measuring those changes were not found. Some final comments on possibilities for future work in this area are made in Section 7.3.

4.4 Post–Processing of Raw Results

Basic radiometric results from high cloud and the Cyclonic Index and geometric results from the grid method were applied to five full storm cases. The raw results contained a large volume of data, which needed to be ordered in a meaningful way. Details of the three primary analytical methods applied to each case are presented below. These include: pressure changes; radiometric analysis; and vector analysis. Some discussion of the PROBE methodology is also included.

4.4.1 Probe

The PROBE system, as described in Chapter 1, was based primarily on traditional meteorological analysis methods. It was heavily based on the work of Roger Weldon ("Cloud Patterns and the Upper Air Wind Field" [46]). Although PROBE could not be implemented here due to lack of the program code and systematic upper level charts, several of the concepts addressed in PROBE are also addressed in the current work. In particular, both methodologies address: rotation of the Jet Stream; sharpening cloud edges; cloud field expansion; and translation velocity.

4.4.2 Pressure

The primary data sources for this work were 1000 hPa (surface level) pressure charts and GOES–West satellite image coverage. Pressure charts were obtained from both the Pacific Weather Centre in photocopy form and digitally from the Canadian Meteorological Centre. Satellite imagery was provided by the Pacific Weather Centre. Ideally, pressure centres could have been directly compared to image–derived measures, and in some instances this was possible; however, coverage did not always overlap in time, and was not always available at regular intervals. Older records from PWC were often incomplete, and pressure charts were only available at 6 hour intervals, compared to half–hour intervals for compressed imagery (the uncompressed, full resolution images used in the development case were available at one–hour intervals). Due to these temporal differences, spline interpolation was used in several measures, and especially importantly for pressure values, to regularise observation intervals. The natural spline method from the R statistical package was used (See Chapter 3 for more details of the R package).

As is standard practice in meteorological work, all time indices were recorded relative to *Coordinated Universal Time*, also referred to as *Zulu Time*. Furthermore, pressure intervals were indicated relative to *Time Zero* (t0), as defined in Chapter 2. Standardising observations temporally allowed comparisons of development between as well as within cases.

Pressure falls alone do not provide sufficient information to charactarise a rapid deepener. As mentioned it Chapter 2, system strength, as measured by the maximum deepening value, and especially system intensity, as measured by maximum rate of deepening, are more informative measures. A standard index is the Bergeron, a value indicating the rate of pressure fall corrected for the effects of latitude. Although various authors choose different standard latitudes, in this work, the criteria for a *Bomb* of one hPa deepening per hour at 60° north was used, as per Sanders and Gyakum [39].

To standardise between–case results, all pairs of pressure observations were compared per system, and the best 6, 12 and 24 hour Bergeron values were computed. To account for occasional gaps in the pressure data, the best interpolated 6 hour Bergeron value was also computed (indicated as *6i* in the Bergeron value tables).

4.4.3 Radiometric Response

Response values of the pixels in each image were also analysed. Initial work on the project focused on radiometric values, but it was found that geometric changes were easier to analyse and appeared to be more informative; however, some basic radiometric results were retained. Two basic measures from the radiometric response were developed: high cloud counts; and the "Cyclonic Index" (C–language code is shown for computing these values in Appendix A-3. High cloud count is simply the total of pixels in each image frame that are -30° Celsius or colder, after masking out areas over land. Preliminary work suggested a correlation between high cloud and regional pressure (see Chapter 3). The cyclonic index was developed to find mature, large cyclones in a series of images. It has a high value when a solid, circular area of high cloud is surrounded by patches arranged in a ring, of cloud–free or low cloud areas. A high contrast between the centre and surrounding patches yields a high index value. A final radiometric measure was the Sobel edge sharpness gradient (described earlier in this chapter). Notes and comments from PWC staff suggested that edge sharpening of clouds might correspond to development ([27], Personal communication with Laurie Neil, 1998. See Section 3.1). As cloud edges sharpen, they become more visually distinct. Values of the absolute Sobel gradient represent the strength of edge transition features in the cloud image, allowing for quantification of the sharpness of the cloud edges.

4.4.4 Vector Calculations

Raw geometric measures were derived from the position of the placed idealised comma grid from each satellite image frame (the grid is described in Section 4.2.4). All grid vector points have been converted from screen units (pixels) to a polar stereographic projection with true latitude at 60° north to match the projection of the pressure charts. Distances were converted to latitude corrected kilometres across the great circle distance between points using the IDL procedure *map_2points* (see Chapter 3 for information on IDL). The first geometric measure is *Jet Tilt*, representing turning of the principal grid axis aligned with the upper level flow. Next is the *Comma "Bulge" Radius*, the mean length of arcs from the centre of the comma head to the location of the outer comma cloud edge, following the radial grid lines. *Comma Head Length* is the distance between the southern point where the comma head first appears below the upper level flow to the northern extent of the comma head along the jet line (principal axis). Finally, *Baroclinic Cloud Width* attempts to track the mean width of the upper level jet baroclinic zone along the grid principal axis, although in practice this measure has been found to be difficult to estimate.

First and second order derivatives of the raw measures were initially calculated from interpolated one–half hour interval values. These values represent the velocity and acceleration of geometric changes in the system. Due to the observation interval differences between image–based and chart–based data source, these results were found to require further processing. In particular, raw image velocity and acceleration graphs were extremely variable, possibly due to the geographic registration performed on the imagery. Since GOES orbits the equator, pixels recorded obliquely will have significant geometric distortion. While these pixels are normalised in post–processing, the spatial accuracy of pixel centres will vary across the image scene. To mitigate some of these problems, the raw values were subsequently smoothed. After computation of velocity, these values were passed through the *R* package "Lowess" smoothing function (see Chapter 3 for details of the R package) with an f–parameter of 1/6. Lowess was too harsh a function for the lesser variability of the acceleration plots; instead of smoothing with Lowess, in the splining process, these values were estimated over a three–hour interval.

4.5 Statistical Treatment

Since raw data in this study was derived from charts and images that did not always have systematic temporal coverage, a good deal of pre-processing was required before statistical analysis could be performed. In Chapter 5, data with extensive regular interval splining is presented, which is quite appropriate for display purposes; however, for numerical analysis, splining artificially increases the number of observations, and so also increases the statistical power and inflates the Type I error rate (the probability of concluding we have found an effect when we really have found a trivial difference) in statistical inference.

Most possible pairs of variables were compared with each other. Specifically, all variables were paired with their time index, all within–case variable combinations were paired, and all equivalent variables were paired between cases. Between case comparisons of different variables were omitted as being of secondary value, thus removing many extra comparisons from the set.

Treatment of each pair of variables was done to normalise time indices, and so allow for paired comparisons without introducing a large number of extra splined observation values. For each XY pair, X effectively becomes the explanatory (independent) variable, and Y the response (dependent) variable. Values for Y were derived by splining raw Y values to the time intervals in X, with splining ensuring original values occurring at the correct time were retained. Not available (NA) values were omitted, as were time indices extrapolated beyond those of the original Y indices. Where time indices in X were missing from key Phase times that were to be explored, these values were introduced through splining as appropriate. The splining allowed correlation and regression models to be computed.

Comparisons between variables were done at two levels: case level comparisons considered the values of the variables across the entire temporal range of the case (a comparison of summary statistics), while observation level comparisons looked at the variables at each time index. A variation of case level analysis was to consider values over the Phase II to Phase III interval, as estimated from the images (see Chapter 5). The purpose of this time standardisation was to look more closely at changes over a readily identifiable interval. The statistical results are presented at the end of Chapter 5.

4.6 Final Remarks

Three useful motion estimation and registration strategies were presented in this chapter. Their original goal was to allow for geometric standardisation between frames of a development sequence. Such a standardisation would allow for the direct geometric superimposition of one frame in the sequence over the previous frame, hence permitting a pixel based comparison of radiometric changes: $F_{n+1(x,y)} \Rightarrow F_{n(x,y)}$ for each (x, y) pixel pair in frames n and $n+1$. The "unrotation" methods (correlation and pixels or target based) do not control for stretching. Both the principal components and grid methods do control stretching, but the principal components method is sensitive to small variations in the shape of the extracted cloud segments. Overall, the grid method was felt to be the most accurate at geometric representation of development, and was used extensively in preliminary testing of radiometric change analysis. Radiometric analysis has proved to be a topic complex enough to warrant a completely separate study; meteorologists traditionally focus on geometric changes in video loops, so the project was re–focused on geometric changes, while retaining only the most basic radiometric analyses. All results are presented in Chapter 5. While the grid method was the most suitable under the circumstances, it is inherently more operator intensive, and less objective than the other methods. Some comments regarding potential use of these methods in automated or quasi–automated analysis are made in the conclusions of this work.

5. RESULTS

Raw results produced by application of the methods described in the previous chapter are presented here. First, data in chart form are presented in light of the four phase model of development described in Chapter 2. Statistical analysis of these data are then presented at the end of the chapter, with special emphasis on variables that predict for the system strength and intensity.

5.1 Introduction

Results from five Pacific storms are presented here in a case–study format. By this means the assembled measurements can be compared and contrasted to reveal trends, patterns and correlations both for different measures within a case and for the same measures between cases. Subsections of this Chapter will deal with each type of analysis done. Results have focused primarily on geometric analysis, as indicated at the end of Chapter 4. Detailed code used for computations, written in the *R* statistical language are presented in Appendix A-4. Results for the development case will be presented in each section in full, followed by a brief presentation of the other cases. All tables and figures appear at the end of Section 5.2 starting on page 73.

As suggested in Chapters 3 and 4, a primary focus of this work was the development of experimental methodologies to track and analyse explosive deepeners from satellite images. A full–resolution (uncompressed) image sequence from April 1996 was used as the developmental case. This storm was selected, with the help of Laurie Neil at Pacific Weather Centre (See Section 3.1), as being very typical of the type of winter storms that develop into explosive deepeners. Under sub–headings following the pattern presented in the methodology description above, results will be presented in detail for the developmental case. Results for the remaining four cases will be presented after those for the development case and discussed under each sub–heading.

5.2 Case Descriptions

Case I: APRIL 24, 1996 DEVELOPMENT CASE

More support information was available for the January 1996 development case than for any of the other cases. Table 5.1 shows the available chart and image data series for this case. A contact sheet at 6 hour intervals is shown in Figure 5.1. Image coverage for Case I was at one hour intervals, and covered the entire development period, from the emergence of the comma head to t0, when the dry tongue became evident. Flow was initially quite zonal, but became more meridional as the system developed.

Case II: JANUARY 12, 1997

This system was quite meridional with entrained cirrus to the west of the low that does not define the comma–head.

Case III: JANUARY 28, 1997

This system began upstream of an earlier low, which failed to develop. The cloud mass that becomes the comma head was initially separate from the jet, but merged with it in later frames. The system was fairly meridional, with the jet in a mostly north–east orientation.

Case IV: NOVEMBER 8, 1997

This case developed upstream of a cloud mass that never developed into a true comma head. Much of the cloud from this system was later entrained by the developing system. In later frames, the system finally tilted past the north–south line. Another downstream system began to form along the jet as this system intensified.

Case V: MARCH 6, 1998

The upstream jet in this case was extremely meridional, tilting well past the north–south line. Cloud about the developing low was very dense and thick, while the width of the jet in other areas was much less. In early frames it was not apparent where the low would develop.

5.2.1 Pressure

Case I:

Case I was a fairly strong system, deepening to an approximate lowest value of 969 hPa (Figure 5.2*A*). Sea–surface pressure measured relative to 1000 hPa indicates a total pressure fall of 31 hPa. The storm was also quite intense, with a 24 hour–based Bergeron value of 1.34 (see Table 5.2). The 6 hour value was 2.71, about twice as intense, while the 12 hour value was 1.38, which is similar to the 24 hour value. Both the vector and pressure centre tracks follow closely, with the vector positions falling to the north–west of their equivalent pressure centre positions. Separations between the observations are generally less than a few degrees. The storm originates within Roebber's [36] maximum frequency zone for storm development. It reaches it's maximum depth just before leaving the 50°north limit for bomb decay (Figure 5.3).

The contact sheet for this case (Figure 5.1) shows a strong cyclonic shape, with a sequence progressing from a well defined baroclinic leaf (Phase I) to an emergent comma cloud pattern (Phase II), the full development of a comma–head (Phase III), and leading to a fully secluded cold core in the "spin down" phase (Phase IV). In Chapter 2 it was postulated that there is a direct connection between the four phases of the physical cyclone model of Rogers and Bosart [37], and the cloud field descriptive model of Neiman and Shapiro [31].

Based on the image data, the Neiman and Shapiro model can be readily applied to Case I. Phase I began with the appearance of a "leaf" structure at 0Z April, 24, or at t-22 hours relative to the emergence of the comma head (where the emergence is taken as an estimate of t0; the beginning of explosive deepening). Phase II began at t-10 hours with the cold front "fracturing"; the first appearance of the early structure of the comma head (Figure 5.1*A*-*B*). This point corresponds with the first applied "comma–grid" vector for the image tracking method. Phase III began at t0, when the comma head pattern first separates from the baroclinic leaf cloud mass (Figure 5.1*C*-*D*). Neiman and Shapiro [31], define Phase III as the emergence of a t–bone comma head along the occluded front west of the frontal triple point, and by definition, t0 in this work represents the same point in time. It also represents the last analysed image frame. Phase IV began at t+14 hours (Figure 5.1*E*-*F*). It represents the point where the frontal lines have completely encircled the surface low, cutting it off from the cold air mass and leading to decay of the system.

The Rogers and Bosart [37] physical model is also fairly well followed by the data. Its Phase II requirement is for a minimum 12 hPa deepening over a 12 hour period. Central pressures for Case I at the start of each phase were: 1002, 996, 986 (interpolated), and 970 hPa. Calculated at a per hour rate, the deepening between the starting pressures of each phase was: -0.50, -1.00, -1.14 and 0.33 hPa/hr, which match expectations except for Phase III, where pressure should be quasi–steady. The Phase IV rate also uses a pressure value at t+20. This is the last available pressure value; during Phase IV, the system should weaken until dissipation. There were three periods in this case that met the Phase II requirements: t-10 to t+2 at a ΔP of 12 hPa, and t+2 to t+14 and t+8 to t+20, at ΔP of 14 and 12 hPa. The case is unusual in that there is a quasi–static pressure state at 984 hPa at t+2 and t+8, which is a requirement if Phase III, but this period is followed by a second period of strong pressure falls. Phase IV would appear to have begun by t+20 as pressure is increasing at this time. Phase I is said to begin at the point when the surface low is first visible on surface charts. Therefore it cannot be compared to an image–based model. Comparing the remaining three phases of the two models yields for the Neiman and Shapiro image model with the Rogers and Bosart physical model times following in brackets: Phase II at t-10 (t-10, t+2 or t+8); Phase III at t0 (t+2); and Phase IV at t+14 (t+20). The pressure derivative graphs emphasize the deceleration of pressure falls about Phases III and IV (Figure 5.2*B*).

Figure 5.4*A* and *B* show velocity and acceleration calculations for the storm centres as tracked from the pressure charts and from the image sequence. The shape of both velocity and acceleration curves is similar between those from the pressure charts and those from the images. The image curves are much more variable, which is not unexpected since they are derived at half hour intervals instead of the six hour intervals for pressure chart centres, and because image centres are not estimated based on a smooth pressure field. At t-10, which corresponds to the image–based estimate for Phase II, velocity is low, but the storm centre is accelerating. At approximately t-3 velocity is at a maximum, but decreases again by Phase III at t0. There is a local minima of velocity at t+14, corresponding to Phase IV. The deepest pressure occurs at t+18, only four hours later.

The system ground track is shown in Figure 5.4*C* and *D* for chart and image centres respectively.

Case II:

Available data for Case II are listed in Table 5.3. Case II's chart and image ground tracks follow each other well, but are separated by several degrees in the early stages (Figure 5.5*A*). The system did not originate in Roebber's [36] zone, and maximum deepening occurred just north of 50 °. This system had quite a complicated cloud field, as seen in Figure 5.7. It is possible the early grid alignment from the image sequence has some systematic error, later resolved as the development became more clear, or that there were difficulties in placing the early storm centres on the surface charts. The case had a 24 hour Bergeron value of 1.19, qualifying it as a bomb (Table 5.4). The 12 and 6 hour values were 2.06 and 3.09, or 1.73 and 2.3 times the 24 hour value respectively. The lowest pressure value (Figure 5.8) was approximately 961 hPa (interpolated). Hourly deepening rates for Phases I to IV were -0.86, -1.86, -0.28 and -0.72 hPa/h, which roughly follows the physical model, except for continued deepening in Phase IV. There were three periods of 12 hr separation that met the Phase II criteria: t-14 to t-2 at a ΔP of 17 hPa; t-8 to t+4 at a ΔP of 15 hPa; and t+16 to t+28 at a ΔP of 14 hPa. Central pressure was quasi-steady at approximately 981 hPa between t-2 to t+16 hours, giving a time for Phase III, but this period precedes the lowest recorded pressures and the second period of strong deepening. Model comparisons for Neiman and Shapiro (Rogers and Bosart) yield: Phase II at t-7 (t-14, t-8 or t+16); and Phase III at t0 (t-2). Figure 5.8 shows that deepening was fastest in the t-14 period. Phase IV values could not be estimated since later pressure values were not available.

Case III:

Available data for Case III are shown in Table 5.5. Case III occurred in a very complicated flow pattern (See Figure 5.9). Several strong, quasi–stationary lows were present throughout the sequence, and it was difficult to locate the actual system. Some guess work was involved in placing the low for the early pressure chart observations, as there was no low indicated on the charts in the area suggested from the imagery. Later observations were more definite, and show a few degrees separation between chart and image centres (Figure 5.5). It originated slightly South–East of Roebber's [36] zone and was at 998 hPa a few degrees before the 50°N line. The only available Bergeron value estimate was 1.65 based on the 6 hour interval (Table 5.6). Insufficient pressure data were available to make a 24 hr Bergeron measure, as the pressure records only spanned 18 hours (mostly since there was no surface low to track in the early stages of development as visible on the imagery). The lowest pressure value occurred at t-2 at 993 hPa (Figure 5.8–*1*), which is inconsistent with t0 representing the onset of maximum deepening. Given its shallow pressure profile it is very likely that Case III was not a Bomb, but a "Dud"–a case that appears to be developing explosively but does not. This conclusion follows from the cloud–field patterns similar to those of the other cases but a lowest pressure and Bergeron value suggesting the system was not strong or intense enough to be a *Bomb*. Very likely no dynamic forcing occurred. The hourly pressure fall could not be calculated for Phase I, since the first pressure observation does not precede the first image observation (it was recorded after the beginning of Phase II), nor could it be calculated for Phase IV, since the last pressure observation precedes the beginning of Phase III. The rates for phases II and III are: -0.63, and -0.19 hPa/h.

Case IV:

Available data for Case IV are shown in Table 5.7. Flow patterns for Case IV are shown in Figure 5.10. This case originates at about 40°N, and 10° East of Roebber's [36] zone (Figure 5.5). Maximum deepening occurs at 53.4°N. Bergeron values for Case IV (Table 5.8) were 1.04 (interpolated at 24 hr), 1.86 (12 hr) and 2.08 (6 hr). The 12 hr and 6 hr calculations are 1.79 and 2 times the 24 hr value, respectively. The system deepened to 954 hPa (Figure 5.8–*1*) at t+30. An hourly rate could not be calculated for Phase I due to lack of early pressure coverage. Rates for Phases II through IV were: -4.22, -1.71 and -0.60 hPa/h, which show stronger than expected deepening in Phases III and IV. There were two candidate 12 hr periods for the start of Phase II: t+12 to t+24 at 19 hPa, and t+18 to t+30 at 16 hPa; however, early pressure data was not available, so this phase may have begun earlier. From Figure 5.8–*2*, it appears more likely that t+12 is the beginning of Phase II, although these two ranges are possibly both part of the same deepening phase. Phase III appears to have occurred between t+30 to t+42 with a pressure of approximately 956 hPa, compared to the image–based time of t0, but lack of early data makes this hard to determine.

Case V:

Available data for Case V are shown in Table 5.9. The flow pattern for Case V is shown in Figure 5.11. The case began at 35°N, about 7°east of Roebber's [36] zone (Figure 5.5). Maximum deepening occurred just south of 50°N. The 24, 12 and 6 hour Bergeron values (Table 5.10) were: 1.45, 2.05, 2.90 and 3.03 (6 hour interpolated). Shorter estimation periods resulted in over–estimation of intensity, and the interpolated 6 hr value exceeds the best recorded 6 hr value. The 12, 6 and 6 hr interpolated values were 1.42, 2, and 2.09 times the 24 hr value. The system deepened to 982 hPa (Figure 5.8–*1*) at t+33.5 and qualifies as a bomb. Phase I hourly deepening could not be calculated due to lack of early pressure data. Phase II to IV rates were: -0.80, -1.20, and 0.11 respectively. The strong deepening in Phase III is unexpected. There were two intervals that qualify for pressure based estimates of Phase II: t+3.5 to t+15.5 at 18 hPa and t+9.5 to t+21.5 at 17 hPa; from Figure 5.8–*2*, t+3.5 appears the more likely candidate. Phase III may have occurred between t+21.5 to t+27.5 with a quasi–steady pressure of 984 hPa; the lowest pressure value of 982 hPa at t+33.5 may also have been part of this period. The start of Phase II may be somewhat off since the first available pressure value was recorded 3.7 hours after the image estimate of Phase II's start. Comparisons of the image (physical) models yields: Phase II at t-6.2 (t+3.5 or t+9.5); Phase III at t0 (t+21.5); Phase IV at t+20 (t+39.5).

5.2.2 Radiometric

Case I:

Phase II, which began at t-10, saw a localised peak in the high–cloud count index (Figure 5.12). A strong upper level flow pattern is evident in the satellite imagery at this point, although development is at an early stage. By Phase III at t0, the index has fallen by about one percent. It peaks at t+8 at 10.3% of maximum (as a percent of high count pixels available in the unmasked portion of the image frame), and is at a minimum of 6% at t+20. The cyclonic index is very strong for this case, growing from approximately 10% at the start of Phase II and reaching nearly 100% of maximum in the range of t-5 to t+5 (Figure 5.13). The storm centre is reasonably well located by the maximum index position. The index then drops to a low of 2% in Phase IV, around the time of maximum deepening. Edge sharpness begins with a low gradient value of around 55 and grows to a peak of 82.6 at t+1 (Figure 5.14). There are peaks in the maximum gradient value in the entire image of 1100 at t-8, and 1168 at t-1 which roughly correspond to Phases II and III. Another peak of 1108 at t+4 occurs in early Phase III.

Case II:

The maximum high–cloud (Figure 5.15) value is 32.66%, occurring at t+19. This time falls after the start of Phase IV, suggesting that the mature system contained the most high–cloud. Cyclonic index follows the same pattern, but with a more pronounced spike at t+23 of 29.8% (Figure 5.16). At this point, the system is quite mature in the imagery, and definitely in decay. The cyclonic index maximum does not actually fall over the low, although it picks the right system and a reasonable time. Edge values range from about 65 around Phase II to a maximum of 84.7 at t+20.5 (Figure 5.17). A small peak at t+1 of 69.8 corresponds to Phase III. Peaks at t+10 and t+20.5 occur in late Phase III and early Phase IV respectively. There is a large edge maximum of 1098 at t-0.5 and a general increase in maximum values around t+18 in early Phase IV.

Case III:

High–cloud values begin around 25% at t-10, drop by about 3.5% around the time of maximum deepening (t-2), and rise to a local peak at t0; the start of Phase III (Figure 5.15). Counts then fall progressively to a minimum of 14.77% before Phase IV begins (the analysis was not continued past t+11, while Phase IV began at t+16). Initial high returns appear from the imagery to be due to a previous low centre in the frame along whose axis the Case III low also later develops. Maximum deepening for this case appears to have occurred at t-2 (an anomaly noted above). The upstream system appears to be in Phase II or III at this point, and is providing the majority of the high–cloud pixels. Cyclonic index in this case was more reliable. The maximum index value is a low 1.9% at t-3, but it correctly places the low both spatially and temporally (Figure 5.16). Edge sharpness begins high and falls to a low at t+2 (Figure 5.17). This anomalous condition can be easily explained by a large upstream system moving over land and hence into the masked out area of the edge sharpness images. The effects of the target cyclone are obscured; an unfortunate weakness of the edge strength method. It is unclear if the large edge maximum value at t-5 is due to the target cyclone or the upstream system.

Case IV:

High–cloud counts peak at t+12.5 with a value of 25.1% (Figure 5.15). This time falls in the middle of Phase III, which is the logical time to expect a well developed upper cloud field. A smaller peak of 17.25% at t+30.5 corresponds to the time of maximum deepening. The cyclonic index has a maximum value of 15.54% at t+37.5. The local spike is quite sharp. This time follows the time of maximum deepening (t+30) by a few hours. The system is completely occluded at this point and is decaying. The maximum index is located on a downstream developing system (Figure 5.16). Edge values dip to a low of 42.3 at t+1, then rise progressively to a value of 59.7 at t+22, corresponding to Phase III and early Phase IV respectively (Figure 5.17). An anomalous high value at t+36 corresponds to an erroneously transmitted image frame which contains a large white semi–circular artifact in the upper portion of the image. This feature is also picked out on the maximum sharpness index. A maximum edge spike at t+18 of 1248 occurs one hour after the start of Phase IV, but is likely caused by a similar but smaller image artifact in that time's image.

Case V:

At t+18.25 the high–cloud index reaches its maximum of 15.14% in a noisy distribution (Figure 5.15). This point is just prior to Phase IV, but precedes the time of maximum deepening by about 15 hours (an inconsistent condition). Smaller peaks at t+0.5 and t+33 correspond to Phase III and the time of maximum deepening. The cyclonic index has a strong maximum peak at t+15 of 84.75% (Figure 5.16). This time is in late Phase III, five hours prior to Phase IV. The index maximum is well located in the upper part of the comma head. The index avoids incorrectly choosing a coincident system to the west. Values begin to rise late in the life–cycle of Case V as a newly developing system enters the image frame. Edge sharpness values vary slightly about a value of 55 until t+29, when they begin to rise to their highest mean value of 65.0 at t+37. They then fall gradually back to previous values around t+50 (Figure 5.17). The 65.0 maximum falls just after the point of maximum deepening at t+33.5. The maximum sharpening value plot reveals a few large peaks, most notably: 1476 at t+34.5, just after the point of maximum deepening (t+33.5); 1296 at t+1.5, just after the start of Phase III; and 1164 at t+54, late in Phase IV. Unfortunately, both the t+34.5 and t+1.5 values are likely caused by a single, bright semi–circular line artifact in each of these images.

5.2.3 Vector

Case I:

Jet Tilt in degrees north of east increased as the system developed from Phase II to Phase III (Figure 5.18). The progression was smooth, as the system tilted from 32.5° at t-10 to 43.1° at t-3. There was a counter–rotation (anti–cyclonic) from t-2 to t-1, followed by a return to rotation to the west at t0. The system rotated at about 1.5° per hour until t-2 (Figure 5.19). The Head Length contracted strongly at t-8 and t-7, followed by a quasi–steady length from t-6 through t-2 averaging 914.2 km. It decreased again just before t0. The mean rate between t-5.5 to t-3 was only -2.71 km/h. Overall, the mean rate was -11.0 km/h. The Comma Bulge also decreased strongly from the start of Phase II until t-7, when it was 367.4 km, which further indicates the system was contracting at this point. The Bulge then increased in size fairly uniformly for the rest of the tracking time, with a mean 18.2 km/h rate of expansion. A small decrease in length of 29.4 km at t-2 corresponds with the drop in rotation rate at the same period. Baroclinic Width roughly increased in time, with a mean rate of 4.8 km/h. Variation about a rate of zero increased with time, probably because the jet width became more difficult to measure in later frames.

Case II:

The jet axis in this case was tilted past north, indicating a meridional flow pattern (Figure 5.20). Tilt increased rapidly in the first few image frames until t-5, then slowed slightly before continuing to increase gradually until t0. The mean rate until t-5 was 4.7°/h; from t-4.5 to t0, the mean rate was only 0.5°/h (Figure 5.21). The comma head shrank rapidly between t-5.5 and t-5 by 147.8 km (-295.6 km/h). By t-3.5 the head had shrunk a total of 234.3 km in just 3.5 hours. In then generally increased in length until t0 at a mean rate of 7.8 km/h for an overall contraction of 129.5 km. The Comma Bulge was generally small before t-3.5 when it was at a minimum of 700.5 km, and then generally increased in size. There was a contraction of 38.8 km between t-2 and t-1.5 but overall between t-3.5 and t0 the bulge increased at a mean rate of 22.6 km/h. Over the entire length of observation, the bulge increased at a mean rate of 12.5 km/h by a total of 87.2 km. Baroclinic Width dropped sharply between t-7 and t-6.5 66.1 km then increased rapidly to a maximum value of 254.8 km at t-4. The width was increasing by 58.6 km/h between t-6.5 to t-4 with a maximum rate of 163.8 km/h at t-5. After t-4, it decreased at a rate of -20.7 km/h. The mean width was 187.9 km, while the widths at the first and last observation were 174.5 and 172 km respectively.

Case III:

Overall, the Jet Tilt increased; however, prior to t-5 there was some variability (Figure 5.20). The range of these measures was very small with a difference between the minimum and maximum tilt values of only 2.42°, suggesting that this variation was not particularly noteworthy. Overall, the mean tilt was 67.5° versus a minimum and maximum tilt of 62.2°and 73.4° respectively. The system tilted at a mean rate of only 0.7°/h over the entire interval (Figure 5.21). Head Length was generally increasing until a sharp drop of 253.3 km between t-4 and t-3. The "steps" prior to t-5 were quite unusual, and somewhat systematic. The early (left) side of the steps showed increases between t-9.5 to t-8.5 and t-7 to t-6 are 91.9 and 110.2 km respectively. The late (right) side of the "steps" showed decreases between t-8.5 to t-7 and t-6 to t-5 of -64.7 and -67.9 km respectively. It is quite possible that the unusual nature of Case III made systematic placement of the comma grid difficult. The overall rate of increase was 17.3 km/h with a mean head length of 1234.2 km. Comma Bulge was similarly asystematic. Generally, the bulge size was large where the tilt was shallow. There was a large drop in size between t-4 and t-3 of 168.5 km. The mean bulge size was 648.3 km with a mean rate of change of 1.88 km/h. For the first six observations, the jet width could not be visually estimated. Over the last four observations, the mean width was 189.9 km.

Case IV:

Geometric measures for Case IV were quite linear in comparison to the other four cases. Jet Tilt was greater than 90° throughout, making this a meridional system (Figure 5.20). Tilt increased quite systematically with a mean rate of 105.5°/hr through a range of 5.5°(Figure 5.21). Comma Head Length increased over time with a mean rate of 67.7 km/h with a mean length of 847.8 km and an overall change in length of 305.1 km. Comma Bulge increased with a mean rate of 39.3 km/h and had a mean length of 367.3 km, increasing overall by 176.8 km. The Baroclinic width was more variable, but increased fairly systematically until t-0.5 with a mean rate of 13.8 km/h. After t-1 there was a strong drop of 82.5 km, which may have resulted from loss or ambiguity of the visual feature being tracked as representative of the jet in later frames.

Case V:

Jet Tilt crosses 90° early Phase II, making this a meridional case (Figure 5.20). Tilt increases fairly systematically until t-1.5 to a value of 103.0°. There is a slowing of the rate of rotation around t-4, and even a small temporary reversal of rotation just after this period (Figure 5.21). The system counter–rotates again after t-2 and then stabilises with a mean rotation rate of 0.2°/hr. From the start of Phase II until t-2, the mean rate is 4.4°/hr, while the overall rotation rate for the case is 2.5°/hr. Comma Head Length begins with a slow increase then jumps rapidly at a rate of 106.4 km/h from t-5.25 to t-4. It then increases in a rough manner at a mean rate of

26.7 km/h. The overall mean rate is 34 km/h. Comma Bulge increases in size in a similar pattern to the rate of rotation. There is a similar slowing of the rate of increase around t-4 through t-3, followed by a fairly steady rate of increase of 32.3 km/h from just after t-2.5. The overall rate of increase for the case is 37.5 km/h with a growth of 234.5 km. Baroclinic Width is broken into two estimate: those up to t-4.75 and those that follow. This is due to the feature initially being tracked becoming indistinct, while a new feature was later visible. It is most likely that both of these features could not be good estimates of the jet width. Rates in the two parts were 26.4 km/h and 11.5 km/h respectively. The mean rate of these two periods (thus removing the jump between features) is 18.9 km/h.

Table 5.1
Event Dates–April 24, 1996, Case I.

	First Chart	Last Chart	First Image	Last Image
Year	96	96	96	96
Month	4	4	4	4
Day	24	25	24	24
Hour	0	18	12	22
Minute	0	0	0	0

Table 5.2
Bergeron Estimates–April 24, 1996, Case I. One Bergeron equals 24 hPa/24 hr deepening at $60° N$· The *6i* interval is interpolated to 6 hours.

Time Interval	Delta mb	Bergeron
6i	14	2.71
6	14	2.71
12	14	1.38
24	26	1.34

Table 5.3
Event Dates–January 12, 1997, Case II.

	First Chart	Last Chart	First Image	Last Image
Year	97	97	97	97
Month	1	1	1	1
Day	12	14	13	13
Hour	18	18	1	8
Minute	0	0	0	0

Table 5.4
Bergeron estimates–January 12, 1997, Case II. One Bergeron equals 24 hPa/24 hr deepening at $60°N$·The *6i* interval is interpolated to 6 hours.

Time Interval	Delta mb	Bergeron
6i	13	3.09
6	13	3.09
12	17	2.06
24	21	1.19

Table 5.5
Event Dates–January 28, 1997, Case III.

	First Chart	Last Chart	First Image	Last Image
Year	97	97	97	97
Month	1	1	1	1
Day	28	29	28	29
Hour	18	12	16	2
Minute	0	0	30	0

Table 5.6
Bergeron Estimates–January 28, 1997, Case III. One Bergeron equals 24 hPa/24 hr deepening at $60°N$. The *6i* interval is interpolated to 6 hours.

Time Interval	Delta mb	Bergeron
6i	6	1.65
6	6	1.65
0	0	0.00
0	0	0.00

Table 5.7
Event Dates–November 8, 1997, Case IV.

	First Chart	Last Chart	First Image	Last Image
Year	97	97	97	97
Month	11	11	11	11
Day	8	9	7	7
Hour	6	12	13	18
Minute	0	0	30	0

Table 5.8
Bergeron Estimates–November 8, 1997, Case IV. One Bergeron equals 24 hPa/24 hr deepening at $60°N$. The *6i* interval is interpolated to 6 hours.

Time Interval	Delta mb	Bergeron
6i	11	2.08
6	11	2.08
12	19	1.86
24	22	1.04

Table 5.9
Event Dates–March, 6, 1998, Case V.

	First Chart	Last Chart	First Image	Last Image
Year	98	98	98	98
Month	3	3	3	3
Day	6	8	6	6
Hour	6	18	2	8
Minute	0	0	15	30

Table 5.10
Bergeron Estimates–March 6, 1998, Case V. One Bergeron equals 24 hPa/24 hr deepening at $60°N$. The *6i* interval is interpolated to 6 hours.

Time Interval	Delta mb	Bergeron
6i	13	3.03
6	13	2.90
12	18	2.05
24	26	1.45

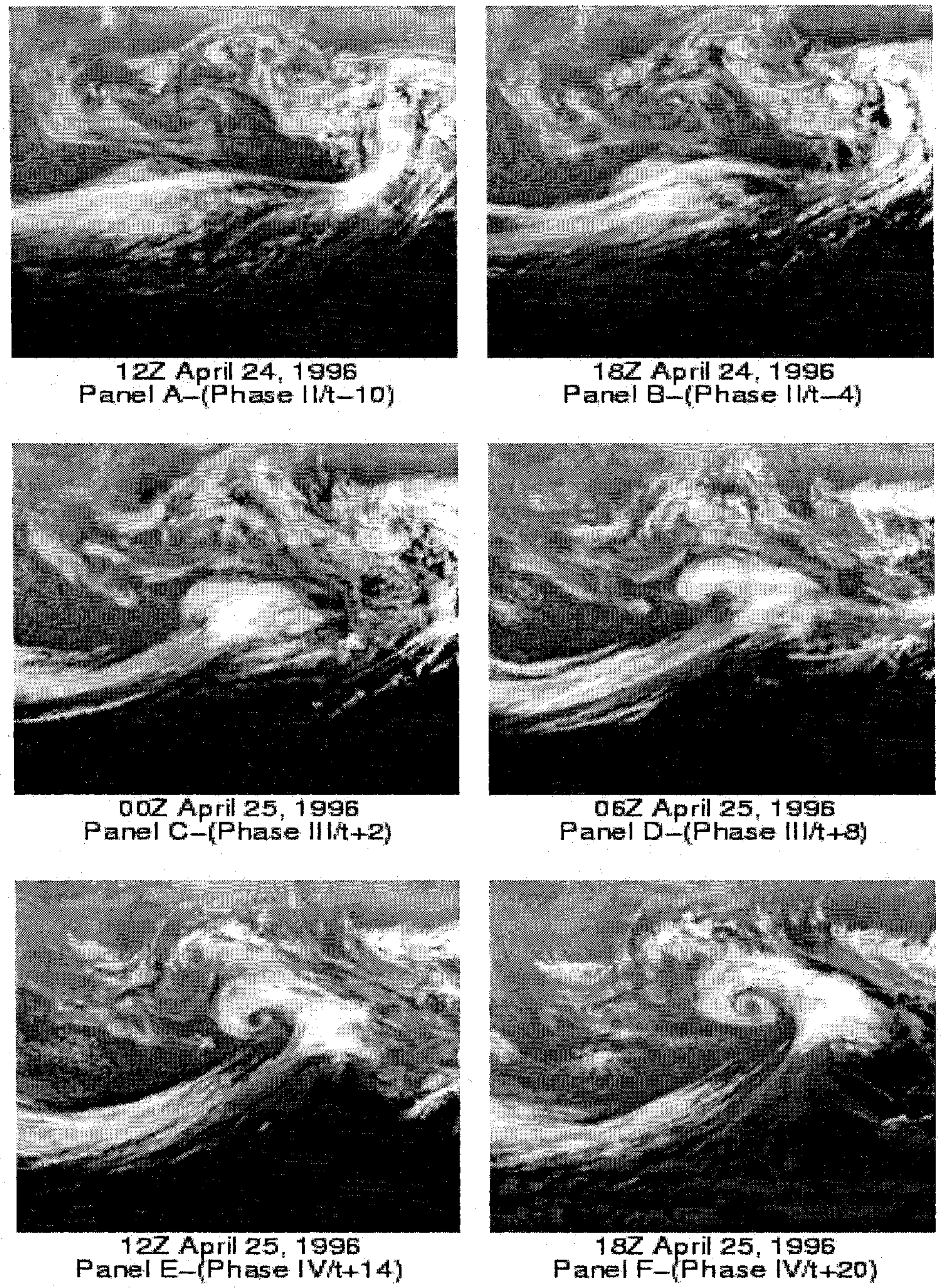


Fig. 5.1. "Contact sheet" indicating development at six hour intervals–April 24, 1996.

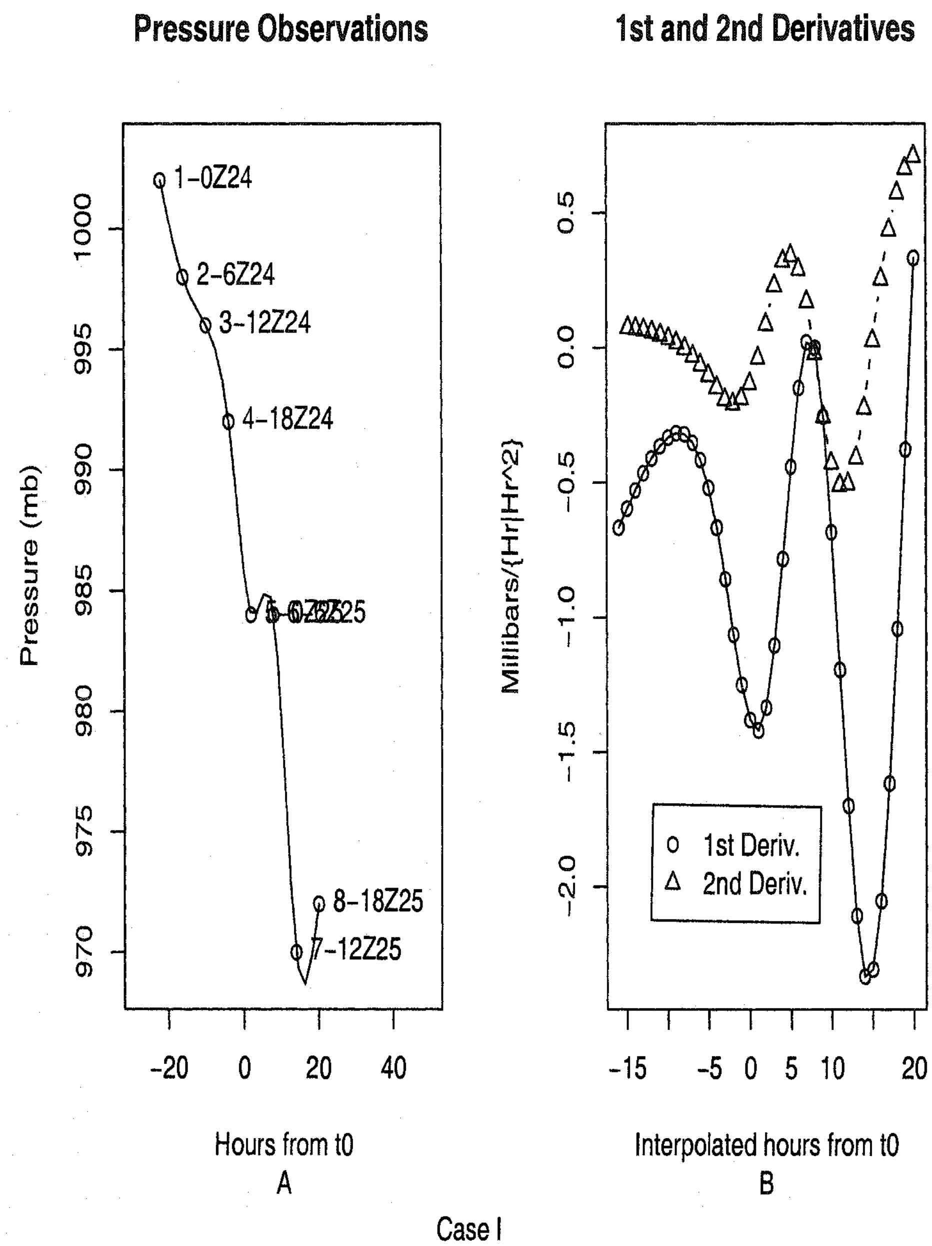


Fig. 5.2. Pressure Record–April 24, 1996, Case I. Pressure centres are from 1000 hPa charts with spline interpolation. t0 denotes the emergence of the dry tongue (see Chapter 2). Plot 2 shows the first and second order time derivatives of pressure.

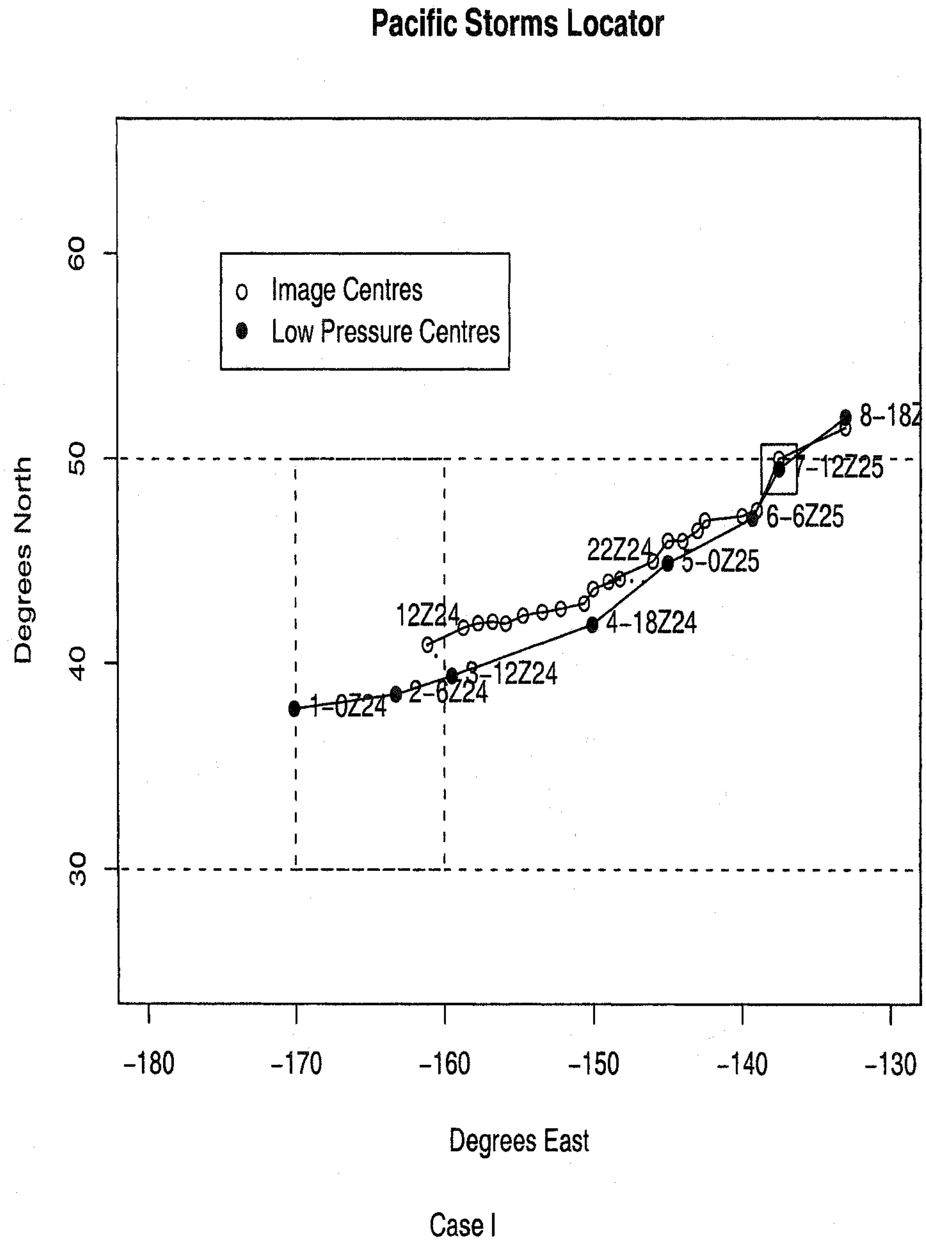


Fig. 5.3. Storm Ground Tracks–April 24, 1996, Case I. The horizontal lines represent normal ranges for bomb development and decay. Vertical lines represent the Pacific maximum frequency position for storm development as per Roebber, 1984. A square is drawn about the location of maximum deepening as found on Figure 5.2. Dashed lines connect Image and Pressure Chart observations that occurred at the same time. The last noted time on the Image line indicates the last observation analysed using the grid method, although subsequent image centres were sometimes available.

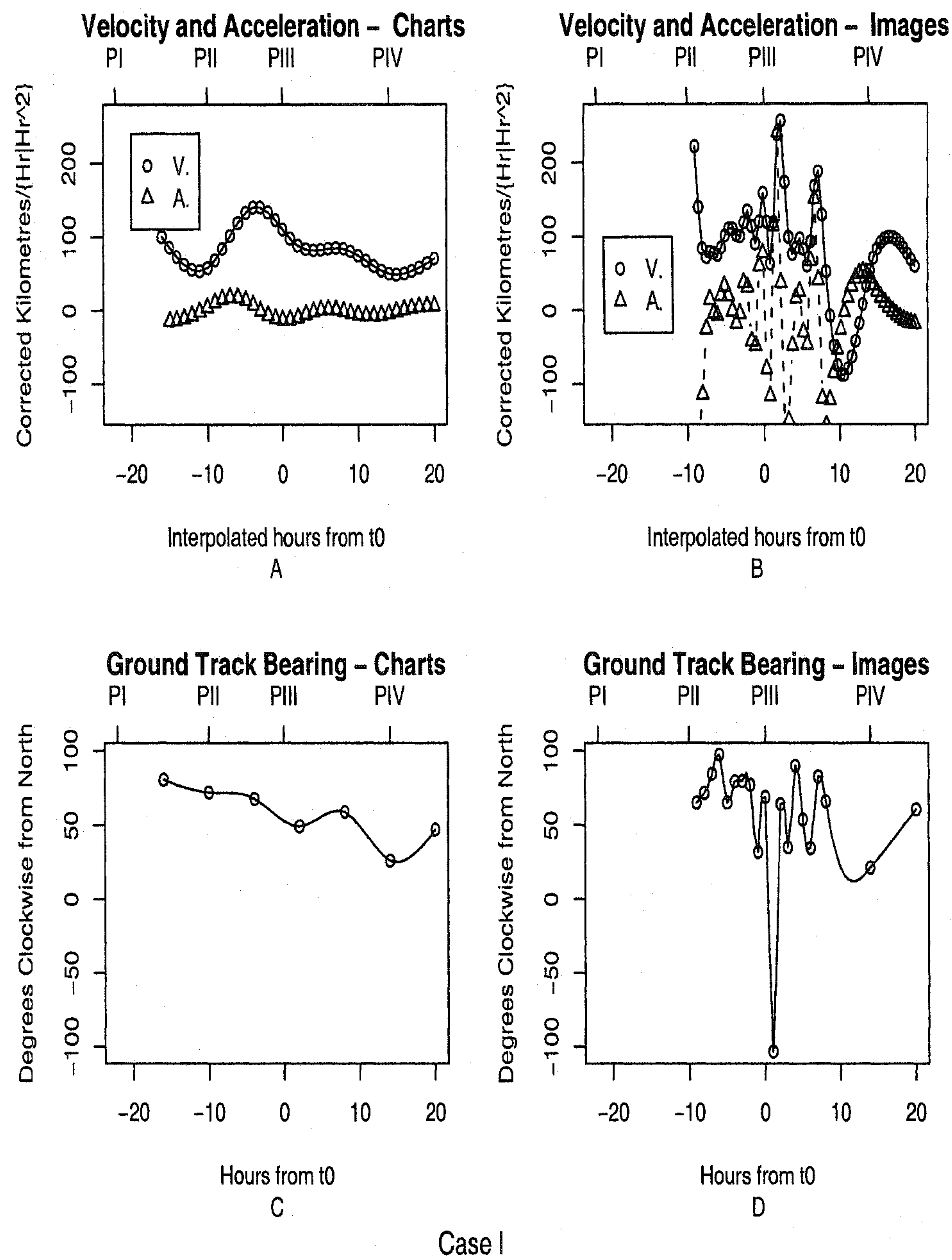


Fig. 5.4. Ground track motions (April 24, 1996, Case I) based on pressure chart low centres (left column) and image low centres (right column). All distance–based measures were computed from an equidistant azimuthal model projection, resulting in true ground units (kilometres).

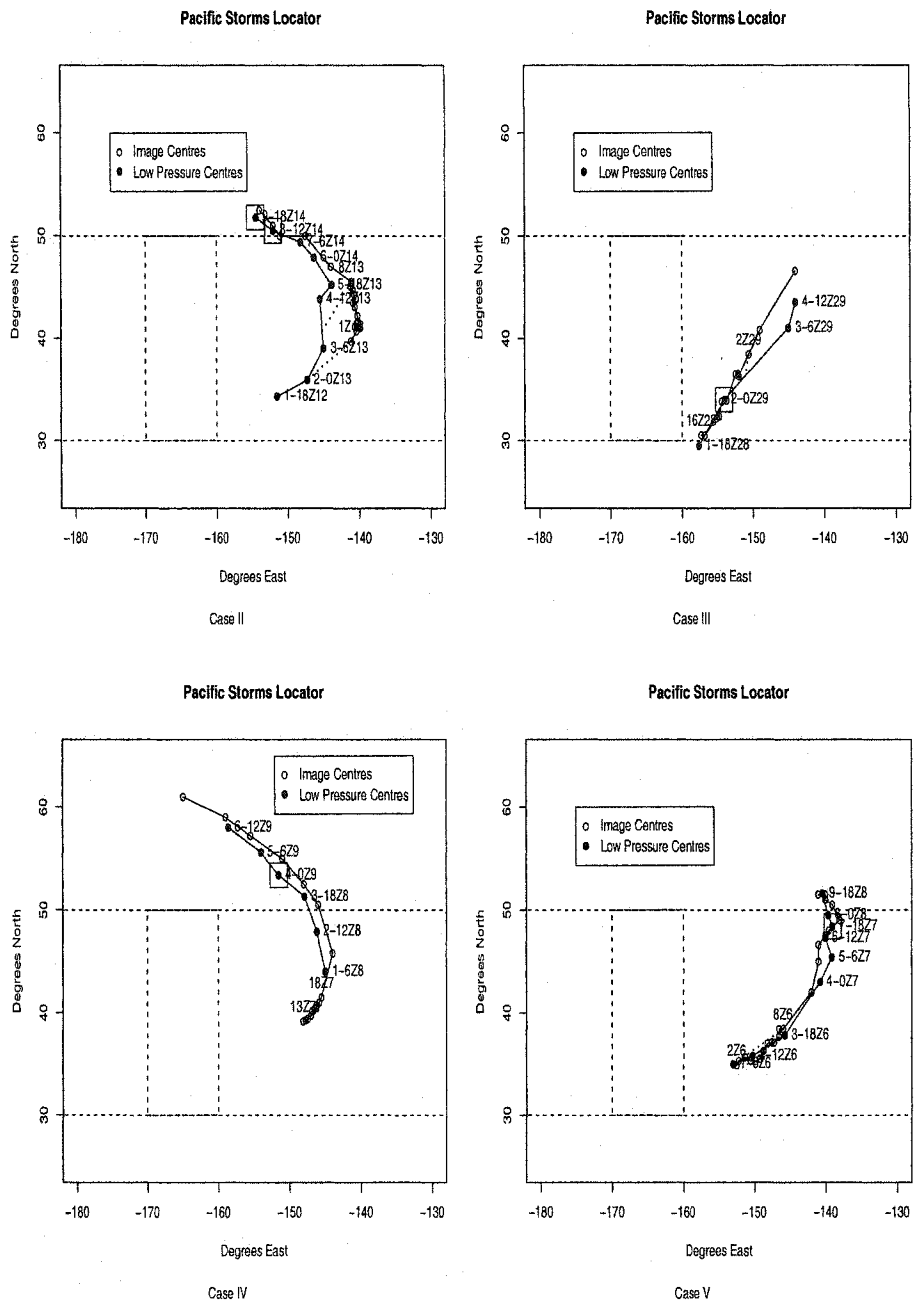


Fig. 5.5. Storm Ground Tracks–Cases II to IV. The North-South lines represent normal ranges for bomb development and decay. East-West lines represent the Pacific maximum frequency position for storm development as per Roebber, 1984.

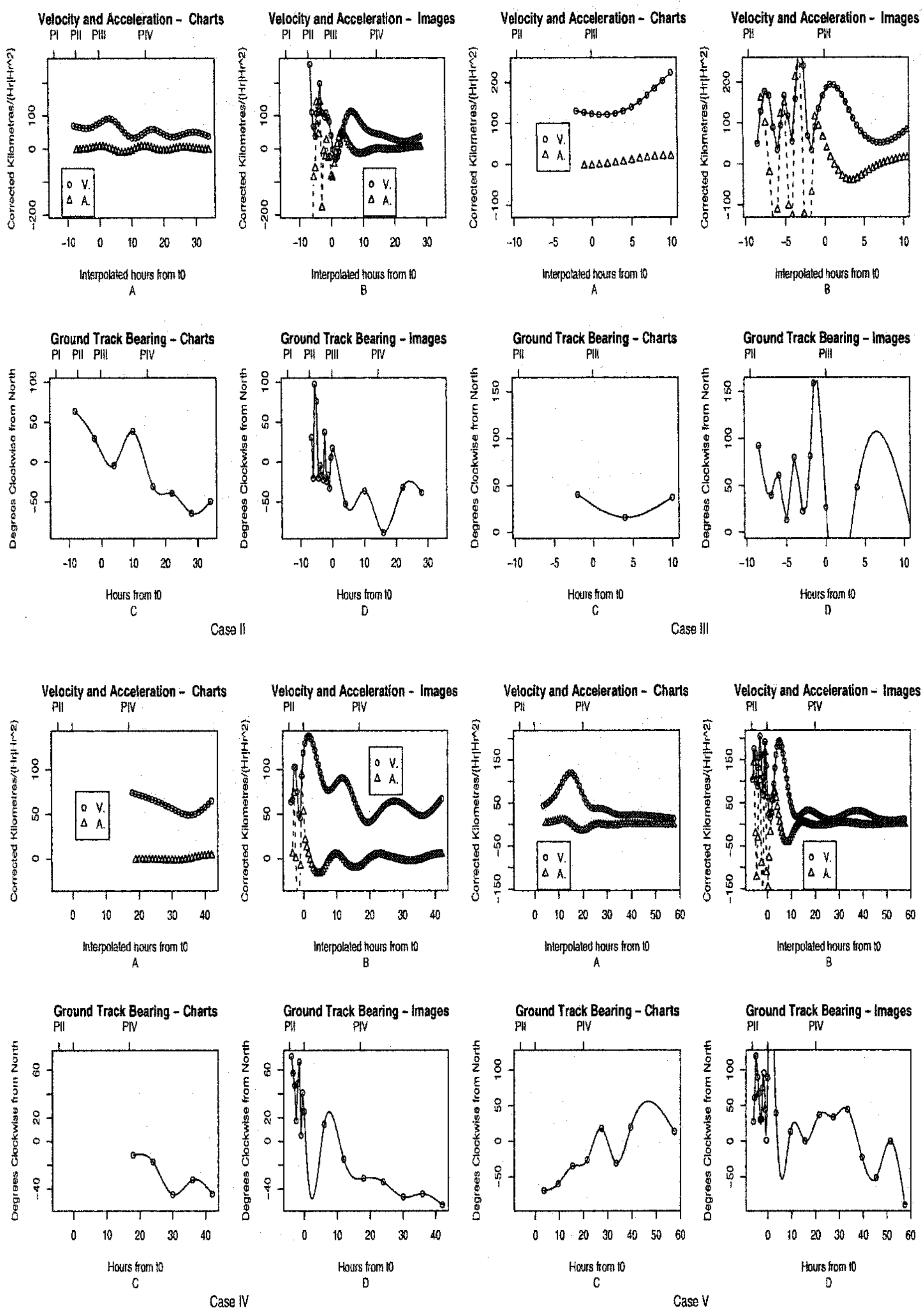


Fig. 5.6. Ground track motions (Cases II to IV) based on pressure chart low centres (left column) and image low centres (right column). All distance–based measures were computed from an equidistant azimuthal model projection, resulting in true ground units (kilometres).

01Z January 13, 1997
Panel A-(Phase II/t-7)

06Z January 13, 1997
Panel B-(Phase II/t-2)

12Z January 13, 1997
Panel C-(Phase III/t+4)

18Z January 13, 1997
Panel D-(Phase III/t+10)

Fig. 5.7. "Contact sheet" indicating development at six hour intervals–January 12, 1997.

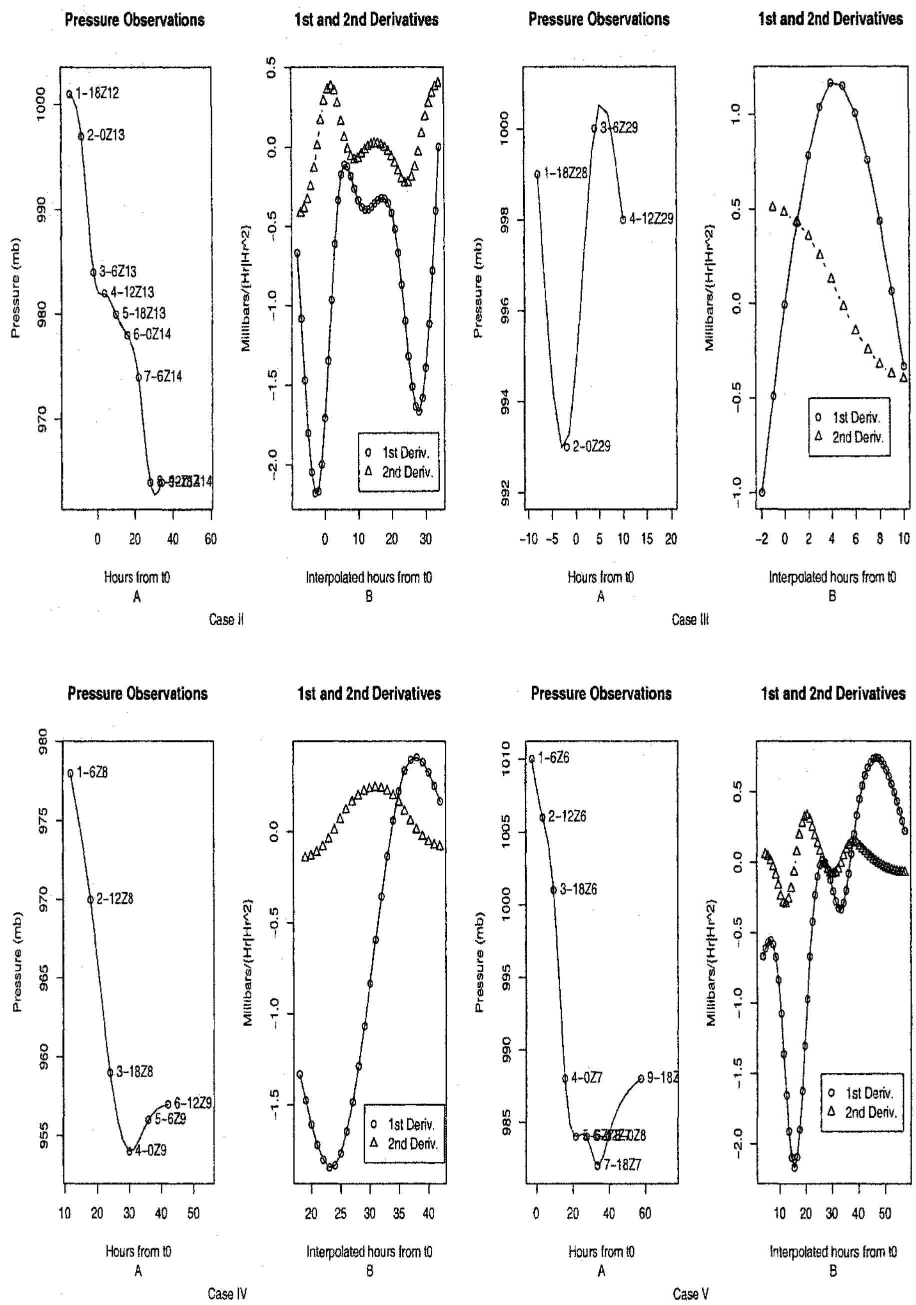


Fig. 5.8. Pressure Record–Cases II to IV. Pressure centres are from 1000 hPa charts with spline interpolation. t0 denotes the emergence of the dry tongue (see Chapter 2 ref.) Plot 2 shows the first and second order time derivatives of pressure.

19Z January 28, 1997
Panel A-(Phase II/t-7)

00Z January 29, 1997
Panel B-(Phase II/t-2)

06Z January 29, 1997
Panel C-(Phase III/t+4)

13Z January 29, 1997
Panel D-(Phase III/t+11)

Fig. 5.9. "Contact sheet" indicating development at six hour intervals–January 28, 1997.

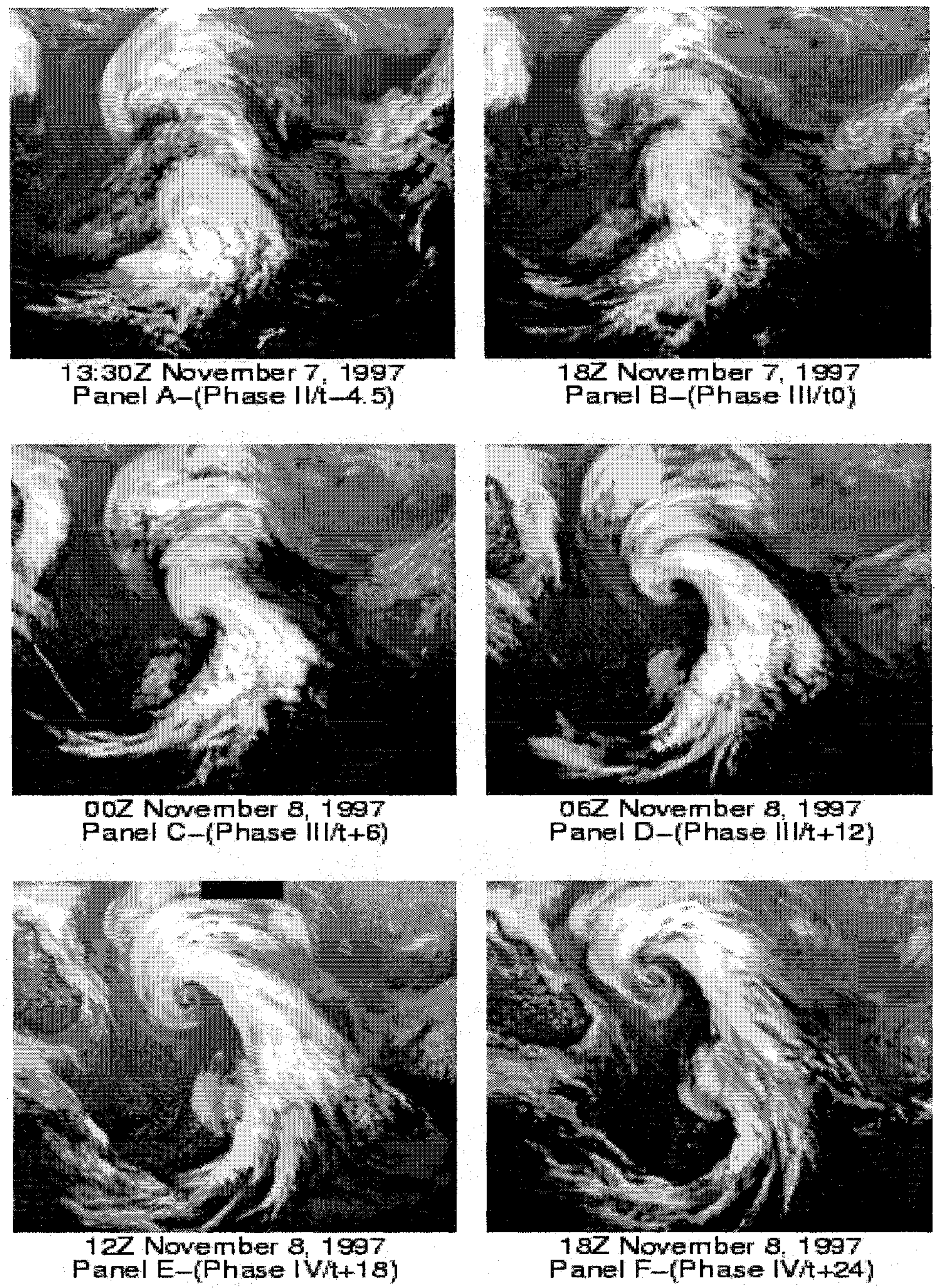


Fig. 5.10. "Contact sheet" indicating development at six hour intervals–November 8, 1997.

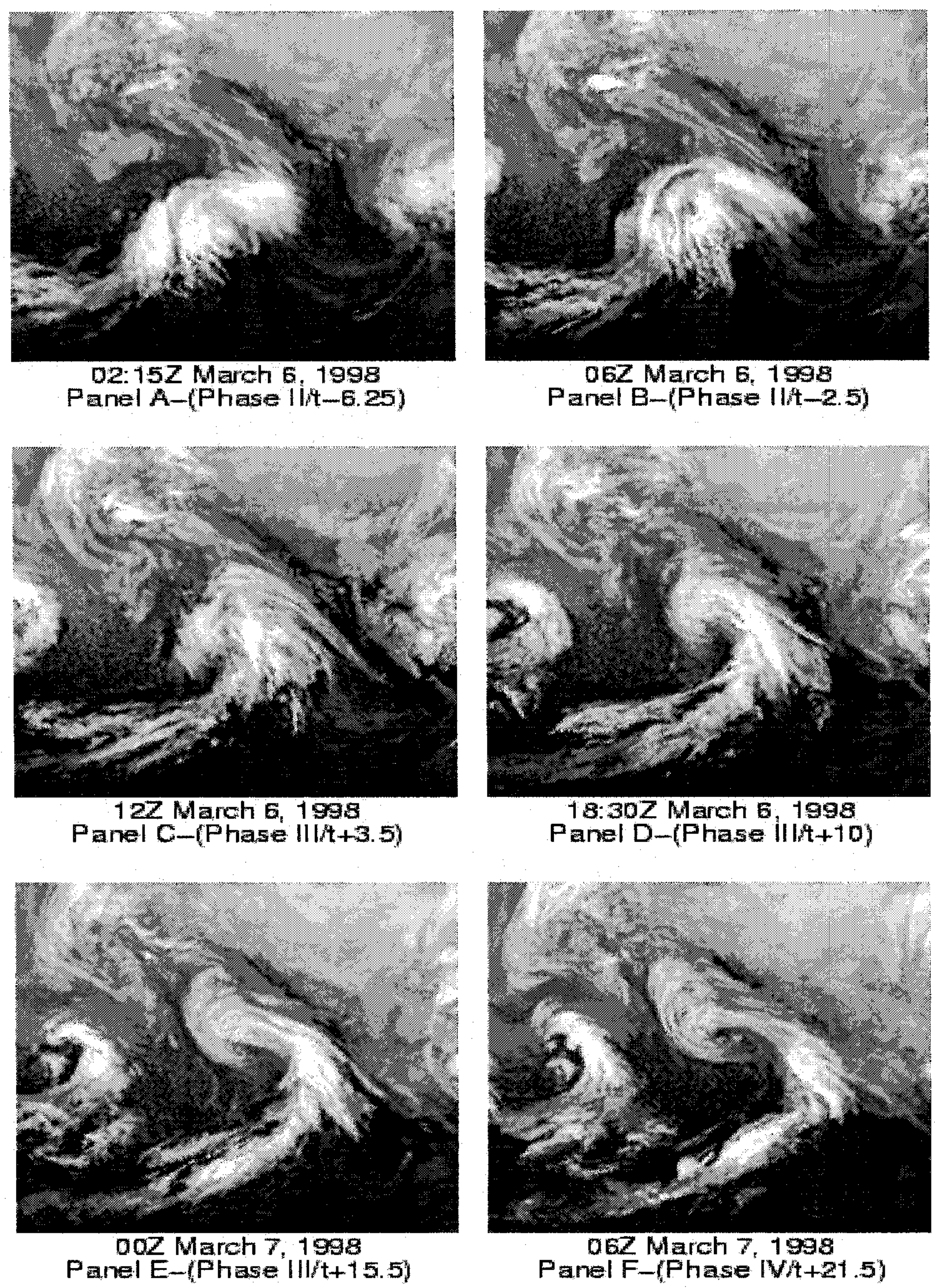


Fig. 5.11. "Contact sheet" indicating development at six hour intervals–March 6, 1998.

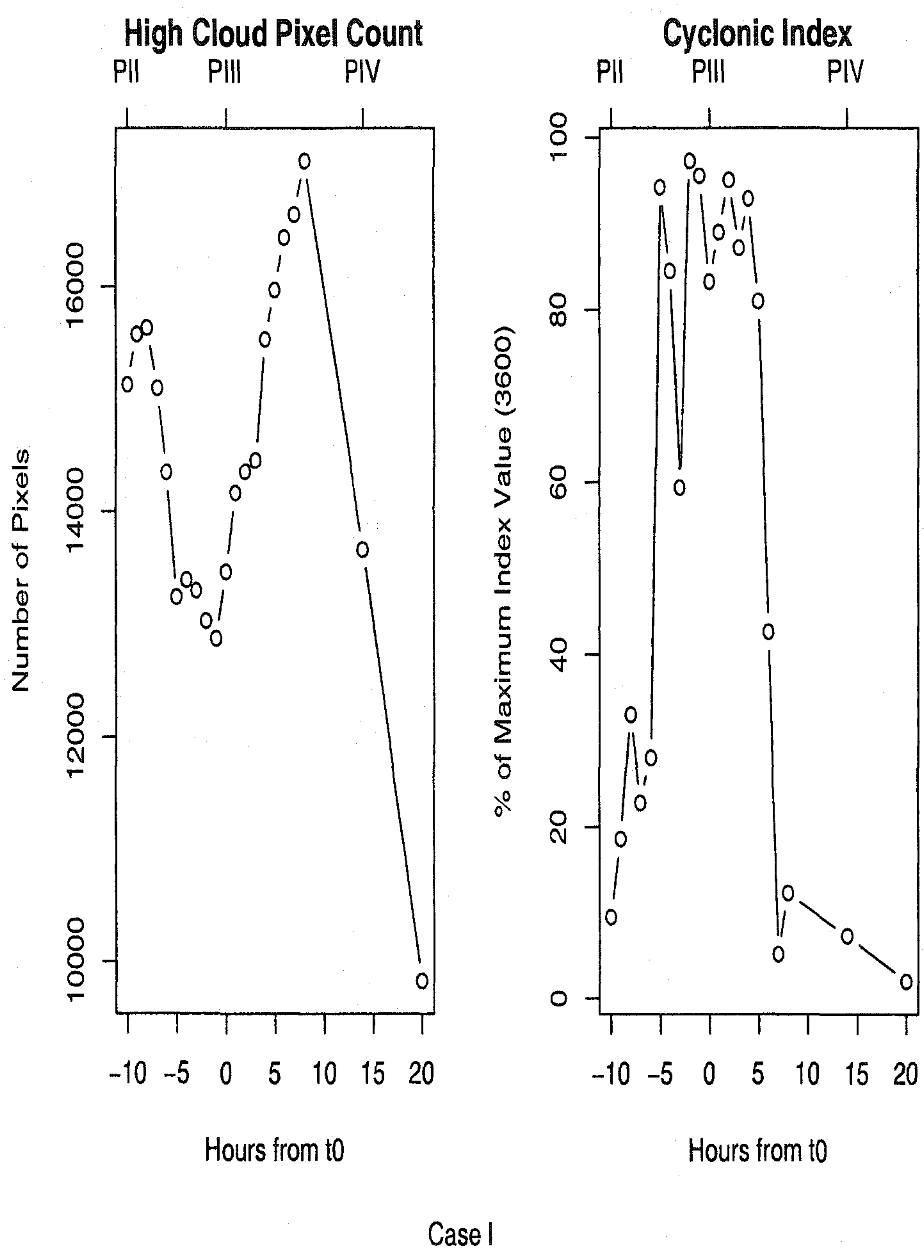


Fig. 5.12. High cloud pixels (April 24, 1996, Case I) have an emissive temperature below 30°C. Cyclonic index is a dimensionless measure indicating the highest cyclone shape match in each time frame.

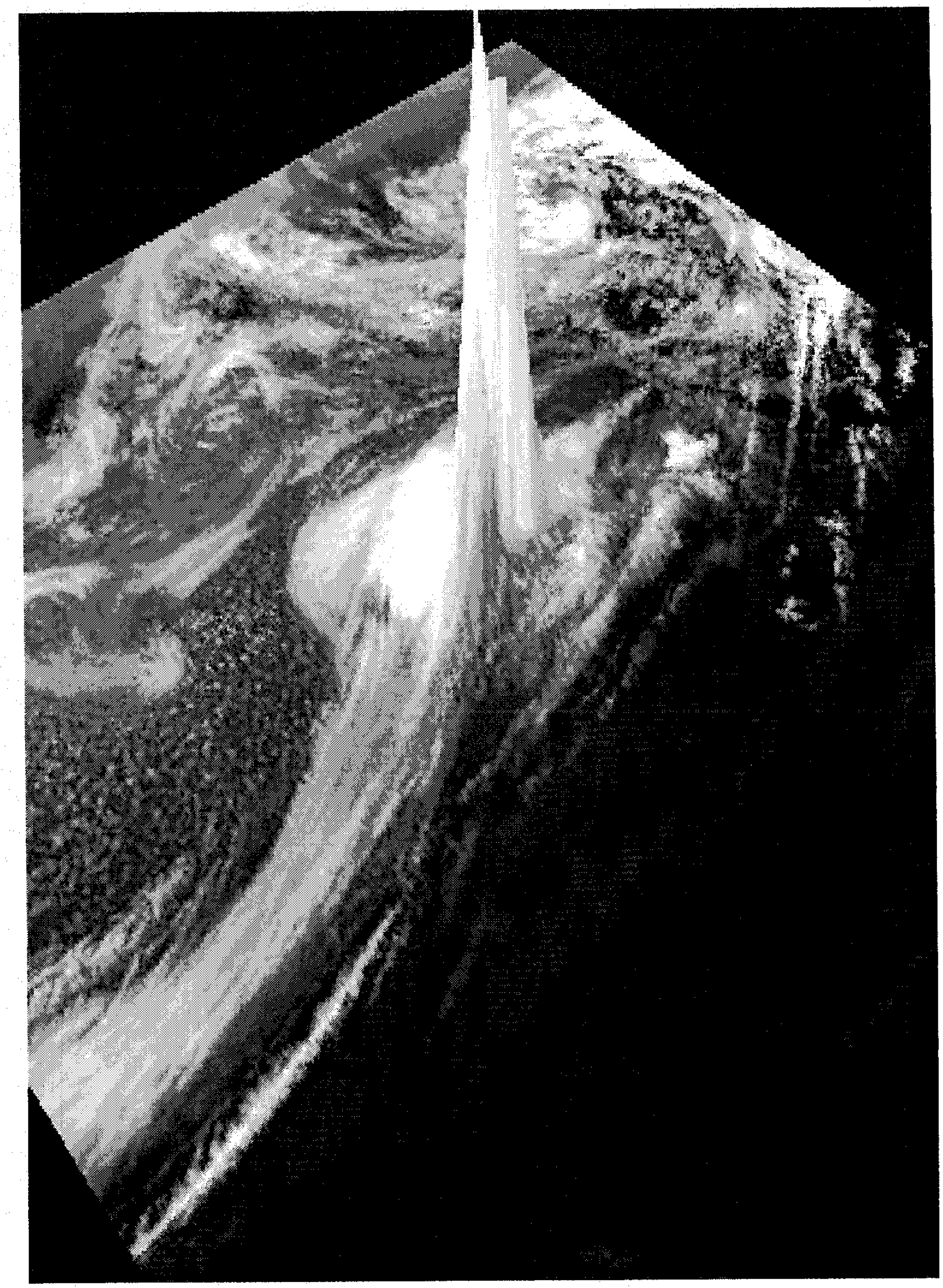
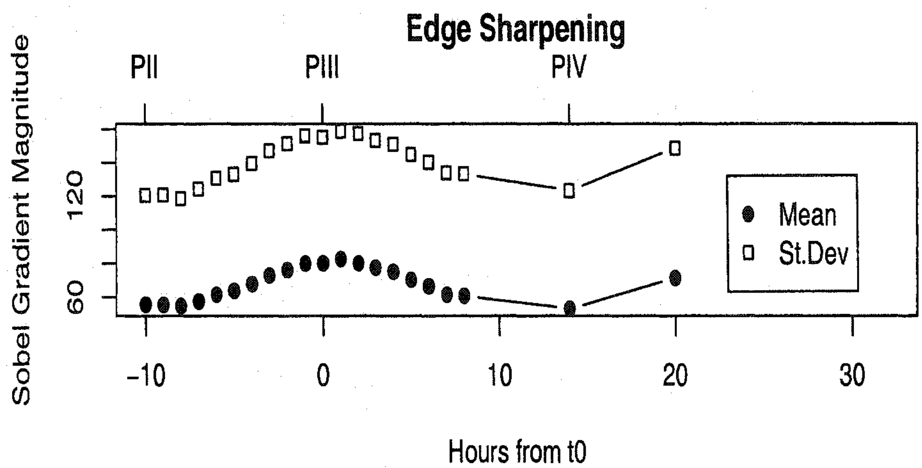
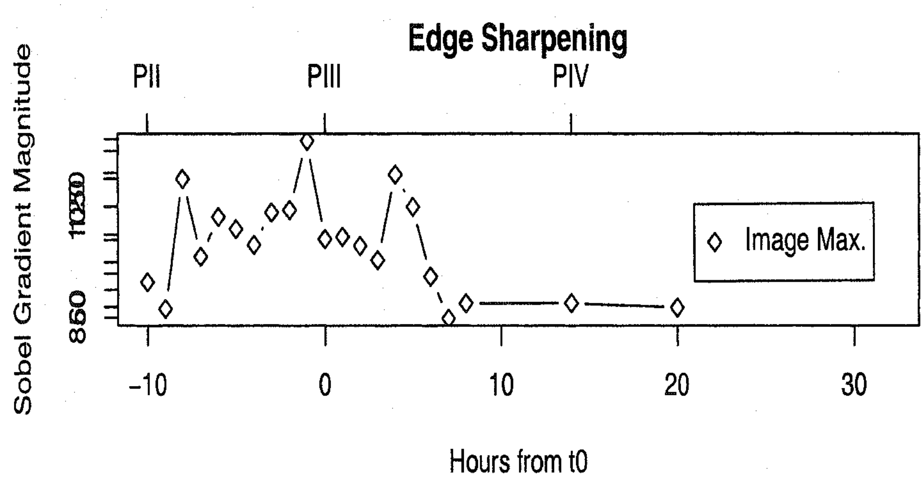


Fig. 5.13. Three dimensional view of Cyclonic Index values–April 24, 1996, Case I. Index is the Z–axis. Vertical exaggeration: 2×





Case I

Fig. 5.14. Results for 3×3 Sobel filter gradient values greater than 100–April 24, 1996, Case I.

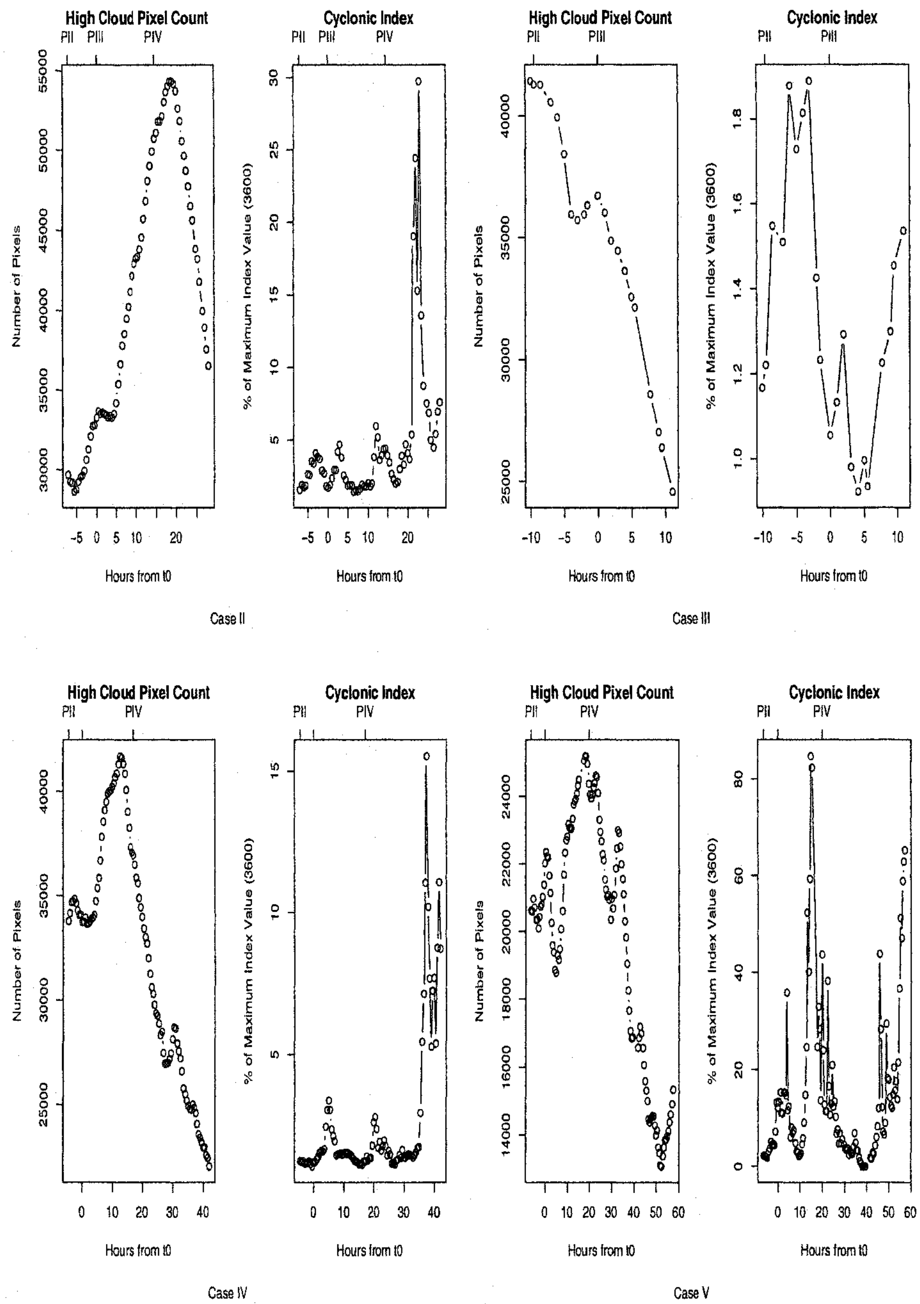


Fig. 5.15. High cloud pixels (Cases II to IV) have an emissive temperature below 30°C. Cyclonic index is a dimensionless measure indicating the highest cyclone shape match in each time frame.

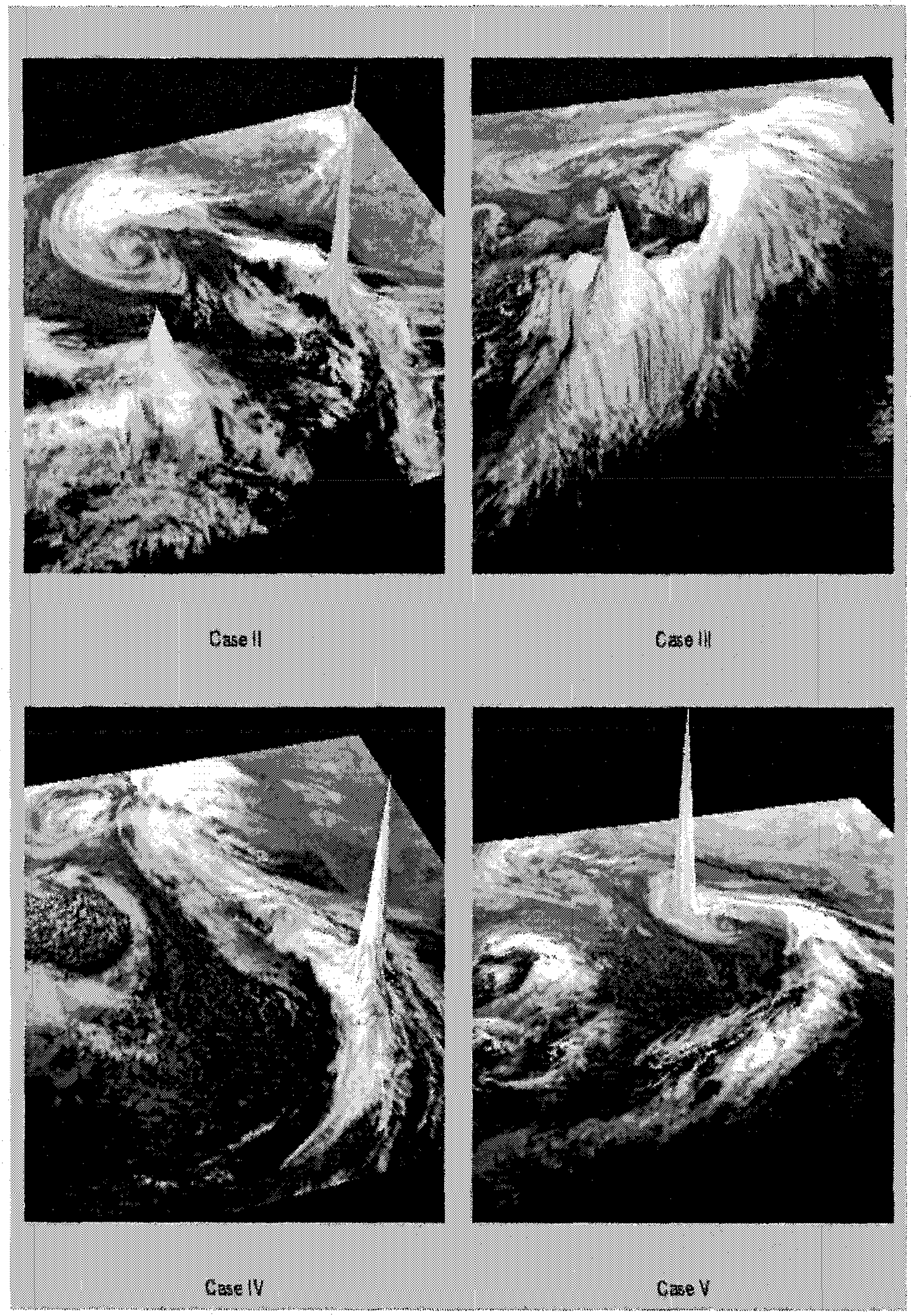


Fig. 5.16. Three dimensional view of Cyclonic Index values–Cases II to IV. Index is the Z–axis. Vertical exaggeration for Cases II through V: 13×, 50×, 3× and 10×.

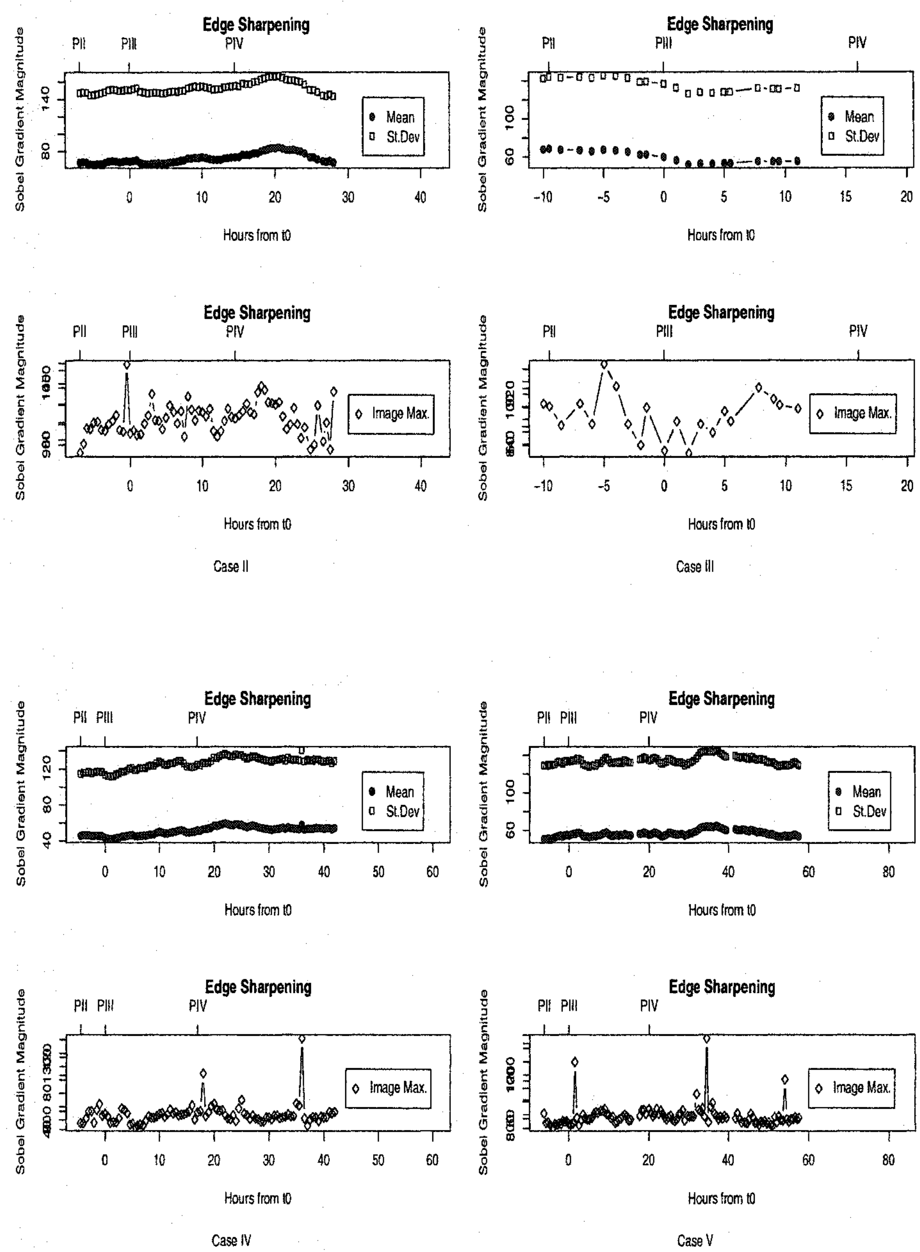


Fig. 5.17. Results for 3 × 3 Sobel filter gradient values greater than 100–Cases II to IV.

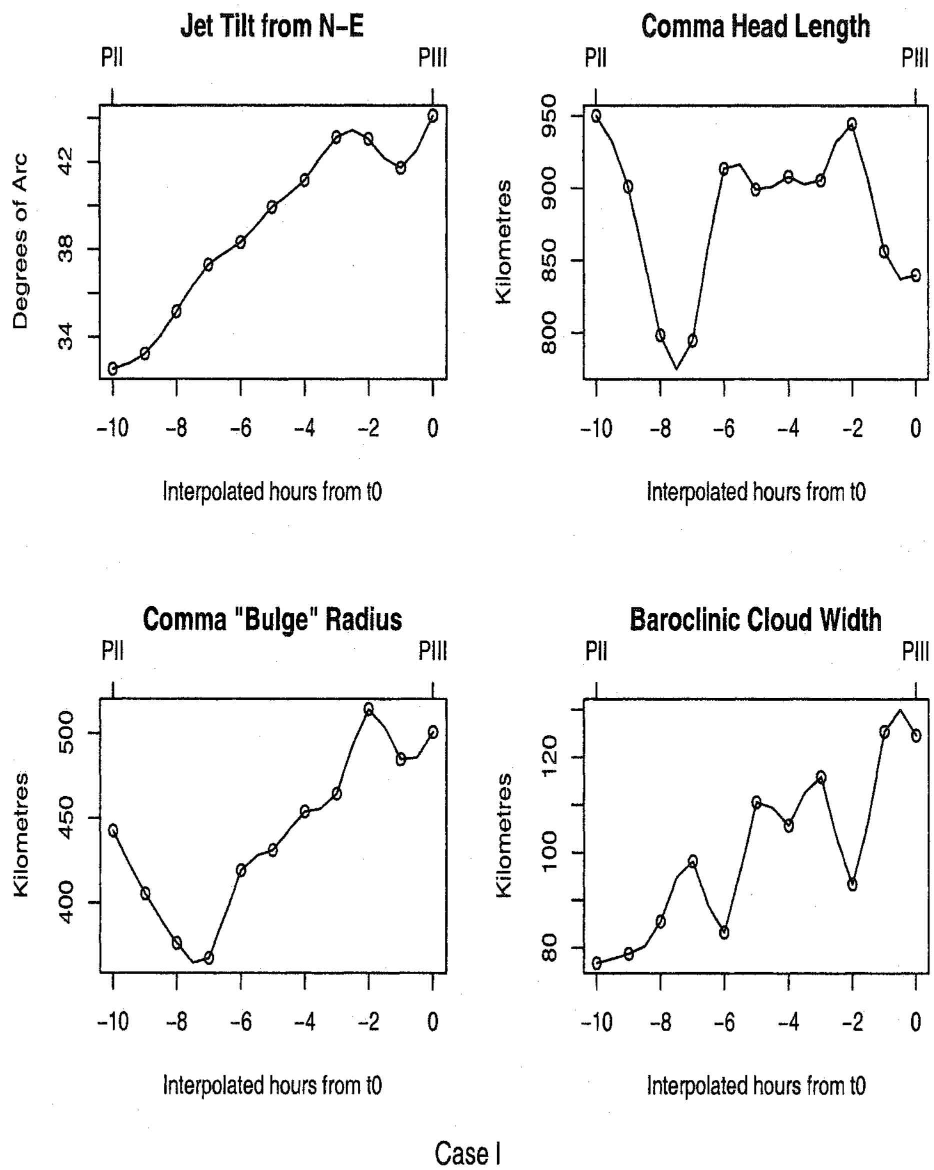


Fig. 5.18. Raw geometric measures (April 24, 1996, Case I), in degrees latitude and longitude, except for rotation, which expresses tilt of the supporting jet axis in degrees of arc. Positive rotation represents a counter–clockwise tilt from the East–West axis.

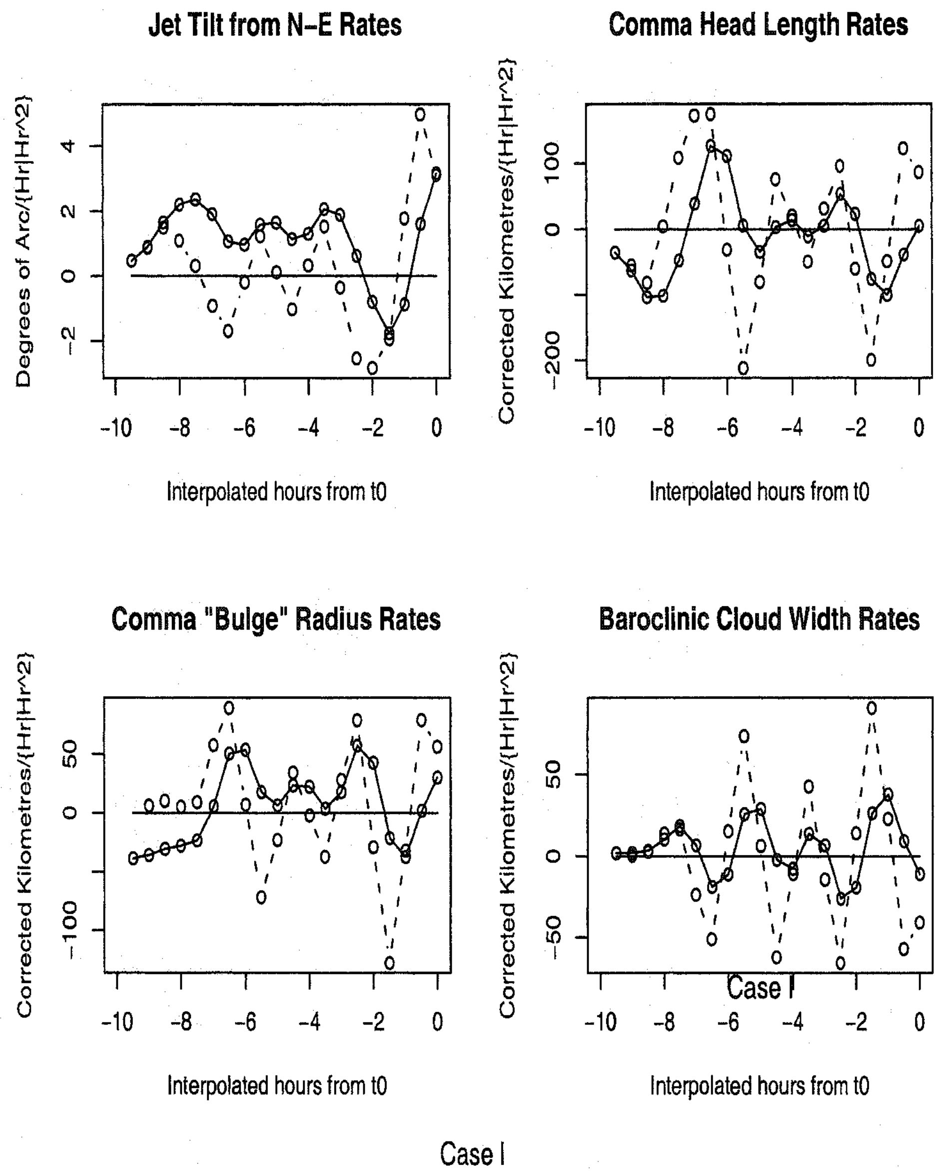


Fig. 5.19. First and second derivatives of geometric measures (solid and dashed lines respectively)–April 24, 1996, Case I. Rotation is in degrees of arc per hour, and per hour per hour. All other measures are in kilometres per hour, and per hour per hour.

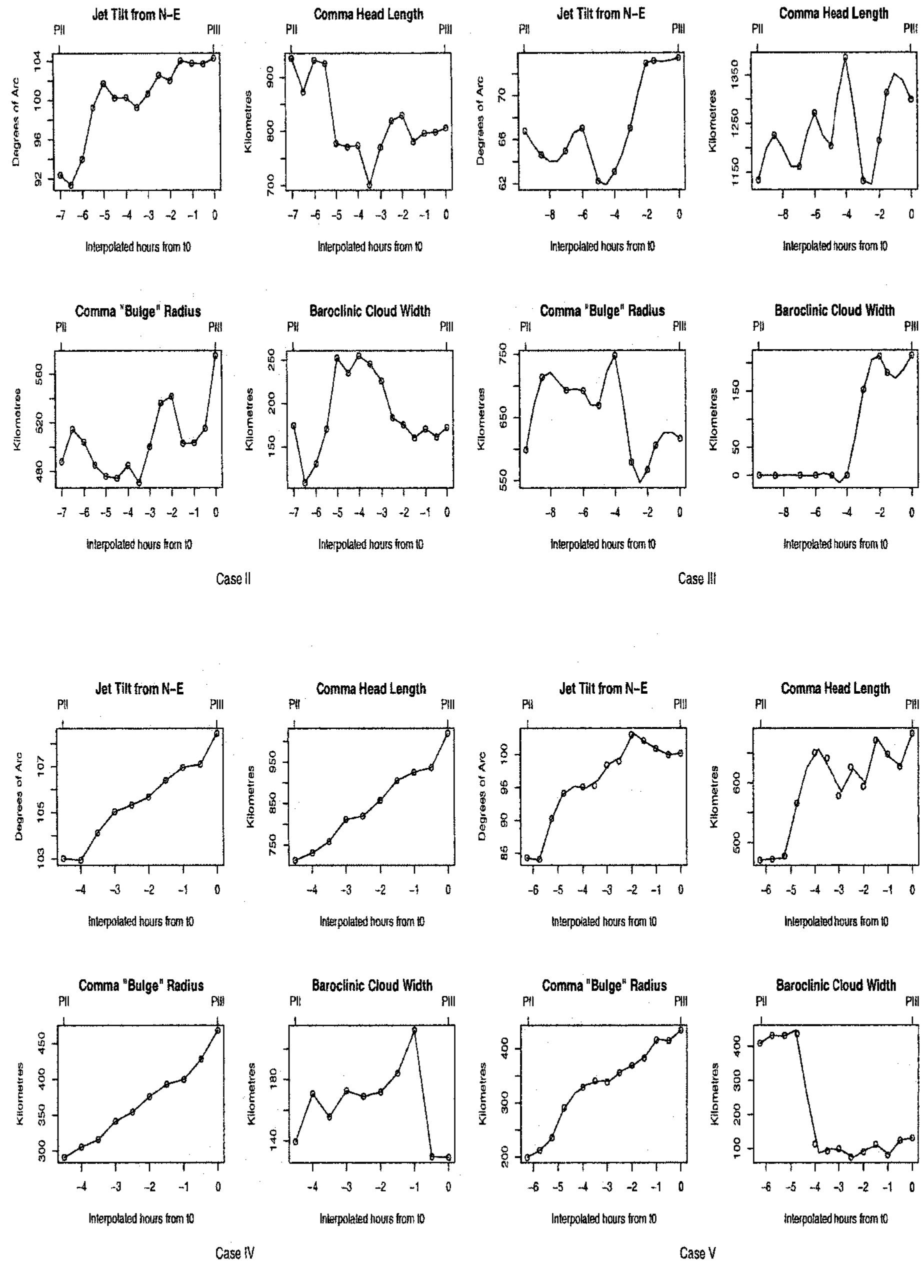


Fig. 5.20. Raw geometric measures (Cases II to IV), in degrees latitude and longitude, except for rotation, which expresses tilt of the supporting jet axis in degrees of arc. Positive rotation represents a counter-clockwise tilt from the East–West axis.

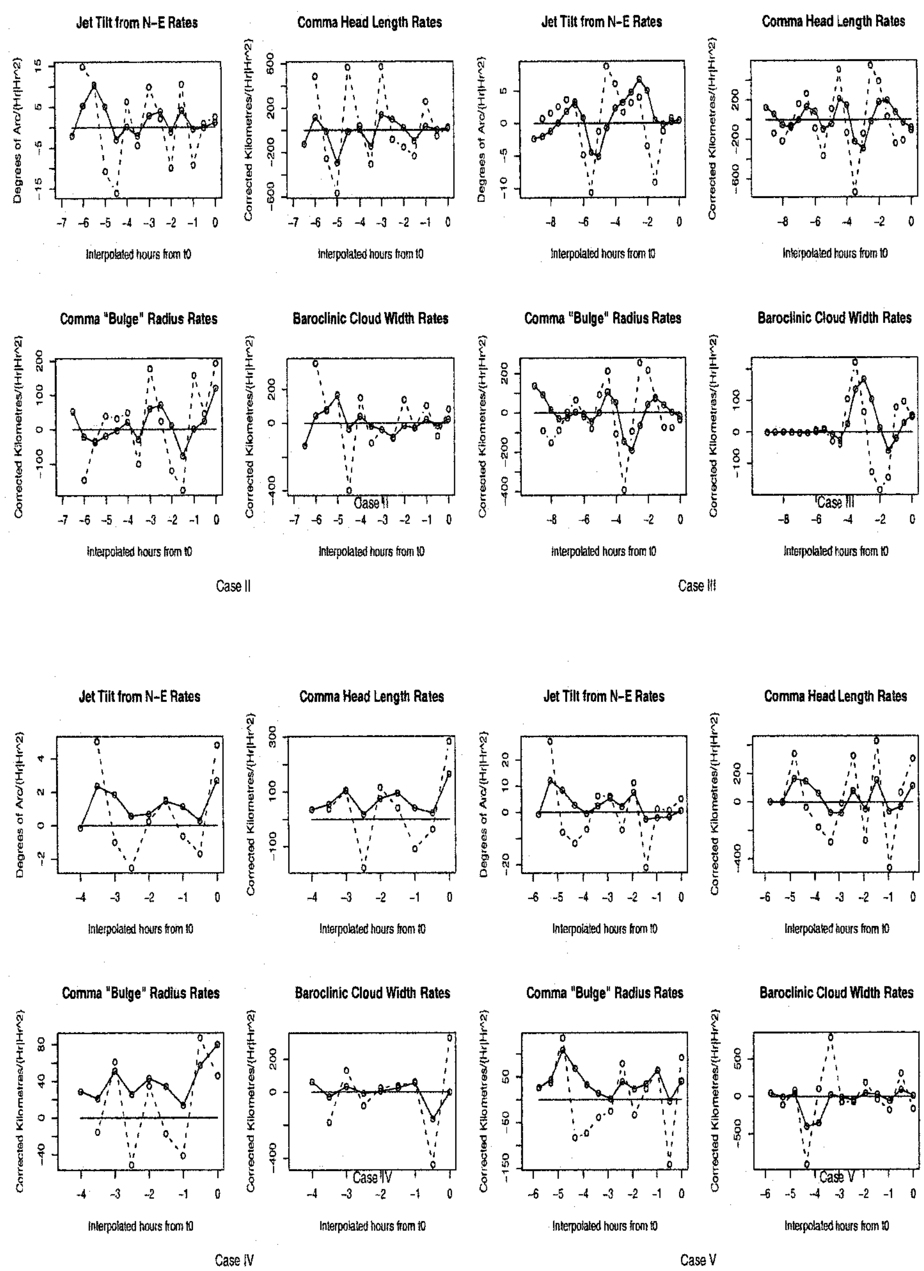


Fig. 5.21. First and second derivatives of geometric measures (solid and dashed lines respectively)–Cases II to IV. Rotation is in degrees of arc per hour, and per hour per hour. All other measures are in kilometres per hour, and per hour per hour.

5.3 Statistical Results

Raw results presented above suggested some trends in the raw data could be captured quantitatively. Statistics describing traditional and image-based methods will be considered in turn. Comparisons between variables will be considered for both within case and between case correlations, and where possible, linear regression models will be presented. For between case comparisons, only same–variable results will be considered. Analysis of aggregate measures, such as Bergeron number will also be shown.

Since the number of cases in the study was small, there was a risk of concluding an effect had been found based solely on the consequence of drawing random trials from a small sample. Wherever possible, corroborating evidence has been presented from results that have a logical tie in to the result being presented, or where the result is expected based on findings in the literature. It should also be noted that the sample of cases was not randomly drawn from the population of all winter storms: a necessity given the preliminary, and exploratory nature of the work. While some results appear strong, it is worth stressing again that the study is preliminary in nature, and results should be taken as suggestive rather than conclusive. All statistical results are presented in linear regression format $Y = (m \pm \Delta m)X + (b \pm \Delta b)$ in Table 5.11.

Table 5.11

: Regression variables and coefficients in order $Y = (m \pm \Delta m)X + (b \pm \Delta b)$.
Δm and Δb are the standard errors of the slope and intercept, respectively.
R^2, p and degrees of freedom are included.

Equ#	Response	Slope	StdErr Slope	Explanatory	Intercept	StdErr Intercept	R^2	p	D.F.
1	Chart.Bearing1	-1.1975	±0.3371	Hours	59.6812	±4.1009	0.7162	0.0163	5
2	Chart.Bearing2	-2.8513	±0.5599	Hours	30.1263	±10.5945	0.8121	0.0022	6
3	Chart.Bearing4	-1.3460	±0.5120	Hours	10.2280	±15.9620	0.6973	0.0784	3
4	Chart.Bearing5	1.6340	±0.4612	Hours	-63.7680	±14.1696	0.6765	0.0122	6
5	Chart.Bearing1	1.7984	±0.2229	Pressure1	-1717.0655	±219.6285	0.9559	0.0040	3
6	Chart.Bearing2	4.6810	±1.5300	Pressure2	-4584.8540	±1494.7510	0.7006	0.0377	4
7	Chart.Bearing4	5.4803	±0.3325	Pressure4	-5272.4997	±318.0168	0.9963	0.0386	1
8	Chart.Bearing5	-2.9260	±1.6760	Pressure5	2869.5810	±1654.6150	0.4325	0.1558	4
9	Pmin	6.1960	±1.8240	PIV-PII	825.5140	±43.4520	0.7936	0.0426	3
10	B24	0.0734	±0.0204	PIV-PII	-0.4554	±0.4774	0.8660	0.0694	2
11	Pressure1	-0.7817	±0.0810	Hours	986.4683	±1.1161	0.9395	7.074e-05	6
12	Pressure2	-0.7528	±0.0710	Hours	987.9722	±1.3091	0.9414	1.452e-05	7
13	Pressure4	-0.7238	±0.2212	Hours	981.8762	±6.3874	0.7281	0.0307	4
14	Pressure5	-0.4269	±0.1418	Hours	1001.6366	±4.1078	0.5644	0.0196	7
15	Pmean	0.8319	±0.1821	Pmin	174.7574	±177.0844	0.8744	0.0197	3
16	B24	0.0133	±0.0026	Pmean	-11.7535	±2.5350	0.9294	0.0359	2
17	Pmin	0.2308	±0.0693	CVmax	942.7555	±9.6676	0.7872	0.0447	3
18	B24	0.0051	±0.0023	CVmax	0.7161	±0.2451	0.7191	0.1520	2
19	Chart.Velocity2	-0.8862	±0.4083	Hours	66.8234	±7.7253	0.4399	0.0730	6
20	Chart.Velocity3	7.8100	±4.5220	Hours	129.8430	±28.6020	0.7489	0.3342	1
21	Chart.Velocity5	-1.2933	±0.6212	Hours	81.7986	±19.0832	0.4194	0.0825	6
22	Image.Bearing2	-2.3984	±0.9037	Hours	-3.1323	±8.9051	0.2929	0.0167	17

Table 5.11: Regressions Continued...

Equ#	Response	Slope	StdErr Slope	Explanatory	Intercept	StdErr Intercept	$R2$	p	D.F.
23	Image.Bearing4	-2.4997	±0.3081	Hours	33.9722	±5.4958	0.8246	1.162e-06	14
24	Image.Bearing5	-1.7783	±0.3476	Hours	55.2769	±8.2549	0.5668	5.269e-05	20
25	Image.Bearing1	2.7219	±0.3055	Pressure1	-2616.8571	±300.1224	0.9754	0.0124	2
26	Image.Bearing4	1.1762	±0.2138	Pressure4	-1167.4969	±206.0122	0.9098	0.0118	3
27	Pmin	0.4565	±0.0615	Image.BearingIII.IImax	922.9242	±6.9330	0.9483	0.0051	3
28	B24	0.0038	±8.087e-04	Image.BearingIII.IIdiff	1.2498	±0.0317	0.9158	0.0430	2
29	B24	0.0086	±0.0022	Image.BearingIII.IImax	0.4294	±0.2197	0.8792	0.0623	2
30	Image.Velocity2	-3.2080	±1.1750	Hours	98.6370	±11.5760	0.3049	0.0142	17
31	Image.Velocity5	-2.5526	±0.4437	Hours	122.3014	±10.5371	0.6233	1.251e-05	20
32	Image.Velocity2	7.0690	±0.9990	Pressure2	-6865.3390	±978.5950	0.9435	0.0058	3
33	Image.Velocity3	11.4380	±4.4490	Pressure3	-11323.6710	±4438.3610	0.7677	0.1238	2
34	Image.Velocity5	5.5094	±0.9317	Pressure5	-5402.6808	±924.6606	0.8535	0.0010	6
35	B24	0.0075	±0.0012	Image.VelocityIII.IImean	0.4334	±0.1394	0.9472	0.0267	2
36	Head.Length4	64.5970	±3.3600	Hours	993.1700	±8.9680	0.9788	5.553e-08	8
37	Head.Length5	31.0320	±6.1440	Hours	688.3450	±22.4310	0.6987	3.717e-04	11
38	Comma.Bulge1	11.3710	±2.9790	Hours	498.7120	±17.6240	0.6182	0.0041	9
39	Comma.Bulge4	36.8700	±1.9400	Hours	450.3040	±5.1780	0.9783	6.074e-08	8
40	Comma.Bulge5	36.9180	±2.4480	Hours	445.5640	±8.9350	0.9539	1.075e-08	11
41	Jet.Tilt1	1.1820	±0.1099	Hours	44.9772	±0.6504	0.9278	1.953e-06	9
42	Jet.Tilt2	1.6513	±0.2467	Hours	105.7418	±1.0148	0.7750	1.488e-05	13
43	Jet.Tilt3	0.9049	±0.3546	Hours	71.7283	±1.9542	0.4488	0.0341	8
44	Jet.Tilt4	1.1683	±0.0691	Hours	108.1334	±0.1844	0.9728	1.515e-07	8
45	Jet.Tilt5	2.7536	±0.4151	Hours	104.3869	±1.5153	0.8000	3.690e-05	11
46	Baroclinic.Width1	4.6083	±0.9204	Hours	122.8502	±5.4452	0.7358	7.321e-04	9
47	Baroclinic.Width3	17.2100	±10.8500	Hours	217.8800	±21.1800	0.5574	0.2534	2

Table 5.11: Regressions Continued...

Equ#	Response	Slope	StdErr Slope	Explanatory	Intercept	StdErr Intercept	$R2$	p	D.F.
48	Baroclinic.Width5	-60.8500	±14.2100	Hours	14.8800	±51.8800	0.6250	0.0013	11
49	Pmin	-0.4098	±0.1225	Baroclinic.WidthIII.IImin	1007.3161	±12.2279	0.8483	0.0790	2
50	B24	0.0377	±0.0108	Jet.TiltIII.IIdiff	0.8313	±0.1277	0.8598	0.0728	2
51	High.Cloud2	-2133.4000	±665.2000	Pressure2	2131724.0000	±651674.8000	0.7742	0.0491	3
52	High.Cloud4	622.3500	±94.5400	Pressure4	-567586.7700	±91084.1800	0.9353	0.0071	3
53	Cyclonic.Index2	-71.6000	±32.2800	Pressure2	70389.0000	±31622.4800	0.6212	0.1133	3
54	Edge.Mean2	0.3826	±0.0496	Hours	67.8700	±0.7177	0.4665	7.348e-11	68
55	Edge.Mean3	-0.8430	±0.0996	Hours	59.9231	±0.6307	0.7902	7.234e-08	19
56	Edge.Mean4	0.2737	±0.0213	Hours	46.2976	±0.4936	0.6416	0.000e+00	92
57	Edge.StDev3	-0.9017	±0.1406	Hours	136.2496	±0.8898	0.6841	3.771e-06	19
58	Edge.StDev4	0.3912	±0.0313	Hours	118.8226	±0.7248	0.6291	0.000e+00	92
59	Edge.Mean2	-1.5005	±0.2673	Pressure2	1542.9573	±261.8734	0.9131	0.0112	3
60	Pmin	-0.1775	±0.0558	Edge.MaxIII.IIdiff	974.6393	±3.8216	0.7714	0.0500	3
61	Comma.Bulge1	4.9135	±0.7265	Edge.Mean1	113.1530	±48.0397	0.8673	2.618e-04	7
62	Comma.Bulge2	12.5350	±2.8050	Edge.Mean2	-338.1400	±187.7720	0.6448	9.494e-04	11
63	Comma.Bulge3	24.3320	±7.4520	Edge.Mean3	-929.4100	±486.5530	0.6399	0.0171	6
64	Comma.Bulge4	-85.1400	±28.5100	Edge.Mean4	4269.7900	±1307.6900	0.5979	0.0244	6
65	Comma.Bulge5	37.5220	±6.7550	Edge.Mean5	-1656.6460	±358.6540	0.7742	3.543e-04	9
66	Jet.Tilt1	0.3385	±0.0622	Edge.Mean1	17.0418	±4.1130	0.8088	9.637e-04	7
67	Jet.Tilt3	-1.8589	±0.1735	Edge.Mean3	188.1842	±11.3307	0.9503	3.909e-05	6
68	Jet.Tilt4	-2.6630	±1.0390	Edge.Mean4	227.6170	±47.6390	0.5229	0.0427	6
69	Jet.Tilt5	3.0145	±0.6816	Edge.Mean5	-63.4143	±36.1883	0.6849	0.0017	9
70	Comma.Bulge1	11.6740	±2.9330	Jet.Tilt1	-22.7440	±115.4610	0.6936	0.0053	7
71	Comma.Bulge3	-12.1620	±4.2140	Jet.Tilt3	1471.7880	±282.2780	0.5813	0.0278	6
72	Comma.Bulge4	28.9330	±3.0770	Jet.Tilt4	-2686.6240	±324.4430	0.9365	8.212e-05	6

Table 5.11: Regressions Continued...

Equ#	Response	Slope	StdErr Slope	Explanatory	Intercept	StdErr Intercept	$R2$	p	D.F.
73	Comma.Bulge5	10.7820	±1.5210	Jet.Tilt5	-706.4640	±147.1090	0.8481	5.734e-05	9
74	Edge.Mean1	-0.0031	±0.0016	High.Cloud1	113.7940	±23.7610	0.1785	0.0715	17
75	Edge.Mean2	5.950e-04	±4.022e-05	High.Cloud2	47.4000	±1.6940	0.7683	0.000e+00	66
76	Edge.Mean3	0.0010	±2.221e-04	High.Cloud3	22.8900	±7.8670	0.5669	1.989e-04	17
77	Edge.Mean4	-4.435e-04	±7.301e-05	High.Cloud4	65.7900	±2.3950	0.2907	2.934e-08	90
78	Edge.Mean1	0.0065	±9.962e-04	Cyclonic.Index1	54.4300	±2.4630	0.7137	5.342e-06	17
79	Edge.Mean3	0.3894	±0.0894	Cyclonic.Index3	40.8691	±4.4401	0.5276	4.285e-04	17

5.3.1 Traditional Methods

Traditional meteorological predictive methods are primarily based on tracking changes in surface and mid–level variables. This mostly consists of charting deterministic variables such as pressure, temperature, vorticity etc. on 1000, 500 hPa and other upper–level charts. Pressure is the primary physical variable as measured from the low center on surface charts. Since pressure falls are the traditional basis for qualifying cyclones as Bombs, pressure is the fundamental physical measure. Thus, correlations to Pressure, and its time–based derivative, Bergeron Value, were of particular interest.

Geographic Dependency:

The geographic measures presented here relate to the ground–track of each system. Chart bearing is measured in degrees clockwise from north. Meridional systems will have an absolute bearing value close to zero, or a negative bearing if the system has tilted west of north.

- *Observation Level Analysis:*

Chart Bearing was strongly correlated with time for Cases I, II, IV and V: Table 5.11 equations 1-4; and Figures 5.4 and 5.6. (The negative correlation for Case V appears to be unusual.) Further, Chart Bearing correlates with Pressure over the length of Cases I, II, IV and V: Table 5.11 equations 5-7; Figure 5.22, and Table 5.11 equation 8 (same figure), which is not particularly significant, but worth investigation with a larger sample of cases. (Once again, Case V shows a negative correlation.) There were not sufficient overlapping observation points to provide results for Case III.

Length of Development:

- *Case Level Analysis:*

The length of time between the Phase II and Phase IV interval proved to be a good predictor of both the lowest pressure and the 24 hour Bergeron value. Phase IV-II length predicts minimum pressure: Table 5.11 equation 9; Figure 5.23*A*. Furthermore, it predicts Bergeron Value: Table 5.11 equation 10; Figure 5.23*B*. (Since case level models are based on five storms, the maximum degrees of freedom is three–one for the model, one for the error term, and three for the regression. Some cases had missing values, leading to fewer degrees of freedom.)

Strength of Development:

- *Observation Level Analysis:*

Pressure correlates with time for Cases I, II, IV and V. Case III had only four observation points, so the lack of correlation for this case is perhaps not significant: Table 5.11 equations 11-14; time plots in Figures 5.2 and 5.8.

- *Case Level Analysis:*

Certain strong correlations on Pressure and Bergeron Index were found. As might be expected, the overall Mean Pressure is related to Minimum Pressure: Table 5.11 equation 15; Figure 5.23*C*. Of more interest is that Mean Pressure predicts the 24 hour Bergeron Value: Table 5.11 equation 16; Figure 5.23*D*.

Ground Speed:

- *Case Level Analysis:*

Maximum Chart Velocity relates to both Minimum Pressure and Bergeron Value with: Table 5.11 equations 17 and 18; Figures 5.24*A* and *B*. While the p value for correlation with Bergeron Value is not clearly statistically significant, the suggested relationship is of interest for investigation with a larger sample of storms.

- *Observation Level Analysis:*

There is a notable time dependency for Chart Velocity in Cases II, III and V. Regression results yield: Table 5.11 equation 19, and 20, which are strong but not particularly significant, and Table 5.11 equation 21; time plots in Figure 5.6. Results were not significant for the other two cases. While there is generally a slight drop in Chart Velocity over time, the variable is not linear over time.

5.3.2 Image–Based Methods

Geographic Dependency:

- *Observation Level Analysis:*

Image Bearing shows a strong time dependency for Cases II, IV and V. Regression results yield: Table 5.11 equations 22-24; time plots in Figure 5.6. There is a large outlier of -103.4 at t+1 for Case I which renders the regression non–significant. Without this outlier, the trend is negative with $R^2 = 0.1915, p = 0.0609$ on 17 degrees of freedom. Image Bearing also predicts Pressure for Cases I and IV: Table 5.11 equations 25-26; Figure 5.24*C-D*. Results were weak or non–significant for the other three cases.

- *Phase II to Phase III Case Level Analysis:*

A strong correlation was found between the maximum Image Bearing value over the Phase II to Phase III interval and minimum pressure for the entire case: Table 5.11 equation 27; Figure 5.25*A*. Furthermore, a strong correlation was found between the difference between the Phase III and Phase II Image Bearing values and the 24 hour Bergeron value: Table 5.11 equation 28; Figure 5.25*B*. A slightly weaker relationship was found between the maximum Phase II to Phase III Image Bearing value and 24 hour Bergeron value: Table 5.11 equation 29; Figure 5.25*C*.

Ground Speed:

- *Observation Level Analysis:*

A dependency to hourly time index is shown for Cases II and V for Image Velocity: Table 5.11 equations 30-31; time plots in Figure 5.6. No strong correlation is evident for the other three cases. Image Velocity is also related to Pressure for Cases II, III and V: Table 5.11 equations 32-34; Figures 5.25*D* and 5.26*A-B*. The p value for Case III is not small, but worth consideration in this context. Results for Cases I and IV were not significant, although Case IV did have a large R^2value (0.5377 with p=0.2667). For all these relations, Image Velocity shows some systematic trends but is generally non–linear. For example, Image Velocity seems to generally vary directly with pressure, although the exact relationship is too complex to be predicted from a linear regression.

- *Phase II to Phase III Case Level Analysis:*

The mean Image Velocity over the Phase II to Phase III interval correlated very strongly with the 24 hour Bergeron Value: Table 5.11 equation 35; Figure 5.26*C*.

Expansion of Comma–Head:

- *Observation Level Analysis:*

Comma Head Length shows strong correlation to the hourly time index for Cases IV and V: Table 5.11 equations 36-37; time plots in Figure 5.20. Results are weak for the other three cases. Comma Bulge Radius also shows strong time dependence for cases I, IV and V: Table 5.11 equations 38-40; time plots in Figures 5.18 and 5.20. Results for the other two cases were not statistically significant.

Change in Jet Axis:

- *Observation Level Analysis:*

Changes in the vector tracking of Jet features show significant time correlations. All five cases show strong relationships between time and Jet Tilt: Table 5.11 equations 41-45; time plots in Figures 5.18 and 5.20. The weaker results for Case III may be significant since this case likely did not develop explosively. Although these results are very promising, it should be noted that Jet Tilt is not always a strictly linear relationship.

While Baroclinic Width seemed difficult to measure consistently, there were some time correlations on this variable. Case I had the strongest dependency: Table 5.11

equation 46; time plot in Figure 5.18. The variable was first defined based on this test case, for which it could be readily identified. Case III shows a dependency as well: Table 5.11 equation 47; time plot in Figure 5.20, which is not particularly statistically significant. Case V shows a fairly strong relationship: Table 5.11 equation 48 (same figure), however, as can be seen from the plot versus time, the feature selected as representative of the Baroclinic Width switched part–way through the analysis; that is to say, the two features tracked may have been physically related, but were not the *same* feature. Results for the other two cases were weak.

- *Phase II to Phase III Case Level Analysis:*

Given the difficulties in measuring Baroclinic Width consistently, a surprisingly strong correlation was found between the minimum Baroclinic Width over this interval and the minimum case Pressure: Table 5.11 equation 49; Figure 5.26*D*. A very similar correlation was found between the Phase III and Phase II Jet Tilt difference and 24 hour Bergeron value: Table 5.11 equation 50; Figure 5.26*A*.

Amount of High Cloud:

- *Observation Level Analysis:*

Correlations for High Cloud count versus time were all fairly strong, but none of them were statistically significant. High Cloud is clearly a non–linear feature over time (Figures 5.12, 5.15), so observation level regression analysis of time correlations is not appropriate for this variable. Some significant correlations were found between High Cloud and Pressure. Results for Cases II and IV yield: Table 5.11 equations 51-52; Figure 5.27*B-C*. Results for Cases I, III and V were not significant. Cyclonic Index (based on High Cloud values) shows a correlation with Pressure for Case II: Table 5.11 equation 53; Figure 5.27*D*. Results for the remaining cases were moderately to fairly strong, but not statistically significant.

Sharpening Edges:

- *Observation Level Analysis:*

As seen in Chapter 5 (Figures 5.14 and 5.17), Edge Sharpening variables (mean, standard deviation and maximum) are not entirely linear over time; however, there is significant linear time correlation for Cases II, III and IV. Regression results versus Edge Mean values for Cases II through IV yield: Table 5.11 equations 54-56; Figure 5.17. The plot becomes non–linear due to a large drop in values after t+20 hours. Correlations with Cases I and V were fairly large but not statistically significant. Similar results were found for standard deviations for Cases III and IV: Table 5.11 equations 57-58; same figure. The shape of the standard deviation curves closely follow those of the mean. Pressure also predicts of Mean Edge Strength for Case II: Table 5.11 equation 59; Figure 5.28*A*. Results for the other cases were not of interest.

- *Phase II to Phase III Case Level Analysis:*

A good correlation was found between the difference between the Phase III and Phase II values of Maximum Edge and minimum case pressure: Table 5.11 equation 60; Figure 5.28*B*.

Other Correlations:

Correlations between variables other than time, Pressure and Bergeron Value are also occasionally of interest, especially when unexpected or where the two variables are derived from independent methods.

- *Observation Level Analysis:*

An interesting correlation was observed between Mean Edge Strength and Comma Bulge Radius. These two variables are independently generated; generation of the edge strength variables is also entirely automated. Results for all five Cases yield: Table 5.11 equations 61-65; Figures 5.28*C-D* and 5.29*A-C*. Mean Edge Strength also predicts Jet Tilt for all but Case II, which was weak and not statistically significant: Table 5.11 equations 66-69; Figures 5.29*D* and 5.30*A-C*.

As would be expected, Jet Tilt also predicts Comma Bulge (since Tilt and Mean Edge are correlated): Table 5.11 equations 70-73; Figures 5.30D and 5.31A-C. A similar pattern was found for High Cloud counts predicting Mean Edge Strength for all but Case V: Table 5.11 equations 74-77; Figures 5.31D and 5.32A-C. Finally, Cyclonic Index is correlated with Mean Edge Strength for Cases I and III: Table 5.11 equations 78-79; Figures 5.32D and 5.33. Clearly, Mean Edge Strength is a highly important measure for prediction.

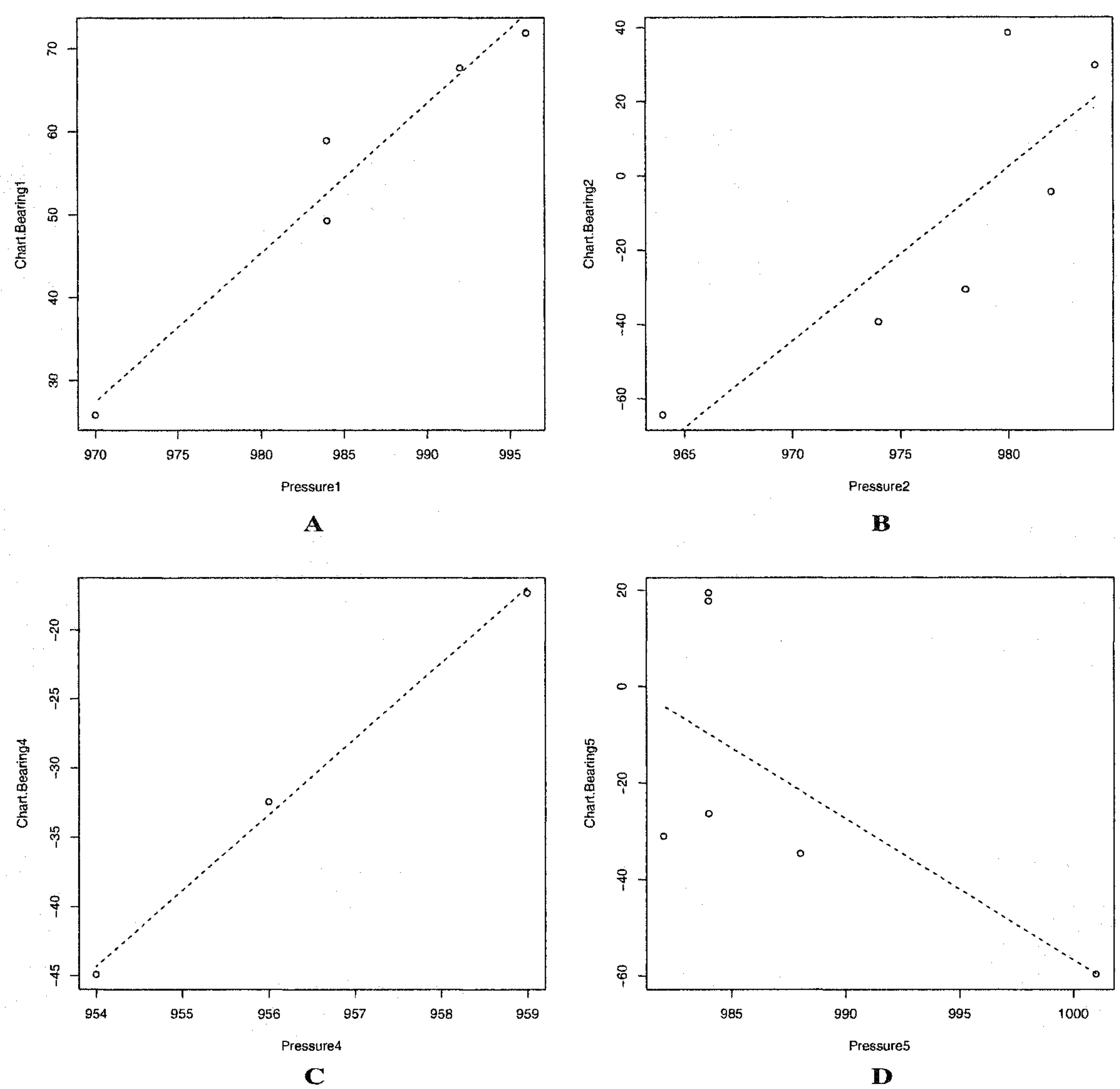


Fig. 5.22. Regression plots for equations 5, 6, 7 and 8 (panels A, B, C and D)

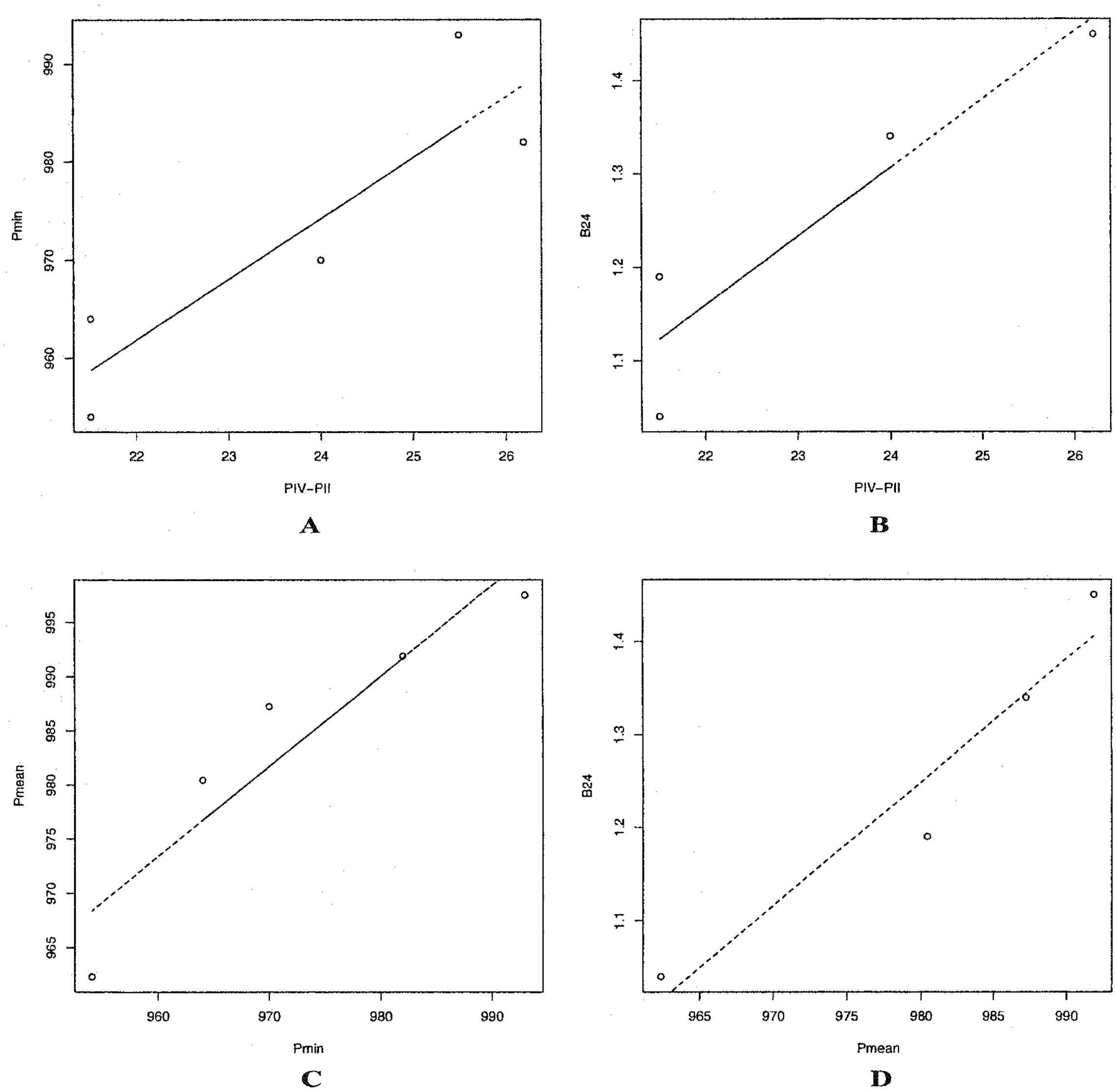


Fig. 5.23. Regression plots for equations 9, 10, 15 and 16 (panels A, B, C and D)

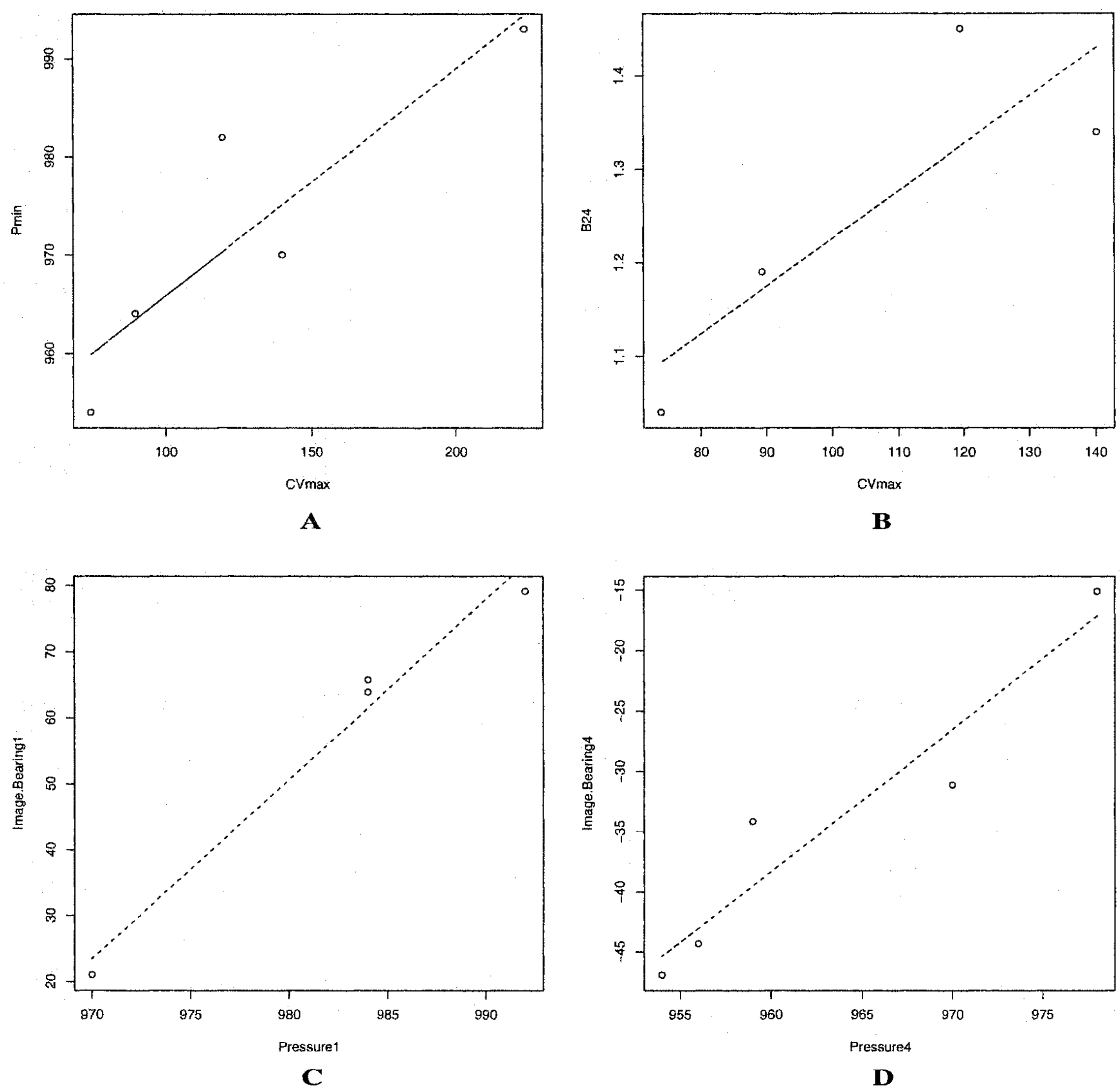


Fig. 5.24. Regression plots for equations 17, 18, 25 and 26 (panels A, B, C and D)

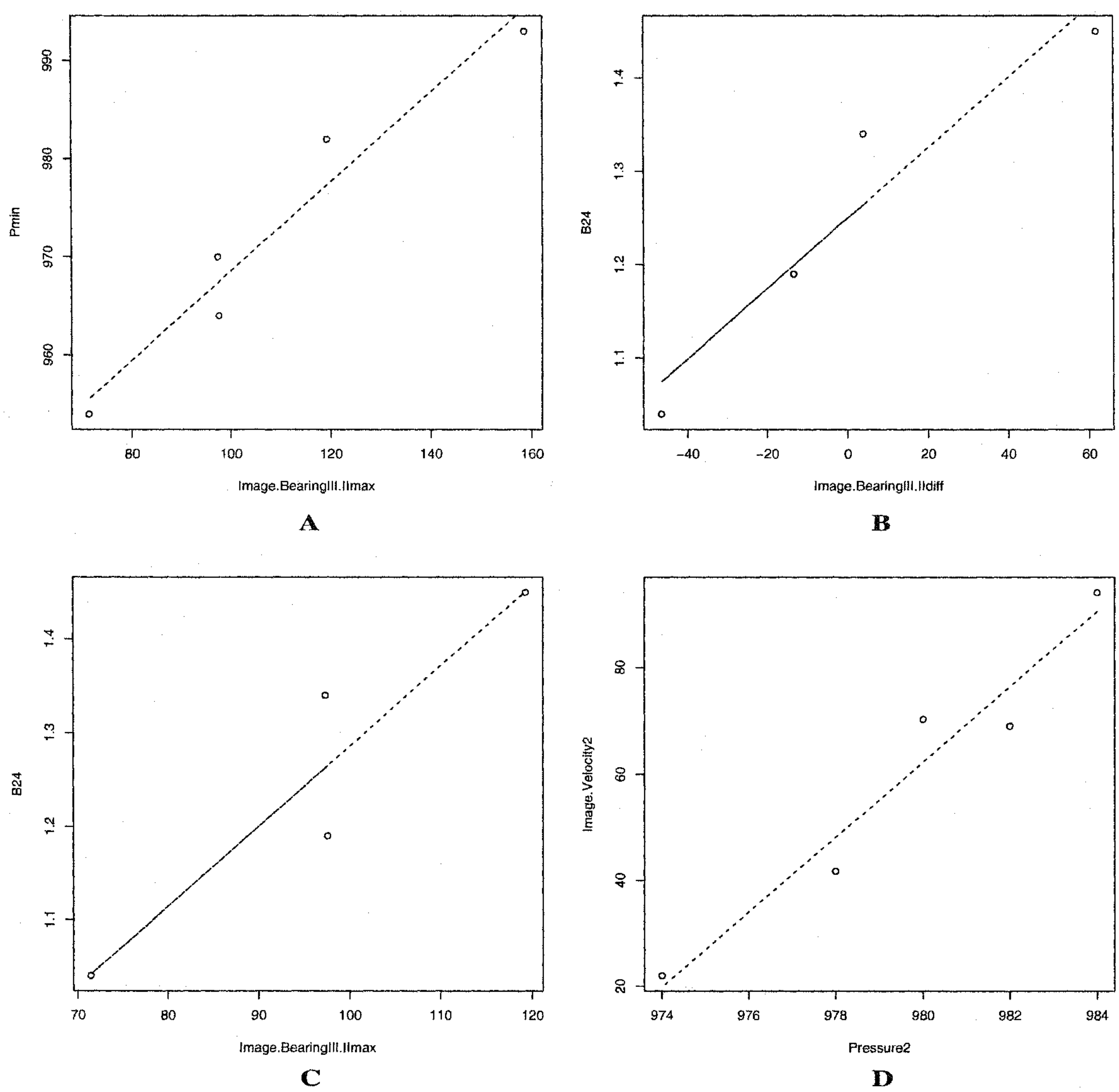


Fig. 5.25. Regression plots for equations 27, 28, 29 and 32 (panels A, B, C and D)

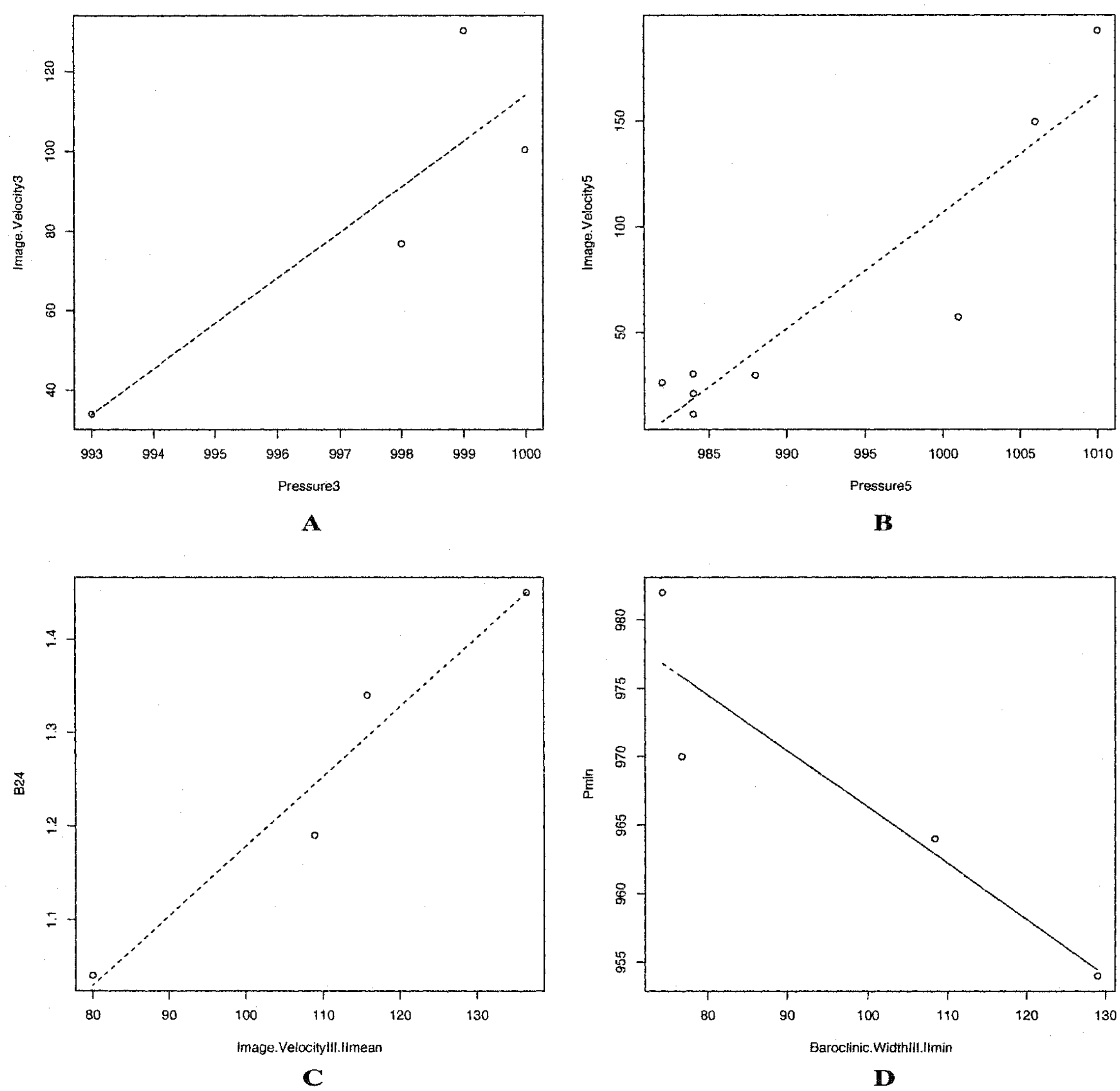


Fig. 5.26. Regression plots for equations 33, 34, 35 and 49 (panels A, B, C and D)

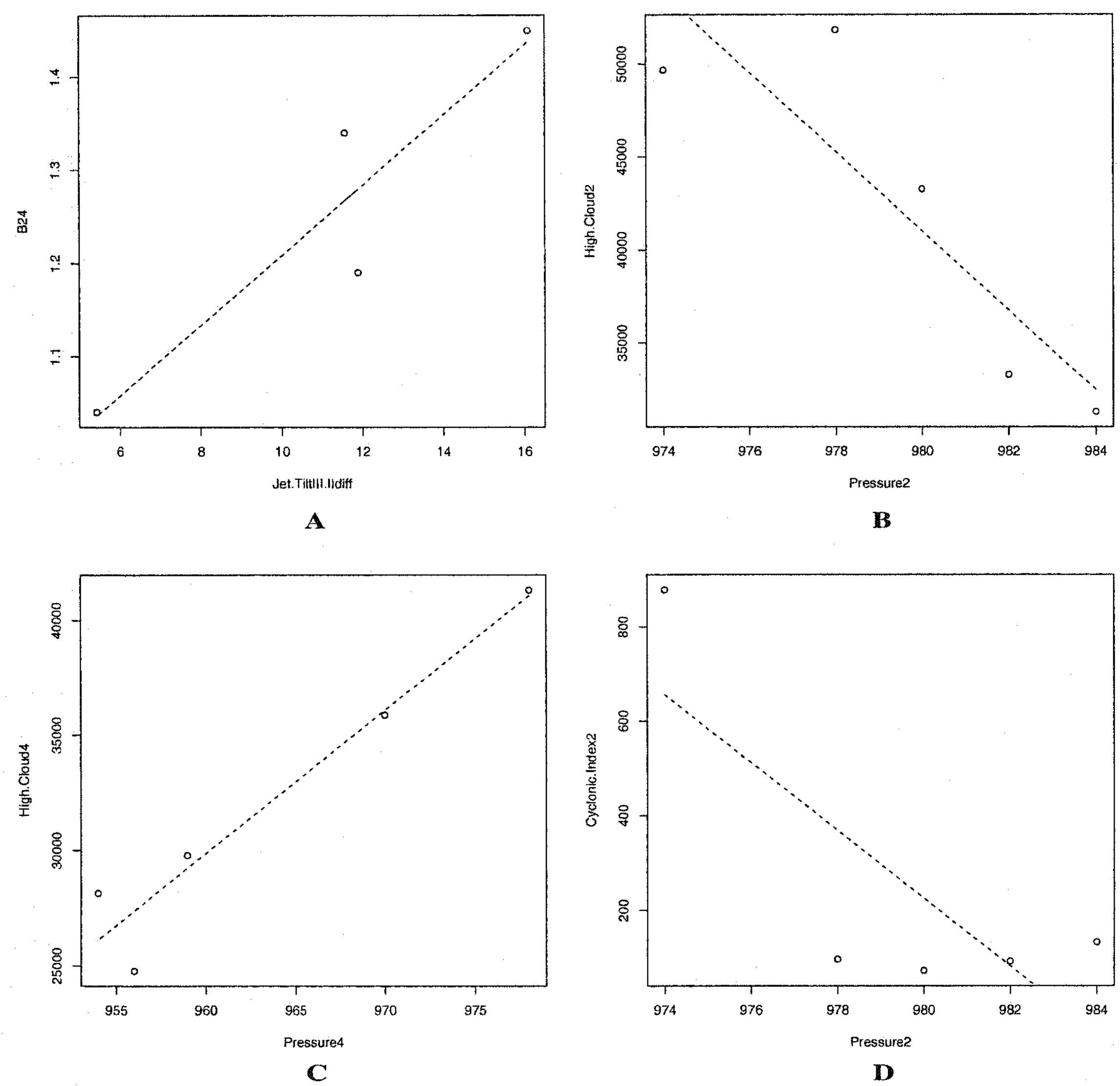


Fig. 5.27. Regression plots for equations 50, 51, 52 and 53 (panels A, B, C and D)

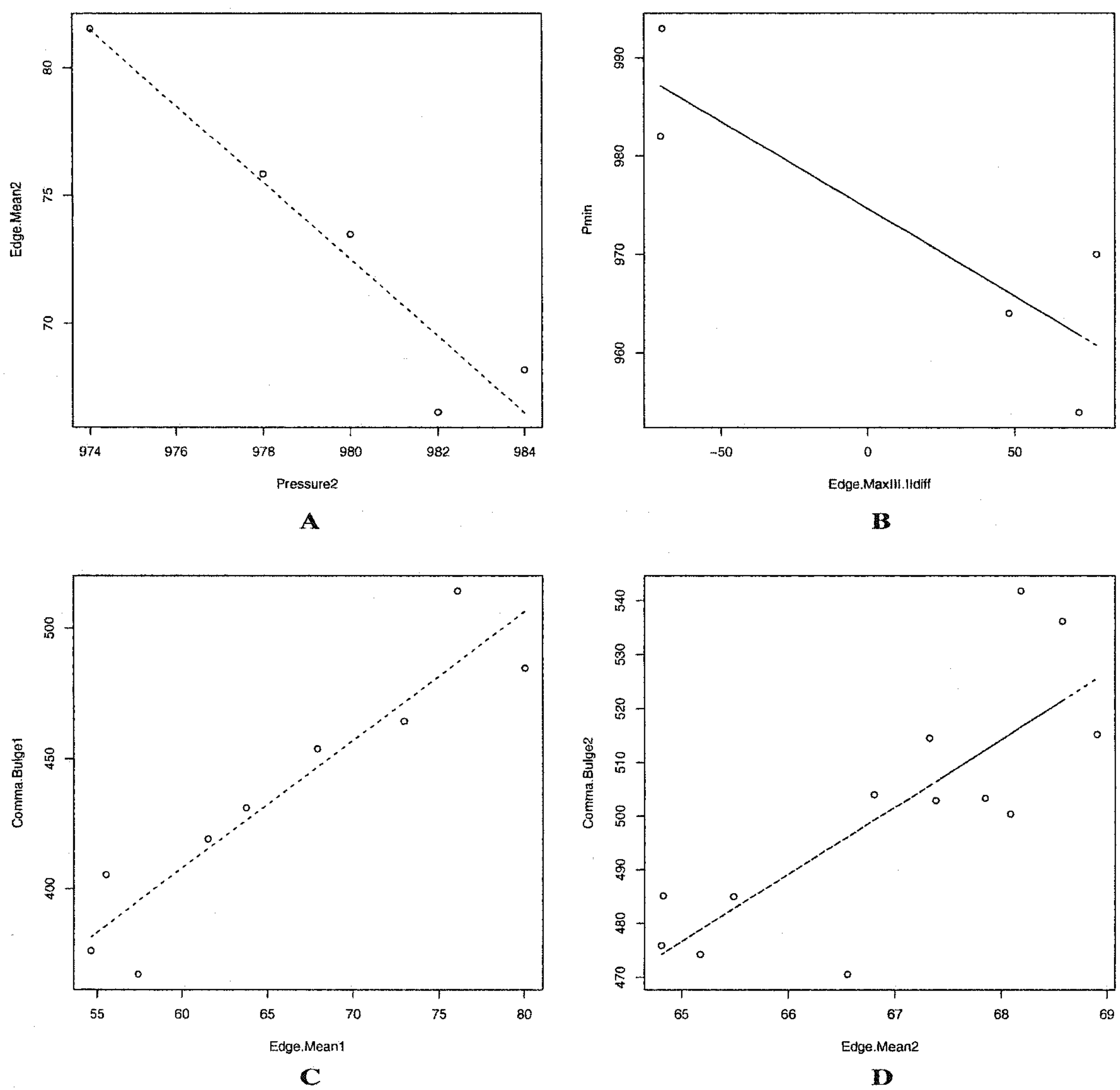


Fig. 5.28. Regression plots for equations 59, 60, 61 and 62 (panels A, B, C and D)

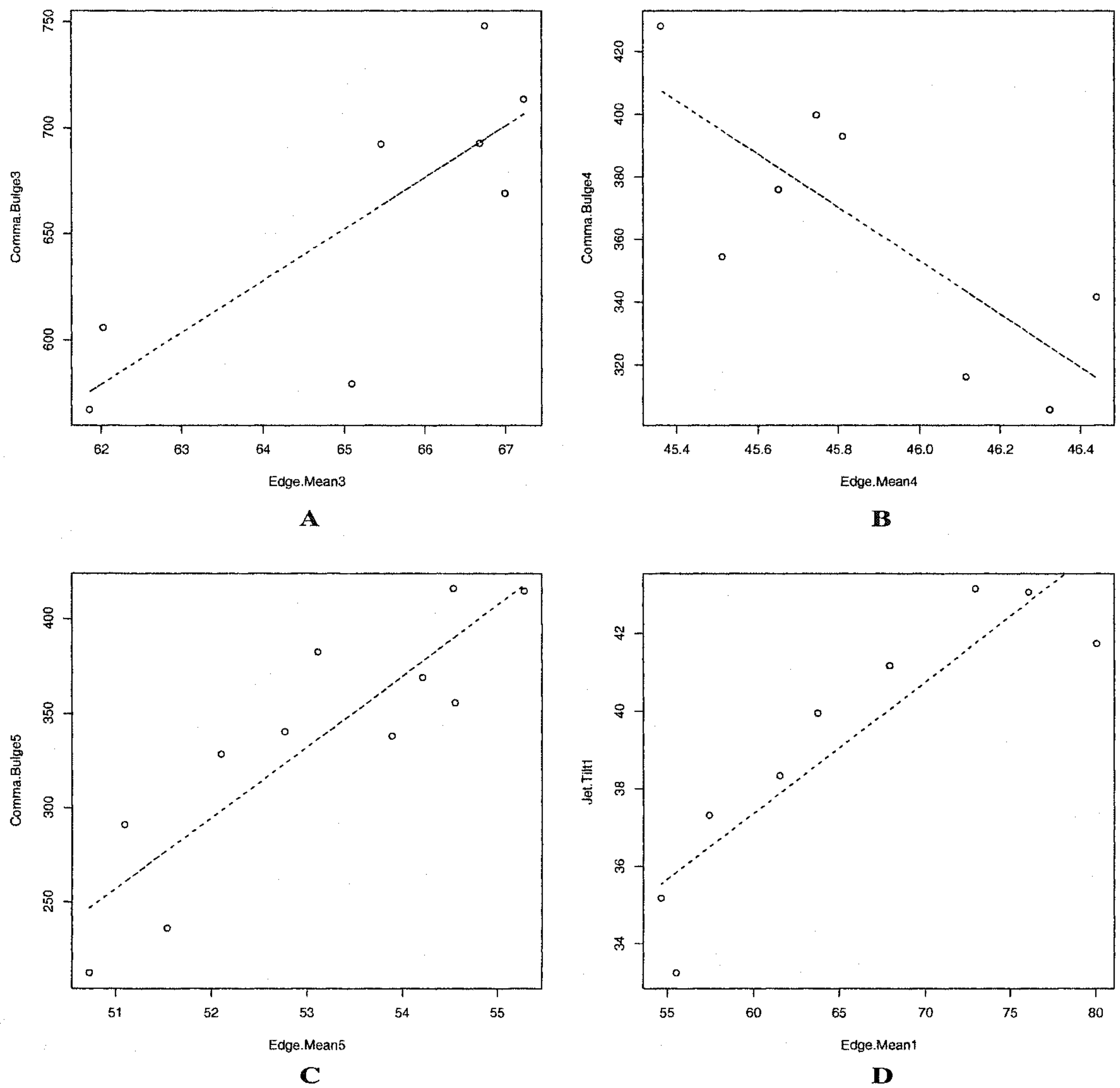


Fig. 5.29. Regression plots for equations 63, 64, 65 and 66 (panels A, B, C and D)

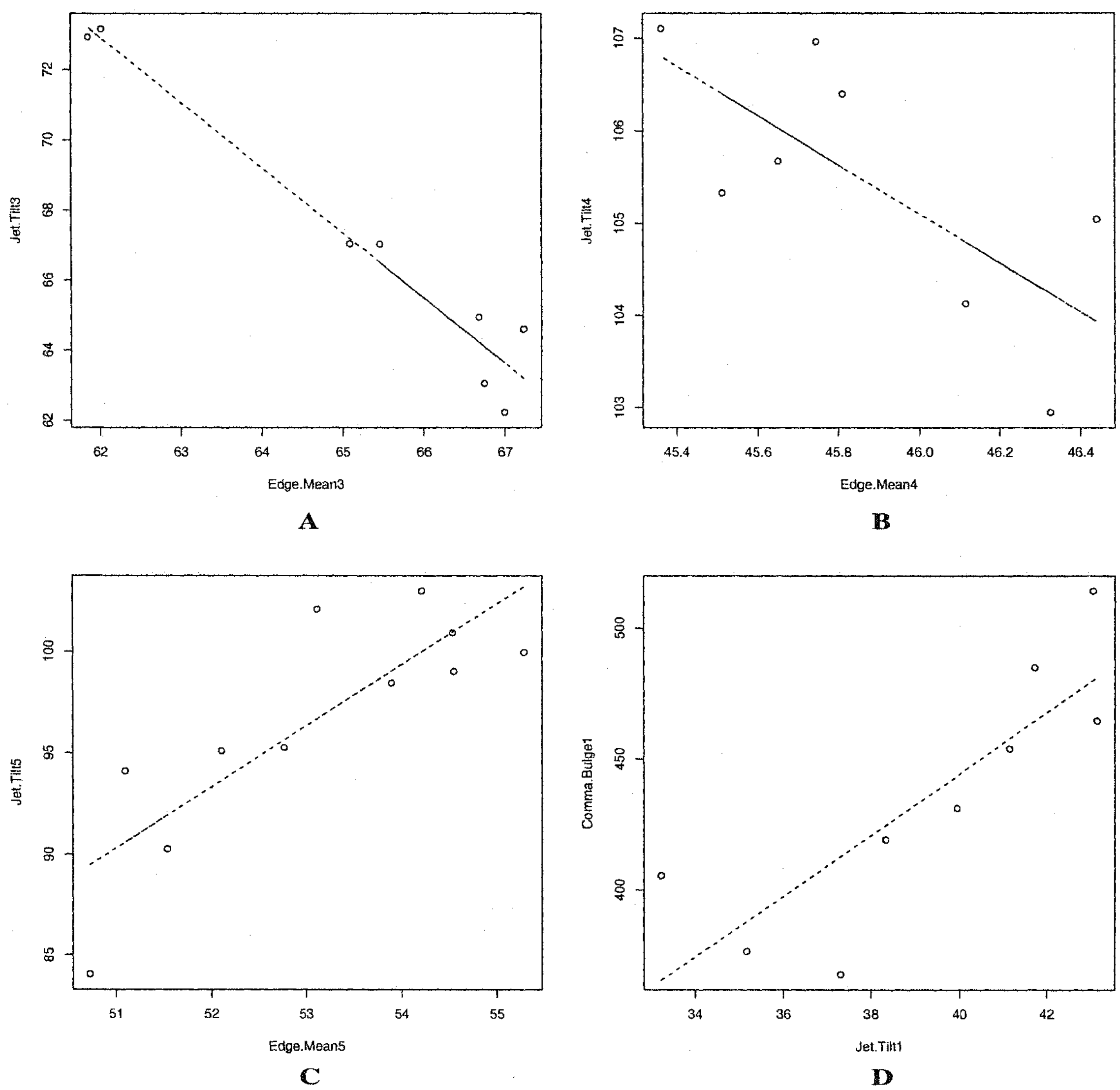


Fig. 5.30. Regression plots for equations 67, 68, 69 and 70 (panels A, B, C and D)

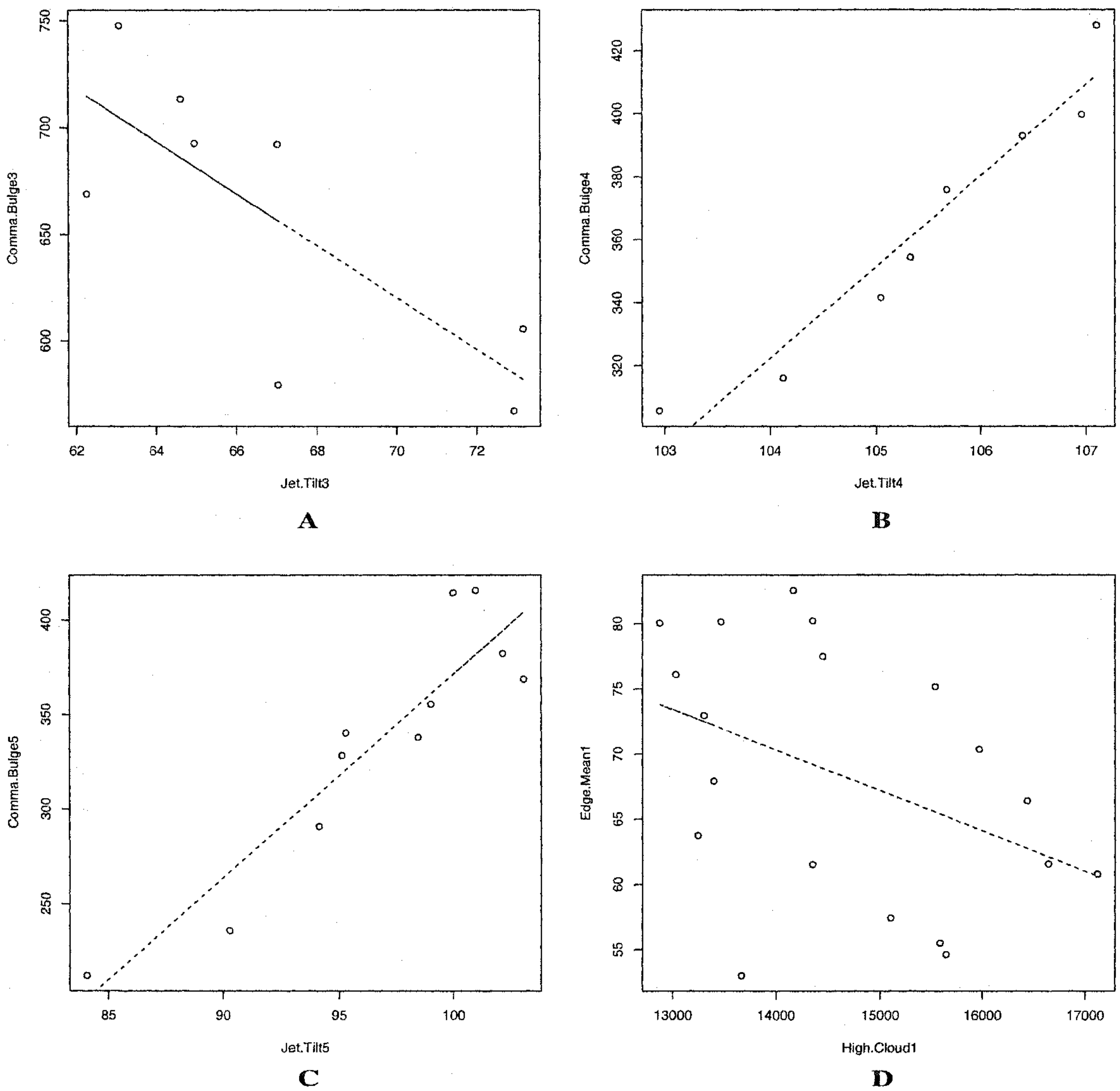


Fig. 5.31. Regression plots for equations 71, 72, 73 and 74 (panels A, B, C and D)

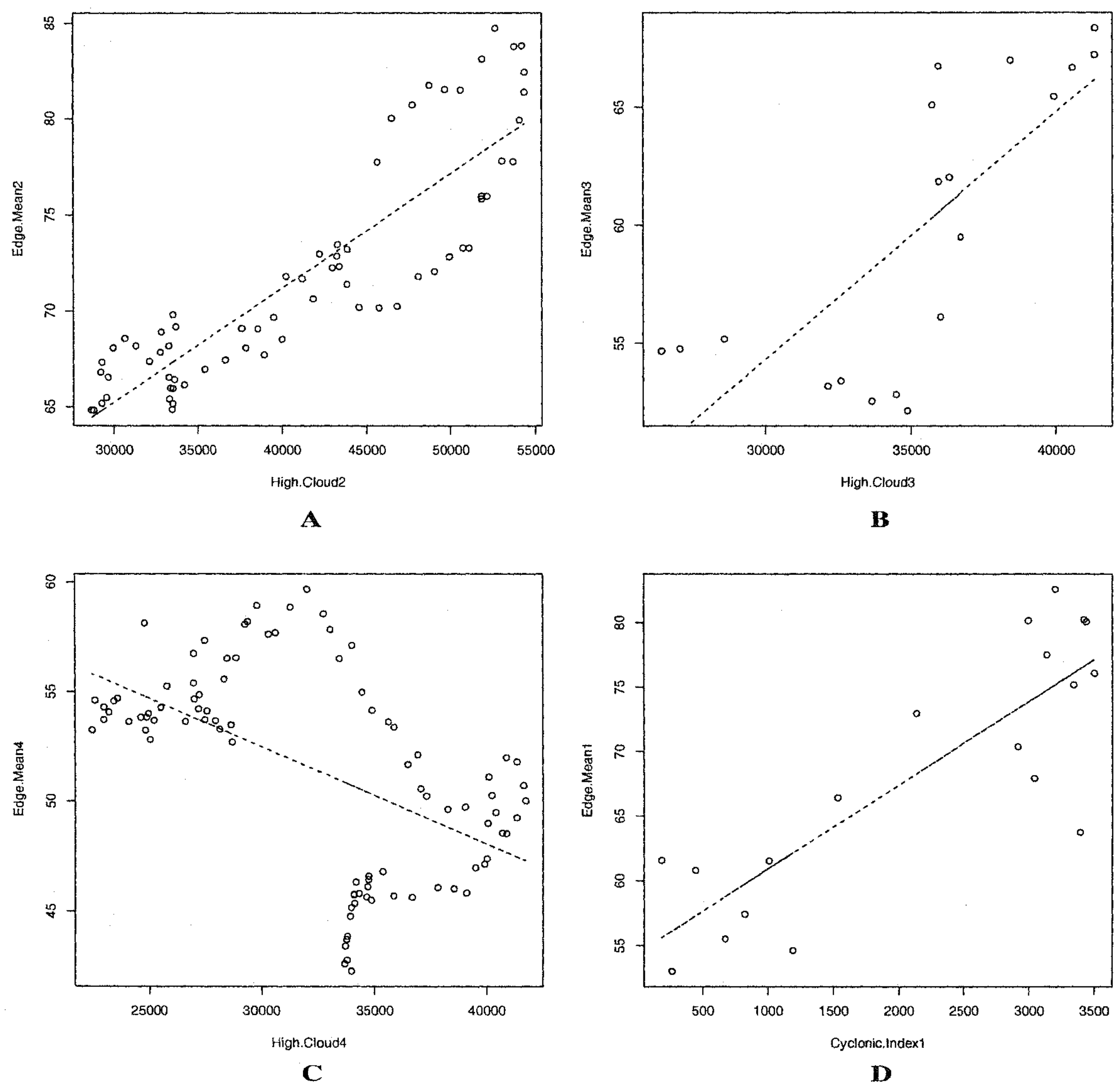


Fig. 5.32. Regression plots for equations 75, 76, 77 and 78 (panels A, B, C and D)

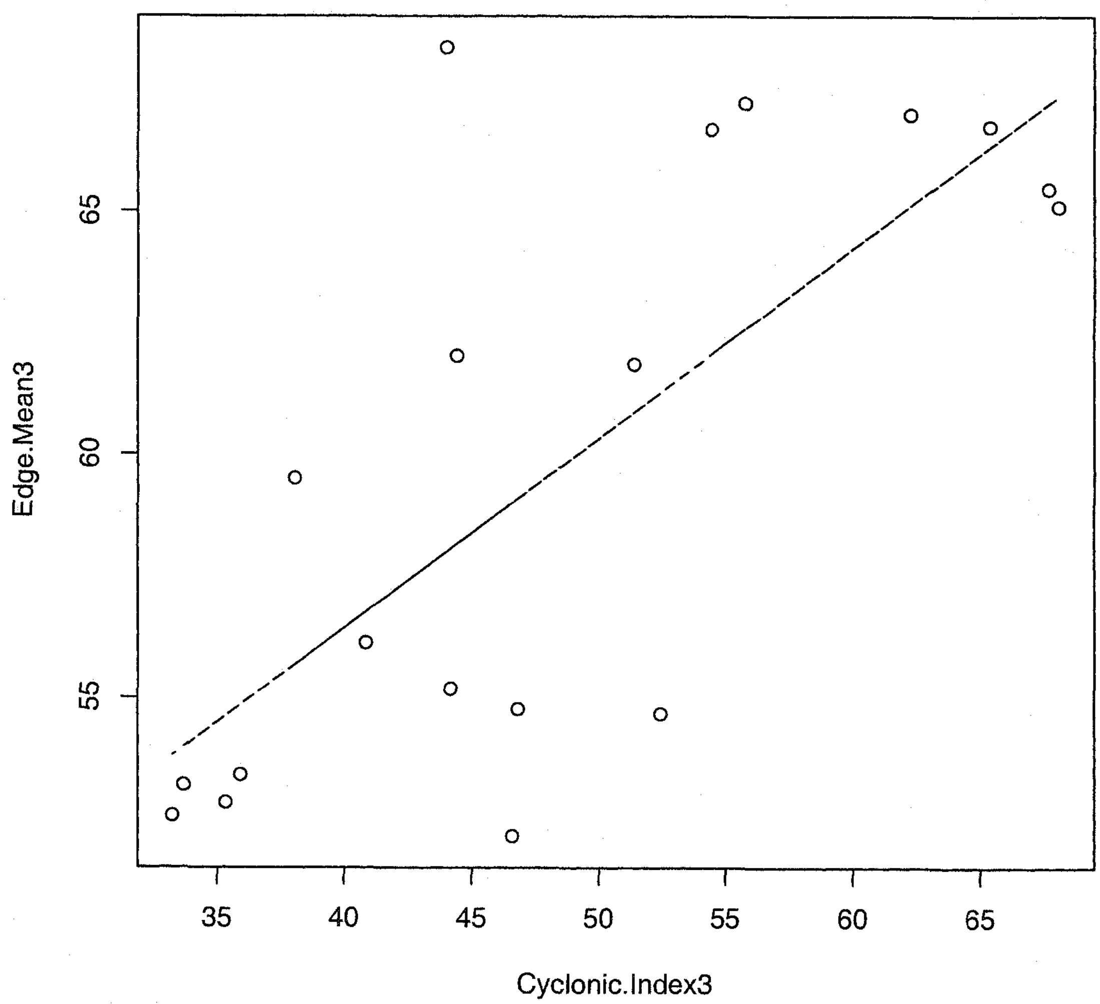


Fig. 5.33. Regression plot for equation 79

6. DISCUSSION

6.1 Introduction

Several findings in Chapter 5 show promise for application to the problems of detecting and predicting explosive cyclones. The two significant sources of data were those derived from mean sea–surface level charts, and from GOES–9 image sequences. In this section, the results are interpreted and classified by the methods that generated them. First, results that could be derived from either chart or image sources are presented. Next, those that could only be derived from charts alone are discussed. Finally, results that were derived from image information solely are considered. As in Chapter 5, reference to the Phase model presented in Chapter 2 helps guide the discussion.

6.2 Methods Applicable to both Chart and Image Data

Geographic Dependency:

For both Chart and Image based results, Bearing is directly proportional with Pressure. Bearing was recorded in degrees East of North, with negative values for Bearings West of North. In this context, the system tends more westward as Pressure drops. Sanders [38] found that strong cyclonic cases tend to be more meridional (follow a more north–south line), which may support the notion of systems turning westward as they deepen. Since both Bearing metrics and Pressure were found to vary inversely with time, it is not unexpected that the Bearing metrics vary directly with Pressure. It is important to note that Pressure is not really a linear variable; rather, Pressure generally falls over the intervals being investigated–values were not included for much time after the point of lowest Pressure, so the subsequent rise in values following the decay of the system is not expressed in these regressions. Since the correlation to time is the more relevant in this case, it might be better to state that the systems tend more westward over time.

Correlations to Pressure are stronger for variables derived from chart–based observations than those based on image observations. As could be seen in the plots of Bearing versus time (Chapter 5), there is a great deal more variability in the Image–based observations than those from the charts. Almost certainly, the fact that chart observations occur at six hour intervals accounts for the discrepancy. Smoothing of Image Velocity observations (Figures 5.4 and 5.6) made these plots more similar to their chart–based counterparts, and smoothing of the regression data would very likely increase the strength of image–based regressions. Similarly, results of Bearing versus time are generally stronger for chart–based observations. Image observations can be more precisely located than chart observations, since fairly precise guidance is provided from the grid method relative to the fairly coarse level isobars of the chart pressure field.

Predictive relationships were found based on the Phase II to Phase III interval. Phase times are estimated from the imagery. While similar information could also be derived from the pressure values (see the comparison of Phase models in Chapter 2), there would be little reason to do so, since the image–based method is so easy to implement. The maximum value of Image Bearing over the interval predicts both the minimum pressure value over the entire case and the 24 hour Bergeron value. The Image Bearing difference over this interval (the value at Phase III minus the value at Phase II) also predicts the 24 hour Bergeron value. Since the Phase II to Phase III interval also represents the length of the explosive deepening stage ([37], [31]), and the standard definition of Explosive Cyclogenesis incorporates the notion of rapid development ([39]), it is not surprising that the change in variables over this interval was found to be significant. If these predictive relationships hold for a larger sample of cases, they would be extremely useful. For the cases presented here, the start of Phase III was on average 22 hours prior to the time at which the deepest pressure was recorded (28 hours if the non–rapid deepening Case III is omitted). Such a lead–

time permits for prediction, which would be very useful for marine interests and may greatly reduce the hazard associated with rapidly deepening systems.

Ground Speed:

At the case level of analysis, there is a quite strong direct correlation between minimum Pressure and the maximum Chart Velocity. The notion that systems with higher maximum velocity had higher minimum pressures (they were weaker) is counter–intuitive, and seems to contradict the climatological findings that explosive systems move faster [38]. Although not statistically significant ($R^2 = 0.5176, p = 0.1707$) the minimum case Pressure also directly varies with maximum Image Velocity over the Phase II to Phase III interval. A similar result finds minimum Pressure directly varying with *mean* Image Velocity over this period ($R^2 = 0.5794, p = 0.135$), and case level results show Image Velocity varying directly with Pressure, all of which suggests the unexpected effect may be real, and is definitely worth further investigation.

A more expected, related finding is that the 24 hour Bergeron value directly correlates with the maximum Chart Velocity. This result is supported by a particularly strong direct correlation between 24 hour Bergeron value and mean Image Velocity over the Phase II to Phase III interval. These results suggests that systems that are faster are also more intense (have a higher rate of deepening). Since climatological studies have found that Bombs travel faster than other cyclones ([38]), it is perhaps not surprising that there is a linkage between maximum speed and storm intensity. The Phase II to Phase III result is particularly significant because of its potential application for prediction.

At the observation level, both Chart and Image Velocity vary inversely with time, indicating that generally systems seem to slow over time. These variables are not linear, and so the relationship should be viewed as expressing an average behaviour for the system only. From the time plots in Chapter 5, it can be seen that decreasing velocity over time appears to be a more consistent trend after about t+10 hours.

Length of Development:

An interesting relationship was found between both minimum Pressure and 24 hour Bergeron value versus the length of the interval between Phases II and IV. Since the Pressure and Bergeron values are derived from Chart information and the Phase estimates come from imagery, these relationships are not so much applicable to *either* Chart or Image data, but rather dependent on both. Both relationships are direct with this time interval. Since Phase II is the beginning and Phase IV the end of the deepening stage, the implication is that a long deepening period leads to intense (large Bergeron value) but weak (high minimum Pressure) systems. Inversely, a short deepening period relates to a strong (deep) low, but a low rate of deepening (small Bergeron value). A short interval might be expected to result in a deep low, since explosive events develop quickly; however, how this condition can be satisfied by a system with a low rate of pressure fall is not immediately obvious. Results for the 12 and 6 hour Bergeron value were both negative relationships, although both were very weak based on the available sample. It is possible that a high rate of deepening occurs over a shorter than 24 hour interval for strong systems, suggesting that the 24 hour standard (part of the "official" definition of a Rapid Deepener–cf. [39]) may not be appropriate. The entire mean length of the Phase II to Phase IV intervals was only 23.74 hours, and in Chapter 5 it was found that shorter interval Bergeron values consistently overestimate the 24 hour interval value. If the *entire* deepening period is only 24 hours, it is obvious that the most intense interval of deepening will be less than 24 hours. In any case, it appears that the Phase II to Phase IV interval length is a very descriptive measure and may be as significant to report as the minimum Pressure or Bergeron value when characterising a particular system.

6.3 Methods Applicable to Chart Data Only

Strength of Development:

Two relationships were found at the case level that are worthy of note: mean pressure over the entire case varies directly with the minimum case pressure, and the 24 hour Bergeron value varies directly with the mean pressure. The first relationship is not unexpected, since the minimum value is one of the observations comprising the mean, and pressure tends to be a smooth field. Hence, minimum pressure will generally not be an isolated value (an outlier), but will anchor one extreme of the range of the variable (See pressure plots versus time in Chapter 5). Whether a large mean pressure would be expected to result in an intense storm is less clear. It is possible that as with the Length of Development findings (above), the 24 hour Bergeron value is estimated on too long a time interval, and hence does not reflect the expected relationship.

6.4 Methods Applicable to Image Data Only

Expansion of Comma–Head:

Both measures of comma head expansion varied directly with time: Comma–Head Length showed strong correlations with time for Cases IV and V. Case IV in particular had a very strong correlation, and was clearly linear from the time plot. Case V was also linear. This is not clearly true for Cases I and II, but does appear to be for Case III, even though its results were non–significant. Similar comments apply to Comma Bulge Radius, with only Cases IV and V being clearly linear. Case I, which also returned statistically significant results, appears to increase linearly after t-7 hours. These results imply that generally, Comma Head size expands over time on both measures. The results are expected. Works such as Weldon's ([46]) indicate that expansion of the Comma Head is characteristic of development over time.

Change in Jet Axis:

Jet tilt was very linear over time for all but Case III. Strong correlations were found for all five cases. Case III had a significant, but notably weaker R^2 value. All five cases generally rotated at a fairly constant rate of approximately 1.5 deg/hr; which agrees with the notion that a building and increasingly tilting trough is typical of cyclonic development (cf. [27]). It would be interesting to see if deviations from this trend correspond to specific physical occurrences for cases with less linear time plots. Baroclinic width also had significant correlations with time for Cases I, III, and V;

however, as noted earlier, the feature could only be reliably identified for Case I. The feature could only be identified from t-4 for Case III, and the feature that was being tracked switched between t-4.5 and t-4 for Case IV.

Both metrics showed strong correlations to Phase II to Phase III interval values. Minimum Pressure varies inversely with the minimum Baroclinic Width across this interval, suggesting a deeper storm for a greater minimum width; however, as mentioned above, Baroclinic Width is a metric of doubtful reproducibility in general. The 24 hour Bergeron value varies directly with the last minus first Jet Tilt observation across the interval, suggesting a greater range of tilt leads to a more intense system. This finding is not unexpected, since Bombs become more meridional as the supporting jet structure amplifies (jet tilt to the West increases). Put another way, more meridional systems tend to be stronger. The relationship may have uses as a predictive function.

Amount of High Cloud:

At the observation level, two correlations of note were found between High Cloud count and Pressure for Cases II and IV. The relationship was inverse for Case II and direct for Case IV. As noted previously, High Cloud is not a linear relationship over time; however, Pressure over the recorded interval is a fairly linear relationship, bringing into question the applicability of linear regression for capturing trends in High Cloud count. By contrast, the plots generated for this relationship do appear linear (Figures 6.6*B* and *C*). Cyclonic Index also correlated to Pressure for Case II. The relationship was inverse, indicating lower pressure for a higher Cyclonic Index, which would make sense since Cyclonic Index was derived to return a high value for a mature system. The plot for this figure, however, is not particularly linear, and results for the other cases were not statistically significant. The utility of the measure may be worth further investigation with a larger sample of cases.

Sharpening Edges:

At the observation level, none of the relationships are strictly linear for Edge Sharpening versus time (Figures 5.14 and 5.17. Cases III and IV are the most linear, and also have the strongest R^2 values; however, neither of them has the same sign (their slopes are opposite). Although some time trend appears to be discernible, it is clearly non–linear. Similar comments describe the findings for Mean Edge Sharpening for Cases III and IV. The most promising findings for Edge Sharpening metrics involve correlations with Pressure measures. At the observation level, the Pressure for Case II showed a strong inverse linear relationship with Mean Edge Sharpening, yielding a believable result: lower pressure values correspond with stronger edge returns; sharpening cloud edges were found by PWC staff to be indicative of development [27]. The result leads nicely to an encouraging finding over the Phase II to Phase III interval: the difference between maximum Edge Sharpening values over the interval predicts the minimum Pressure for the case. This is also an inverse relationship, interpreted to mean that when the strongest Phase III Edge Sharpening value greatly exceeds that of Phase II, the system is strong. If corroborated with a larger sample, this relationship would be very significant for early warning.

6.5 Other Significant Image–based Results

An interesting observation level relationship paired Mean Edge Strength and Comma Bulge Radius. Results were significant for all five cases, and varied directly for all but Case IV. If a positive relationship is found to be typical, it would suggest that Edge Strength increases across the image as the Comma Bulge expands. Another related result found that Mean Edge Strength also predicts Jet Tilt for all but Case II; however, two results were inversely and two directly related. It might seem more intuitive to expect Jet Tilt to increase as the system deepens, and generally becomes more meridional, as suggested above. This suggestion is supported by a direct correlation between Jet Tilt and Comma Bulge. Some significant correlations were also found with mean Edge Strength predicting Jet Tilt, but for the four significant cases, two had a direct and two an inverse relationship (Figures 6.8*D* and 6.9*A*, *B*, and *C*). A potentially useful finding was that Cyclonic Index predicted mean Edge

Strength for Cases I and III, with fairly linear regression plots (Figure 6.11*D* and Figure 6.12).

6.6 Conclusions

The statistical treatment of raw values in Chapter 5 yielded several interesting and potentially useful results. Some results help shape a general impression of explosive events, some are unexpected, while the most interesting have strong predictive potential that need only be confirmed and refined with a large sample.

The most unusual finding was that the lowest recorded pressure of a system followed the maximum translation velocity as measured from the low centre of the pressure charts. The implication of this finding is that fast storms appear to be weaker; a notion that contradicts climatological findings in the literature (eg. [38]). There was support for the finding from other related measures, as discussed above.

Some trends of general interest were revealed. The radius of the developing comma head was found to be directly related to the increasing tilt of the supporting Jet Stream. It was also related to the mean value of cloud edge strength. Edge mean strength also increased with the value of the Cyclonic Index, which was designed to locate cloud bodies representing mature cyclones. These correlations show the progress of the developing system, indicating some of the key features of development. It is possible that these findings could be used to help determine the degree of development of the system. It is also possible that they could assist in a procedure to automate tracking of the low pressure centre.

Some suggestive findings yielded potentially predictive relationships that are very promising. Pressure relationships returned three predictive equations over the Phase II to Phase III interval. In the first, the lowest pressure value (the overall system strength) was low for a low maximum image bearing value over the interval. One interpretation of the result is that more meridional systems are stronger, which is expected [38]; the reasoning here is that if the storm tended more westward in an eastward jet flow, then it must have had a large northerly component (been more meridional). The regression was quite strong. As explained previously, summary relations over the Phase II to Phase III interval have great potential for early warning.

The second relationship was between minimum pressure and The maximum Edge Strength difference across the interval. The relationship was inverse, so when the range of edge values was greater, lowest pressure was lower. The result is suggestive. The most obvious interpretation is that systems where clouds have sharpened greatly over the interval are strong. A related finding was a statistically significant relationship for Case II that found pressure falling as the mean edge strength across the image increased. The third relationship was between minimum pressure and the width of the baroclinic jet level cloud across the interval. As mentioned previously, it is questionable how well the method used actually captures this width; however, an inverse relationship was found between the variables, suggesting that as the width increases, pressure falls. Further study and refinement of the method would be necessary before the result should be held in much regard.

Five strong results were also found relating to the 24 hour Bergeron value across the Phase II to Phase III interval. A large Bergeron value indicates an intense storm (one where pressure falls rapidly). Bergeron value varied directly with the maximum storm bearing as calculated from imagery over the interval. The result is somewhat unexpected since it would seem to suggest that a more easterly system is more intense. An alternative interpretation is that a system which starts as easterly flow and then becomes more meridional across the interval is more intense. Further investigation would be needed to assess the full implication of the result.

A very strong direct correlation with the mean image–based velocity over the interval was found. Since the literature suggests rapid deepeners travel faster [14], the result is supported. A similar relationship at the case level found a dependence on the maximum chart–based velocity. Furthermore, the difference in image velocity over the interval directly predicts Bergeron value, suggesting that an accelerating system deepens faster. The last Bergeron relationship was a direct dependence on the range of the tilt of the supporting jet over the interval. This last result indicates that when the jet is rapidly becoming more meridional that strong systems result, supporting the result for difference in image velocity above.

6.6.1 Towards a Definition

In the course of this work, results and a review of the literature have tended to suggest that the traditional Bergeron value standard for defining a rapid deepener may not be appropriate. The first difficulty is that various authors have adopted different standards for calculating the Bergeron number to qualify a system as a Bomb. Some use different intervals to calculate the change in pressure, while others correct to different standard latitudes. In the process of characterising a rapid deepener it is critical to apply a consistent definition. The results of this work tend to suggest that the 24 hour interval is too long. On average, all five cases studied here had an interval from onset of deepening to beginning of decay of about 24 hours; since the Bergeron value should reflect the period of *maximum* deepening, it appears that a shorter interval is required. This concern must be balanced with the reality that most pressure data is only available at 6 hour intervals at best. For post–hoc analysis, the length of the interval from Phase II to Phase IV, as identified from imagery, was found to strongly correlate with both the minimum pressure of the system and the 24 hour Bergeron value. It is proposed here that the Phase II to Phase IV interval length could, or perhaps should, be part of the standard definition of a rapid deepener. Further climatological study with a large sample size would be required to validate this finding.

7. CONCLUSIONS

7.1 The Goals of Characterisation

As outlined in Chapter 1, the theoretical underpinning of this work was the development of the concept of Characterisation. As an extension and generalised case of the remote sensing principle of "signatures", Characterisation tries to bridge the gap between the physical properties of elements in an image scene and the meaning of the scene itself. In Chapter 1 it was suggested that while a collection of material signatures could describe a geometrically static scene in terms of its character, these signatures would not be comparable for a dynamically changing system in which both radiometric change was occurring as well as geometric change. Early aspects of this work tackled the problem of rectifying or describing geometric change so that the underlying radiometric changes could be investigated. Subsequently, it was found that the descriptions of geometric change themselves contained significant information, which was easier to capture and describe for the case of explosive deepening than the radiometric changes themselves.

Since the work has been exploratory, no attempt was made to uncover causal relationships. What was sought was a series of "signs" which would form a description (and ideally a definition) of character, much like how a series of "signatures" might describe the components of an image scene. Correlations between signs and specific events were sought, and in some cases good results were found.

7.2 Image Processing Results

As a result of funding cuts and the demands of running the PROBE system, MSC began to explore alternatives in 1992. Their original proposal, never completed, was the design and implementation of an expert system that could guide the processes used in PROBE to generate similar results. The work presented here was not designed to replicate PROBE, but several of the findings and methods may ultimately be usable for the same purposes.

Early aspects of this work focused on more traditional remote sensing methods, and sought to capture radiometric patterns of brightness and texture. Geometric deformations resulted in difficulties in analysis since the object of study did not superimpose in subsequent frames (the target was deformable and non–stationary). While some image processing methodologies are geometrically invariant (not sensitive to geometric changes), the work focused on rectification of geometric deformations and motions. Much of the radiometric work was abandoned as being a sizeable and difficult topic in its own right. Some methods that were explored are described in Section 4.3. A series of methods were developed and refined to perform this rectification, and it was found that the information they captured about the evolving system was rich.

The image processing methods developed can be categorised as: (1) pre–analysis enhancements; (2) "unrotation" rectification; (3) Principal Components rectification; (4) and manual rectification. The enhancements visually sharpened cloud edges, highlighted different cloud temperature regions, and improved image contrast. The Principal Components based enhancements (different from Principal Components rectification, discussed below), also combined information from the Water Vapour satellite channel, creating a synthetic image rich in information content. The remaining methods focused on strategies for capturing the motions and deformations between image frames.

The unrotation methods focused on applying a clockwise rotation to subsequent frames (since cyclones rotate counter–clockwise in the norther hemisphere) and calculating a "goodness of fit" for each angular change until a best match was found. These methods were at least semi–operator assisted, since the specification of the surface low and cleaning up of extraneous cloud masses was necessary for their operation. There is strong potential for these methods to be developed into fully automated operational procedures. They capture translation of the system by comparison of the low centres, as well as rotation of the cloud body (actually, of the Jet tilt rather than rotation of the comma head relative to the supporting Jet). They do not account for stretching or expansion, and so are only applicable to frames separated by a short interval, where these changes are minimised. Of these methods, the fairly simple "Pixels on Target" metric produced clean correlation curves with definite maxima. To assess the accuracy of the method, its results would need to be compared to an external reference.

Principal Components rectification shows promise for automation. The main operator requirement to implement it is to remove extraneous cloud from the image frame. There are some ways this process might be automated (briefly discussed in Section 7.3). Unlike unrotation methods, Principal Components captures scaling and translation information as well as rotation. Currently its main weakness is that it is sensitive to bulges in the segmented cloud representing the jet and comma head.

Manual rectification was initially performed using standard remote sensing image–to–image registration procedures. It was found that suitable "tie points" could not be consistently located in the evolving cloud field, which led to the first series of enhancements. These attempted to reveal natural edge transition features that could be tracked, but even then the process was not very accurate. Development of the Idealised Grid–based method then followed, where each progressive refinement to the grid model captured new information about the changing cloud field. While the grid method is fairly operator intensive, it is expected that acceptable results could be obtained fairly consistently and rapidly with minor operator training. A well thought out operational system should allow an operator to place the grid from each one–half hour GOES frame long before the next image arrives. The main weakness of the system is that currently it appears some backtracking and refinement of the placement of previous frames is necessary as the operator gets new details of the developing clouds; however, it appears unlikely that more than a series of a few subsequent frames of development (or a few hours) would be required to place the grid accurately, which would allow sufficient lead–time for early warning.

Based on a four–phase model of explosive development, statistical analysis of variables derived from the methods developed here revealed some significant correlations. The most promising were based on the length of the Phase II to Phase IV interval and on the change in variables over the Phase II to Phase III interval. The length of the Phase II to Phase IV interval correlated strongly with both the 24 hour Bergeron value and the minimum recoded case pressure ($R^2 = 0.79$, $p = 0.04$ and $R^2 = 0.87$, $p = 0.07$ respectively). The difference in measured values at the Phase III and Phase II intervals correlated with the 24 hour Bergeron value for image–based measures of storm bearing and jet tilt from imagery ($R^2 = 0.92$, $p = 0.01$ and $R^2 = 0.86$, $p = 0.07$ respectively); and the maximum image gradient strength of the storm edges predicted minimum case pressure ($R^2 = 0.77$, $p = 0.05$).

7.2.1 Characterisation

A synthesis of findings in the meteorological literature suggests that 24 hours is too long an interval upon which to base the definition of a rapid deepener, or to compute the Bergeron value describing its intensity. For the cases in this study, the length of the entire deepening interval was on average about 24 hours (Chapter 6). If this result is consistent with a large sample, it would suggest that 24 hours *must* span more than the interval of most rapid deepening. In Chapter 5, it was found that Bergeron value estimated over shorter time intervals (12 and 6 hours) consistently over–estimated the 24 hour Bergeron value. A standardised Bergeron value, perhaps corrected to a 12 hour interval at 45° north, could form one consistent descriptor of the character of these events. Standard terminology would also help in defining explosive deepeners. The author suggests the adoption of *strength* to describe lowest pressure and *intensity* to describe highest deepening rate. Findings in Chapter 6 strongly suggest that the length of the Phase II to Phase IV interval, as estimated from imagery, strongly represents both the strength and intensity of the system. Since this interval represents the length of the development period (from onset of rapid deepening to beginning of decay), it is intuitive to expect the length of the interval would have some bearing on the "explosiveness" of the system. The author suggests that further exploration of the Phase II to Phase IV length relationship may reveal that the length of this interval is a fundamental expression of the character of explosive events.

The original goals of this work where:

> Given human interpreters' abilities to subjectively infer the character of explosive cyclogenesis from satellite image loops, it is asserted that:
>
> **1** Computer–based systems should be able to use the same information as systems like PROBE to produce similar predictions
>
> **2** Rapid deepeners **do** possess identifiable characteristics that separate them from conventional cyclonic systems
>
> **3** Those characteristics can be isolated and used to come up with an alternative "definition" for rapid deepeners
>
> **4** The concept of *characterisation* from images is a valid and useful extension of the existing *spectral signature* concept in image processing

The author feels that the preliminary results based on the five cases presented show that most of the above goals can be met using image–based techniques. Item #2 in the list is still in question, and would require a larger sample of storms, including non–explosive events, to adequately test. Refinement and testing of these methods with a larger sample show great promise for revealing insight into the explosive process and for prediction of explosive events. Such future work is definitely worth pursuing.

7.2.2 Prediction

A rich series of metrics were developed based on the results of the grid method, and also some radiometric methods developed previously. In Chapter 2 it was asserted that the change from Phase I to Phase II (the onset of rapid deepening) would be a critical period to watch for useful prediction. While predictive methods on this interval undoubtedly be desirable, it proved too early for analysis using the grid method, which only spans the Phase II to Phase III (mature stage) interval; however, from this work it appears that Phase III, as assessed from the imagery (see Chapter 5), tends to occur from 20 to 30 hours prior to the time of maximum deepening. Hence, predictions based on information up the Phase III would be quite useful.

Several useful correlations between image and traditional metrics were found in Chapter 6. Of these, potentially the most useful were predictive relationships found over the Phase II to Phase III interval. Strong correlations to maximum pressure depth and to Bergeron value were found on a series of metrics. Of these, the prominent metrics were derived from variables measuring: storm bearing; cloud edge sharpness (edge strength); storm translation speed; and rate of Jet tilt. Of these, only the Jet tilt metrics are difficult to compute, while the edge sharpness measure could very likely be fully automated.

7.3 Future Considerations

Radiometric work that was discarded in favour of following the geometric rectification results would make a useful and independent study of the character of explosive winter storms. Preliminary work focused on banding in and around the comma head and adjacent Jet stream. These bands can be easily identified visually, and are shown schematically in Figure 2.2. Their development and placement may have fundamental implications for the system itself. Scaling remained a problem in this analysis, and a start was made with more invariant methods such as fractal dimension and use of the wavelet transform to summarise textures. The original goal was to classify areas of like–textural patterns and then analyse the nature of these texture clusters using landscape ecology style methods.

Cloud motion tracking has been developed by some authors (see Chapter 4). Further refinement of earlier work may help automate or semi–automate the cyclone analysis system developed here. Morphological operators hold promise for removing extraneous cloud masses, and use of the Cyclonic Index, or some refinement on that concept could very likely find a suitable centre point for estimating the surface low. The author imagines the most likely use of this work would be to develop an automated operational system which produces coarse–level statistics that serve to warn a human operator when a hazard condition threshold is crossed. At that point, the operator would use either a human–assisted method, such as those described here, or a system like PROBE to further refine prediction. Thus, the requirement for reducing subjectivity and operator–intensive work would be met, while still generating high quality predictions that would help prevent property damage and the loss of human life.

BIBLIOGRAPHY

[1] Arnaud, Y., Desbois, M., Maizi, J., 1992. "Automatic Tracking and Characterization of African Convective Systems on Meteosat Pictures" *Journal of Applied Meteorology*, **31**, 443–453

[2] Baum, B.A., Uttal, T., Poellot, M. et al., 1995. "Satellite Remote Sensing of Multiple Cloud Layers" *Journal of the Atmospheric Sciences*, **52(23)**, 4210–4230

[3] Bjerknes, V., 1921. "On the dynamics of the circular vortex with applications to the atmosphere and atmospheric vortex and wave motions" *Geofys. Publ.*, **2**, 1–88

[4] Bosart, L.F., 1981. "The Presidents' Day Snowstorm of 18–19 February 1979: A Subsynoptic-Scale Event" *Mon. Wea. Rev.*, **109**, 1542–1566

[5] Carlson, T.N., 1991. "Mid-latitude weather systems" Routledge, Chapman & Hall, Inc. New York, 507 pp.

[6] Carlson, T.B., 1980. "Airflow through Midlatitude Cyclones and the Comma Head Pattern" *Mon. Wea. Rev.*, **108**, 1498–1509

[7] Cherdak, A.S. (Ed.) 1993. "Imaging from Nondedicated Satellites" The MITRE Corporation, McLean, VA.

[8] Coakley, J.A., Baldwin, D.G., 1984. "Towards the Objective Analysis of Clouds from Satellite Imagery Data" *Journal of Climate and Applied Meteorology*, **23**, 1065–1099

[9] Defense Advanced Research Projects Agency, 1992. "Proceedings: 21st Image Understanding Workshop" Software and Intelligent Systems Technology Office, Arlington, Va., Morgan Kaufmann Publishers, San Mateo, Ca. 1062 pp.

[10] Ebert, E., 1989. "Analysis of Polar Clouds from Satellite Imagery Using Pattern Recognition and a Statistical Cloud Analysis Scheme" *Journal of Applied Meteorology*, **28**, 382–399

[11] Gonzales, R.C., Woods, R.E., 1993. "Digital Image Processing" Addison-Wesley, Don Mills, Ontario, 716 pp.

[12] Gyakum, J.R., 1983a. "On the Evolution of the QEII Storm. I: Synoptic Aspects" *Mon. Wea. Rev.*, **111**, 1137–1155

[13] Gyakum, J.R., 1983b. "On the evolution of the QEII storm. II: Dynamic and thermodynamic structure" *Mon. Wea. Rev.*, **111**, 1156–1173

[14] Gyakum, J.R., Anderson, J.R., Grumm, R.H., Gruner, E.L., 1989. "North Pacific Cold-Season Surface Cyclone Activity: 1975–1983" *Mon. Wea. Rev.*, **117**, 1141–1155

[15] Gyakum, J.R., Roebber, P.J., Bullock, T.A., 1992. "The role of Antecedent Surface Vorticity Development as a Conditioning Process in Explosive Cyclone Intensification" *Mon. Wea. Rev.*, **120**, 1465–1489

[16] Hobbs, P.V., 1978. "Organization and Structure of Clouds and Precipitation on the Mesoscale and Microscale in Cyclonic Storms" *Reviews of Geophysics and Space Physics*, **16(4)**, 741–755

[17] Jensen, J.R., 1986. "Introductory Digital Image Processing: A Remote Sensing Perspective" Prentice-Hall, Englewood Cliffs, New Jersey, 379 pp.

[18] Kegley, C.W., Kegley J.A., 1984. "Introduction to Logic" University Press of America, New York, 525 pp.

[19] Kohonen, T., 1995. "Self–Oranizing Maps" Springer–Verlag, New York, 362 pp.

[20] Kuo, Y-H., Low-Nam, S., 1990. "Prediction of Nine Explosive Cyclones over the Western Atlantic Ocean with a Regional Model" *Mon. Wea. Rev.*, **118**, 3–25

[21] Kuo, Y-H., Reed, R.J., Low-Nam, S., 1991. "Effects of Surface Energy Fluxes during the Early Development and Rapid Intensification Stages of Seven Explosive Cyclones in the Western Atlantic" *Mon. Wea. Rev.*, **119**, 457–476

[22] Levkowitz, H., Herman, G.T., 1993. "GLHS: A Generalized Lightness, Hue, and Saturation Color Model" *CVGIP: Graphical Models and Image Processing*, **55(4)**, 271–285

[23] Lillesand, T.M., Kiefer, R.W., 1987. "Remote Sensing and Image Interpretation: Second Edition" John Wiley & Sons, Toronto, 721 pp.

[24] Marroquin, J., Mitter, S., Poggio, T., 1987. "Probabilistic Solution of Ill–Posed Problems in Computational Vision" *Journal of the American Statistical Association*, **82(397)**, 76–89

[25] McGarigal, K., Marks, B.J., 1995. "Fragstats: Spatial Pattern Analysis Program for Quantifying Landscape Structure" United States Department of Agriculture, Forest Service, General Technical Report PNW-GTR-351, 122 pp.

[26] McLennan, N., Neil, L., 1988. "Marine Bombs Program (Phase II)" Environment Canada, Pacific Weather Centre, Vancouver, B.C.

[27] McLennan, N., Neil, L., Wells, G., 1988. "PROBE–An Operational 'Bomb' Forecasting System" Environment Canada, Pacific Weather Centre, Vancouver, B.C.

[28] Meteorological Service of Canada, 1992. "Artificial Intelligence – Explosive Marine Storms Prediction" Project Proposal, Environment Canada, Pacific Weather Centre, Vancouver, B.C.

[29] National Aeronautics and Space Administration, 1996. "GOES I-M DataBook" Goddard Space Flight Center, Reference #S-415-19, Revision 1 - 31 August 1996, pp. 197

[30] Neil, L., McLennan, N., 1988. "Explosive Cyclogenesis–A Case Study" Environment Canada, Pacific Weather Centre, Vancouver, B.C.

[31] Neiman, P.J., Shapiro, M.A., 1993. "The Life Cycle of an Extratropical Marine Cyclone. Part I: Frontal-Cyclone Evolution and Thermodynamic Air-Sea Interaction" *Mon. Wea. Rev.*, **121**, 2153–2176

[32] Oesterle, J.A., 1963. "Logic: The Art of Defining and Reasoning" Prentice-Hall, New York, 279 pp.

[33] Petterssen, S., Smebye, S.J., 1971. "On the development of extratropical cyclones" *Quart. J. R. Met. Soc.*, **97**, 457–482

[34] Reed, R.J., Albright, M.D., 1986. "A case study of explosive cyclogenesis in the Eastern Pacific" *Mon. Wea. Rev.*, **114**, 2297–2319

[35] Reed, R.J., Grell, G.A., Kuo, Y–H., 1993. "The ERICA IOP 5 Storm: Part I: Analysis and Simulation" *Mon. Wea. Rev.*, **121**, 1577–1593

[36] Roebber, P.J., 1984. "Statistical Analysis and Updated Climatology of Explosive Cyclones" *Mon. Wea. Rev.*, **112**, 1577–1589

[37] Rogers, E., Bosart, L.F., 1986. "An Investigation of Explosively Deepening Oceanic Cyclones" *Mon. Wea. Rev.*, **114**, 702–718

[38] Sanders, F., 1986. "Explosive Cyclogenesis in the West-Central North Atlantic Ocean, 1981–1984." *Mon. Wea. Rev.*, **114**, 1781–1794

[39] Sanders, F., Gyakum, J.R., 1980. "Synoptic-Dynamic Climatology of the 'Bomb"' *Mon. Wea. Rev.*, **108**, 1589–1606

[40] Schmetz, J., Holmlund, K., 1993. "Operational Cloud–Motion Winds from Meteosat Infrared Images" *Journal of Applied Meteorology*, **32**, 1206–1225

[41] Schneider, S.H. (Ed.), 1996. "Encyclopedia of Climate and Weather" Oxford University Press, New York, 929 pp.

[42] Shapiro, L.S., 1995. "Affine Analysis of Image Sequences." PhD. Dissertation, Cambridge University Press, New York, 210 pp.

[43] Shapiro, M.A., Keyser, D., 1990. "Fronts, Jet Streams, and the Tropopause" Extratropical Cyclones: The Erik Palmén Memorial Volume, C.W. Newton and E. Holpainen, Eds. American Meteorological Society, Boston., 167–191 pp.

[44] Uccellini, L.W., Keyser, D., Brill, K.F., Wash, C.H., 1985. "The Presidents' Day cyclone of 18–19 February 1979: Influence of upstream trough amplification and associated tropopause folding on rapid cyclogenesis" *Mon. Wea. Rev.*, **113**, 962–988

[45] Uccellini, L.W., Kocin, P.J., Petersen, R.A., Wash, C.H., Brill, K.F., 1984. "The Presidents' Day cyclone of 18–19 February 1979: Synoptic overview and analysis of the subtropical jet streak influencing the pre-cyclogenetic period" *Mon. Wea. Rev.*, **112**, 31–55

[46] Weldon, R., 1979. "Cloud Patterns and the Upper Air Wind Field" *Satellite Training Services Course Notes, Part IV*, Air Weather Service Document AWS/TR-79/003, 79 pp.

[47] Zachor, A., 1983. "Spectral Pattern Recognition in IR Remote Sensing" *Applied Optics*, **22(17)**, 2699–2703

[48] Zhang, J., Modestino, J.W., Langan, D.A., 1994. "Maximum–Likelihood Parameter Estimation for Unsupervised Stochastic Model–Based Image Segmentation" *IEEE Transactions on Image Processing*, **3(4)**, 404–419

A-1 PCI Scripts

```
:::::::::::::: APR8PCA.EAS :::::::::::::: local i

file="apr8 eign=1,2 midpoint= devrange=3 mask= rtype="long
report="apr8pca.txt

for i=1 to 8 dbic=i,i+16 dboc=i+24,i+32 r pca
endfor

report="term

:::::::::::::: CLOUD.EAS :::::::::::::: local #i

for #i=1 to 139 model on "janA
if(%{18+#i}<=20 and %{18+#i}>=-5)
%%{2+(#i-1)*3}=1; if(%{18+#i}<-5
and %{18+#i}>=-30) %%{3+(#i-1)*3}=1;
if(%{18+#i}<-30 and %{18+#i}>=-70)
%%{4+(#i-1)*3}=1;
endmodel endfor

:::::::::::::: COLLATE-B.EAS :::::::::::::: ! Combine IR
and WV frames in pairs

local i,out,DBin,DBout,channels,passes,base

system("clear")

ask "Input file: " fili ask "Output file: " filo

DBin=DBOpen(fili,"r") channels=DBChannels(DBin)
file=filo tex1="IR/WV sequence tex2=
dbsz=DBLines(DBin),DBPixels(DBin) pxsz= dblayout="band
if(mod(channels,2)=0) then
passes=channels/2 dbnc=channels
else print "Problems" stop
endif s cim r cim

for i=1 to passes base=(i-1)*2+1

dbic=i,i+passes dboc=base,base+1

s iii r iii
!ask "" tex1 endfor

:::::::::::::: COLLATE.EAS :::::::::::::: ! put
aprstorm.pix into IR,WV order

local i

for i=1 to 21 dbic=i,i+42 dboc=(i-1)*2+1,(i-1)*2+2 s iii
r iii
endfor

:::::::::::::: CORR.EAS :::::::::::::: local
#i,#means[2],#cov,#stdev[2] local #ch[2]

rem set dbic to the two input channels for correlation
#ch(1)=dbic(1) #ch(2)=dbic(2)

nsam=0 file="subanal trim= hisw= mask=2 report="/dev/null

for #i=1 to 2 dbic=#ch(#i) r his #means[#i]=imstat(2)
endfor

dbic=7 dbiw= exclude= arrayop= arrseg=3 dbsn="junk
dbsd="used by rcstats to get the image sum

model on "subanal %7=0;
%7=(%{#ch(1)}-#means[1])*(%{#ch(2)}-#means[2])*%%2
rem covariance under bitmap endmodel

r rcstats #cov=imstat(2)

for #i=1 to 2 model on "subanal
%7=0; %7=(%{#ch(#i)}-#means[#i])^2*%%2;
endmodel

r rcstats #stdev(#i)=imstat(2)
endfor

imstat(1)=#cov/(#stdev(1)*#stdev(2))^0.5 report="term

:::::::::::::: CORRECT.EAS :::::::::::::: ! Program to
assist in storm sequence registration

! Open display handler vd0:="Flow Normalisation
vd0# = 19,1024,1024,2,2,0,0,0,0,0,0,0,0,0,0,0 vd00="VD0:
r imageworks
```

```
:::::::::::::: CORRTEST.EAS :::::::::::::: local #i
local #means[5],#covs[4],#stdev[5],#rsdp[4] rem rsdp =
root of standard deviations product rem denominator of
correlation equation

nsam=0 file="subanal trim= hisw= mask=2 report="/dev/null

for #i=1 to 5

dbic=#i r his

#means[#i]=imstat(2) print #means[#i]
endfor

for #i=1 to 4 #covs[#i]=0 #rsdp[#i]=0
endfor

printf "\nCovariances:\n"

report="corr.txt dbic=7 dbiw= exclude= arrayop= arrseg=3
dbsn="junk dbsd="used by rcstats to get the image sum

for #i=1 to 4

model on "subanal %7=0; %7=(%{#i}-#means[#i]) *
(%{#i+1}-#means[#i+1])*%%2;
endmodel rem %%2 is a bitmap circle about the
image centre

r rcstats

rem imstat(2) from rcstats yields the image sum -- messy
but it works #covs[#i]=imstat(2) print #covs[#i]
endfor

printf "\nRSDP:\n"

for #i=1 to 5 rem stdevs for denominator
model on "subanal %7=0;
%7=(%{#i}-#means[#i])^2*%%2;
endmodel

r rcstats #stdev[#i]=imstat(2)
endfor

for #i=1 to 4 #rsdp[#i]=(#stdev[#i]*#stdev[#i+1])^0.5
print #rsdp[#i]
endfor

printf "\nCorrelation coefficients:\n"

for #i=1 to 4

print #covs[#i]/#rsdp[#i]

endfor

report="term

:::::::::::::: DUMP.EAS :::::::::::::: ! Dump E enhanced
(IR&WV) as jpg

! Variables local
DBin,channels,$in,loop,last,calc,dx,dy,$fbase

dx=512 dy=512

ask "Start IW (y) " tex1
if(f$extract(f$lowcase(tex1),1,1)="y" or tex1="") then
! Start the IW session vd0:="Flow Normalisation
vd0# = 19,dx,dy,2,2,0,0,0,0,0,0,0,0,0,0,0
vd00="VD0:

r imageworks ask "Hit <Enter> when the display
comes up" tex1
endif

ask "Image archive file: " fili $fbase=fili DBin =
DBOpen(fili,"r") channels = DBChannels(DBin)

! Set up bitmap colours r dcp "gc 1,196,196,196 r dcp
"gc 2,0,0,0

! Loop over image channels last=0 for loop=1 to channels/2
system("clear") printf "\nProcessing pair
%d\n",loop

! Load Image Frame ! hide work in progress r dcp "ib r dcp
```

```
"gb

file=$fbase calc=(loop-1)*2+1 dbic=calc,calc+1
vdoc=1,2 dbiw= vdow=

r ivi

! Load Bitmaps dbib=calc+1,calc+2 vdob=1,2 omod="off

! r ivb

! Show image r dcp "ls
! r dcp "gd 1,2 r dcp "io 2,1,2

! Wait(?)  if(last=0) then
ask "Ok? " text1 if(text1="y") last=1
endif

! Dump to capture filo="captmp caparea="all compres="false
s capture r capture

! Export image fili="captmp
filo=$fbase+f$string(loop)+".jpg" dbiw= dbic=1,2,3
dbib= dbvs= dblut= dbpct= ftype="jpg foptions=
s fexport r fexport

! Remove old captmp file system("rm captmp.pix")
endfor

! Cleanup call DBClose(DBin)

:::::::::::::: DUMPVECS.EAS :::::::::::::: ! Output vector
layers from file

local $infile,i,DBin,seg,flag,i

input "Dump from file: " $infile file=$infile
DBin=DBOpen($infile,"r")

seg=-1 flag=1 i=1 while flag=1
seg=DBNextSeg(DBin,"VEC",seg) if(seg<>-1) then
dbvs=seg filv=$infile+".vec"+f$string(i)
dmsform= fldnme="zcoord

s vwrite r vwrite
else flag=0
endif i=i+1

!ask ":" text1 endwhile

:::::::::::::: EDGE.EAS :::::::::::::: model on "test.pix
if(%4>100) then %%3=1
else %%3=0
endif endmodel

:::::::::::::: GCPBIT.EAS :::::::::::::: local i

dbsz=600,600 dbnc=1 upleft=-300,-300 loright=300,300
mapunits="metres dbsd= gcpform="xyxy fili="regions dboc=2
resample="cubic order=1 memsize= pciop="add pcival=1

for i=2 to 7 file="reg"+f$string(i) r pcimod

model on file if(%2=255) %3=%1;
endmodel endfor

:::::::::::::: GCPS.EAS :::::::::::::: local i

dbsz=600,600 dbnc=1 upleft=-300,-300 loright=300,300
mapunits="metres dbsd= gcpform="xyxy fili="regions dboc=1
resample="cubic order=1 memsize=

for i=1 to 8 file="reg"+f$string(i) r cim r geoset

file="regions dbsn="outline"+f$string(i)
rem tfile="vec"+f$string(i)+".gcp"
tfile="handy"+f$string(i)+".gcp" dbgc= r gcpread

filo="reg"+f$string(i) dbic=8+i
rem dbgc=i+1 dbgc=i+9 r reg
endfor

:::::::::::::: HANDY.EAS :::::::::::::: local i

file="handy dbiw= dbsd= polyinfo="points border="off
contint= dmsform="off fldnme= for i=1 to 8 rem    dbic=i
rem dbsn="outline"+f$string(i) rem    r rtv

dbvs=i+1 filv="handy"+f$string(i) r vwrite

endfor

:::::::::::::: IM.EAS :::::::::::::: local #i,#j

for #i=1 to 5 for #j=#i to 5
dbic=#i,#j r corr print "Correlation
(",#i,",",#j,"): ",imstat(1)
endfor endfor

:::::::::::::: IMP.EAS :::::::::::::: local i,$in,file

file=TEXTopen("list")

i=1 while($in!="<EOF>")
$in=TEXTread(file) fili=$in filo="test dbic=1
dboc=i dbiw= dbow= structur=1024,1024,1,0,216
datatype="8U flip="OFF swapfl="NO

s imagerd r imagerd

i=i+1 endwhile

call TEXTclose(file)

:::::::::::::: IMPORT.EAS :::::::::::::: local #i,mstring
names,$tp

dbic=1 dbiw= dbow= names = Text$Import("imp.txt") for #i=1
to f$len(names)
fili="import/"+names(#i)
filo="import/"+f$extract(names(#i),1,f$len(names(#i))-4)+".pix"
s fimport r fimport

fili=filo filo="janA dboc=#i s iii r iii
endfor

:::::::::::::: IMSTAT.EAS :::::::::::::: local i i=1 !for
i=1 to 63
dbic=i s his r his print imstat
!endfor

:::::::::::::: JPGINA.EAS :::::::::::::: ! Read a sequence
of jpeg IR and WV files for FUSION product

local $dir,i,mstring irfiles,mstring wvfiles,string
newfile,mstring stamp local pairs,base

system("clear") input "Read from directory: " $dir system
"ls -1 "+$dir+"/bc*" >irfiles system "ls -1 "+$dir+"/bd*"
>wvfiles

pairs=f$len(irfiles) if(pairs!=f$len(wvfiles)) then
print "Number of IR and WV files not equal!"  exit
endif print "Found "+f$string(pairs)+" pairs" ask "Hit
Enter: " text1

newfile=$dir file=$dir dbnc=2*pairs s cim r cim

for i=1 to pairs base=(i-1)*2
! IR file fili=irfiles(i) filo="tmpimp dbiw=

s fimport r fimport

fili="tmpimp filo=$dir dbic=1 dboc=base+1 dbiw=
dbow=

s iii r iii

file="tmpimp r dim

! WV file fili=wvfiles(i) filo="tmpimp dbiw=

s fimport r fimport

fili="tmpimp filo=$dir dbic=1 dboc=base+2 dbiw=
dbow=

s iii r iii

file="tmpimp r dim

file=newfile+".pix" ! IR image
source="%"+f$string(base+1)+"=255-%"+f$string(base+1)
s model r model
! WV image
source="%"+f$string(base+2)+"=255-%"+f$string(base+2)
s model r model

system "echo "+irfiles(i)+" | perl mkdate.pl"
```

```
>stamp cm01=stamp(1) dboc=base+1 s mcd r mcd

!ask "" tex1 endfor

system "ls "+$dir+"/bc* | perl mkdate.pl >"+$dir+".lst"

::::::::::::: JPGIN.EAS ::::::::::::: ! Read a sequence
of IR jpeg files

local $dir,i,mstring files,string newfile,mstring stamp

system("clear") input "Read from directory: " $dir system
"ls -1 "+$dir >files

newfile=$dir file=$dir dbnc=f$len(files) s cim r cim

for i=1 to f$len(files) fili=$dir+"/"+files(i) filo="tmpimp
dbiw= dblayout="pixel

s fimport r fimport

fili="tmpimp filo=newfile+".pix" dbic=1 dboc=i
dbiw= dbow=

s iii r iii

file=newfile+".pix"
source="%"+f$string(i)+"=255-%"+f$string(i)
s model r model

system "echo "+files(i)+" | perl mkdate.pl" >stamp
cm01=stamp(1) dboc=i s mcd r mcd

file="tmpimp r dim

!ask "" tex1 endfor

::::::::::::: LABELS.EAS ::::::::::::: local i,infile

file="metbase infile=textopen("charts") for i=1 to 31
cm01=textread(infile) dboc=i

r mcd !ask "hit return:" tex1 endfor

::::::::::::: NOLOW.EAS ::::::::::::: local i

for i=1 to 8 model on "apr8
if((0.584*(255-%{i})-90.1)<-5) %%{i+1}=1;
endmodel
endfor

::::::::::::: PCOR.EAS ::::::::::::: local i

for i=1 to 21 dbic=i,i+21,i+42 s pca r pca
endfor

::::::::::::: POLYS.EAS ::::::::::::: local #i,#j,$tp

file="temps

log start

for #i=1 to 6 for #j=1 to 3
dbic=18+#i dboc=#j+(#i-1)*3

if(#j=1) polv=-4,20 if(#j=2) polv=-29,-5
if(#j=3) polv=-70,-30

r ipg endfor
endfor

log stop

::::::::::::: PPRO2.EAS ::::::::::::: ! Modified for
IR only

! Sub-module for PREPRO.EAS to create single frame
enhancement ! in image prepro.pix - to be recycled !
Input Plane:  1 - IR

! Initial low-cloud bitmap (pass 1) print "Creating pass
1 low-cloud mask"

model on "prepro.pix !       if(0.584*(255-%1)-90.1>-5)
then
if(%1<109) then \ ! from temperature LUT =
>-5C %%2=1
else %%2=0
endif endmodel
```

```
! Pass 1 low cloud from eign2 print "Adjusting low-cloud
regions..."

model on "prepro.pix if(%%2=0) then
%3=%2 else
%3=0 endif
endmodel

! Mode filter low cloud (pass 2) file="prepro.pix dbic=3
dboc=3 flsz=7,7 thinline="off keepvalu= print "   Module
FMO" r fmo

! Pass 2 low-cloud to bitmap print "   Storing results"

model on "prepro.pix if(%3>0) then
%%2=0 else
%%2=1 endif
endmodel

! Sobel edges print "Edge enhancing boundaries..."

factor=1 print "   Module FSOBEL" r fsobel

print "   Storing results" model on "prepro.pix
if(%3<=200) then %%3=0
else %%3=1
endif endmodel

print "DONE!"

::::::::::::: PPRO-B.EAS ::::::::::::: ! Sub-module
for PREPRO.EAS to create single frame enhancement ! in
image prepro.pix - to be recycled !  Input Planes:  1 -
IR, 2 - WV

! Initial low-cloud bitmap (pass 1) print "Creating pass
1 low-cloud mask"

model on "prepro.pix !       if(0.584*(255-%1)-90.1>-5)
then
if(%1<109) then \ ! from temperature LUT =
>-5C %%2=1
else %%2=0
endif endmodel

! Histogram matched LUT print "Calculating variance
adjusted principal components..."

file="prepro func="matc dbic=2 dblut=4 dbsn="match
dbsd="histogram matching ostr= sdpt= trim= ! mask should
be set by calling routine mask= \ ! but...  dbhc=1 print
"   Module FUN" r fun

! Make changes to channel dboc=2 print "   Module LUT"
r lut

! Decorrelation stretch dbic=1,2 eigensup=1 ! global
var set from PREPRO report="prepro.eign print "   Module
DECORR" r decorr report="TERM

! Pass 1 low cloud from eign2 print "Adjusting low-cloud
regions..."

model on "prepro.pix if(%%2=0) then
%3=%2 else
%3=0 endif
endmodel

! Mode filter low cloud (pass 2) dbic=3 dboc=3 flsz=7,7
thinline="off keepvalu= print "   Module FMO" r fmo

! Pass 2 low-cloud to bitmap print "   Storing results"

model on "prepro.pix if(%3>0) then
%%2=0 else
%%2=1 endif
endmodel

! Sobel edges print "Edge enhancing boundaries..."

factor=1 print "   Module FSOBEL" r fsobel

print "Storing results" model on "prepro.pix
if(%3<=100) then %%3=0
else %%3=1
endif endmodel

! Code for IR temp and PC1 fusion print "Creating fusion
```

```
images"

! create pseudo-colour composite from IR encoding="over
dbic=1 dbpct=5 dboc=4,5,6 dbiw=

print "   Module PCE" r pce

dbic=4,5,6 dboc=dbic ihsmodel="cylinder

print "   Module IHS" r ihs

dbic=2 dblut=6 dboc=4  \ ! overwrite I with stretched PC1

print "   Module LUT" r lut

dbic=4,5,6 dboc=dbic

print "   Module RGB" r rgb

print "DONE!"

:::::::::::::: PPRO.EAS :::::::::::::: ! Sub-module for
PREPRO.EAS to create single frame enhancement ! in image
prepro.pix - to be recycled !  Input Planes:  1 - IR,
2 - WV

! Initial low-cloud bitmap (pass 1) print "Creating pass
1 low-cloud mask"

model on "prepro.pix if(0.584*(255-%1)-90.1>-5) then
%%2=1 else
%%2=0 endif
endmodel

! Histogram matched LUT print "Calculating variance
adjusted principal components..."

file="prepro func="matc dbic=2 dblut=4 dbsn="match
dbsd="histogram matching ostr= sdpt= trim= ! mask should be
set by calling routine dbhc=1 print " Module FUN" r fun

! Make changes to channel dboc=2 print "   Module LUT"
r lut

! Decorrelation stretch dbic=1,2 eigensup=1 ! global
var set from PREPRO report="prepro.eign print "   Module
DECORR" r decorr report="TERM

! Pass 1 low cloud from eign2 print "Adjusting low-cloud
regions..."

model on "prepro.pix if(%%2=0) then
%3=%2 else
%3=0 endif
endmodel

! Mode filter low cloud (pass 2) dbic=3 dboc=3 flsz=7,7
thinline="off keepvalu= print "   Module FMO" r fmo

! Pass 2 low-cloud to bitmap print "   Storing results"

model on "prepro.pix if(%3>0) then
%%2=0 else
%%2=1 endif
endmodel

! Sobel edges print "Edge enhancing boundaries..."

factor=1 print "   Module FSOBEL" r fsobel

print "   Storing results" model on "prepro.pix
if(%3<=100) then %%3=0
else %%3=1
endif endmodel

print "DONE!"

:::::::::::::: PREPRO2.EAS :::::::::::::: ! Modified for
sequences with only the IR channel

! Perform flow normalisation pre-processing steps
! Source: image database of temporal sequence !
of TIR only !  !        Dest:    source image with !
low-cloud and edges bitmaps

! Local variables local DBin, channels, passes, loop local
string filename

system("clear") ask "Image sequence filename: " fili

filename=fili

! Open source and get needed info DBin =
DBOpen(filename,"r") channels = DBChannels(DBin)

printf "Found %d channels in %s\n",channels,filename

passes=channels ask "Proceed? " tex1
if(f$extract(tex1,1,1)<>"y") then
call DBClose(DBin) return
endif

! -=-=-=-

! set up and process each channel pair for loop=1 to
channels
system("clear") printf "\nProcessing channel
#%d\n\n",loop

! Swap input channels to prepro.pix for enhancement
fili=filename filo="prepro dbic=loop dboc=1 dbiw=
dbow=

r iii

! Run PPRO preprocessing enhancement r ppro2

print "Transferring results to original image !
Swap output to original image
fili="prepro filo=filename dbiw= dbow=

! Transfer enhancement bitmaps dbib=2,3 dbob=

r iib endfor

! Clean up call DBClose(DBin)

:::::::::::::: PREPRO-B.EAS :::::::::::::: ! Perform flow
normalisation pre-processing steps !    Source: image
database of temporal sequence !        of TIR
and WV scenes !  !  Dest:  TIR and PC2 image, !
low-cloud and edges bitmaps

! Local variables local DBin, DBout, channels, passes,
loop, base local string filename, outfile, string stamp

system("clear") ask "Image sequence filename: " fili
filename=fili

! Open source and get needed info DBin =
DBOpen(filename,"r") channels = DBChannels(DBin)

printf "Found %d channels in %s\n",channels,filename

! Assuming they are in order TIR, WV...
if(mod(channels,2)<>0) then
print "The input file should contain TIR and WV
pairs only for each scene!"  return
endif

passes=channels/2 printf "Assuming %d channels for each
of TIR and WV\n",passes ask "Enhanced image filename:
" fili outfile=fili

file=fili tex1="RGBs from PCT enhanced PC1" tex2=
dbsz=DBLines(DBin),DBPixels(DBin) pxsz=1,1 dbnc=passes*3
dblayout="tiled jpeg" \ ! compress that image!

s cim print "Creating output file" r cim

! -=-=-=-

! set up and process each channel pair for loop=1 to passes
system("clear") base=(loop-1)*2+1 printf
"\nProcessing channels #%d and #%d\n\n",base,base+1

! Swap input channels to prepro.pix for enhancement
fili=filename filo="prepro dbic=base,base+1
dboc=1,2 dbiw= dbow=

r iii

! Run PPRO preprocessing enhancement r ppro

print "Transferring results to enhanced image"
base=(loop-1)*3+1 fili="prepro filo=outfile
dbic=4,5,6 dboc=base,base+1,base+2 dbiw= dbow=

r iii
```

```
! Label image

! Transfer enhancement bitmaps dbib=2,3 dbob=

r iib endfor

! Clean up call DBClose(DBin)

::::::::::::: PREPRO.EAS ::::::::::::: ! Perform flow
normalisation pre-processing steps !    Source: image
database of temporal sequence !       of TIR
and WV scenes !  !  Dest:  TIR and PC2 image, !
low-cloud and edges bitmaps

! Local variables local DBin, DBout, channels, passes,
loop local string filename

system("clear") ask "Image sequence filename: " fili
filename=fili

! Open source and get needed info DBin =
DBOpen(filename,"r") channels = DBChannels(DBin)

printf "Found %d channels in %s\n",channels,filename

! Assuming they are in order TIR, WV...
if(mod(channels,2)<>0) then
print "The input file should contain TIR and WV
pairs only for each scene!"  return
endif

passes=channels/2 printf "Assuming %d channels for
each of TIR and WV\n",passes ask "Proceed? " text1
if(f$extract(text1,1,1)<>"y") then
call DBClose(DBin) return
endif

! -=-=-=-

! set up and process each channel pair for loop=1 to
channels by 2
system("clear") printf "\nProcessing channels #%d
and #%d\n\n",loop,loop+1

! Swap input channels to prepro.pix for enhancement
fili=filename filo="prepro dbic=loop,loop+1
dboc=1,2 dbiw= dbow=

r iii

! Run PPRO preprocessing enhancement r ppro

print "Transferring results to original image !
Swap output to original image
fili="prepro filo=filename dbic=2 dboc=loop+1
dbiw= dbow=

r iii

! Transfer enhancement bitmaps dbib=2,3 dbob=

r iib endfor

! Clean up call DBClose(DBin)

::::::::::::: PROJECT.EAS ::::::::::::: ! input 512x512
vecs (otherwise run vec_5.pl first) ! output vwrite vectors
to directory in long/lat

local $infile,DB,nSEG,i,$outdir local mstring instr,vecsegs

input "Data file: " $infile DB=DBOpen($infile,"r") input
"Output directory: " $outdir

system("rm vecsegs.txt") file=$infile ltyp="short aslt=116
assn= report="vecsegs.txt" r asl aslt= report="TERM

system "cat vecsegs.txt | perl vecsegsread.pl" >vecsegs
nSEG=f$len(vecsegs)-2

system("cp metbak.pix metbase.pix") ! copy
vecs to metbase fili=$infile filo="metbase
dbsl=f$value(vecsegs(nSEG+1)),-1*f$value(vecsegs(nSEG+2))
dbos= s iia r iia

for i=1 to nSEG fili="metbase ingeo=1 dbvs=2+i filo=
outgeo= ounits= dbsn=vecsegs(i) dbsd=

s vecpro r vecpro
endfor

for i=1 to nSEG dbvs=2+i+nSEG filo=fili outgeo= ounits="lon
dbsn=vecsegs(i)

s vecpro r vecpro
endfor

file="metbase system "mkdir "+$outdir

for i=1 to nSEG dbvs=(nSEG*2+2)+i
filv=$outdir+"/"+vecsegs(i) dmsform= fldnme="zcoord

s vwrite r vwrite
endfor

::::::::::::: RCORR.EAS ::::::::::::: rem correlation
as plane two is rotated wrt plane one rem dbic(1)=static
channel rem dbic(2)=rotated channel

local #angle,#ch1,#ch2,string result #ch1=dbic(1)
#ch2=dbic(2)

rem system("echo `date` >>rcorr.txt") print
boxclear(1,1,80,24),@(1,1) for #angle=1 to 30
fili="subanal filo=fili dbic=#ch2 dboc=6 dbgc=
dbiw= angle=#angle rcentre= resample="cubic
monitor="off

r rot

dbic=#ch1,6 r corr

dbic=#ch1,6 vdoc=1,2 dbiw= vdow= r ivi

%%4=0 if(%1<>0 and %6<>0) %%4=1 dbib=4 vdob=3
omod="off dbiw= vdow= r ivb

result="Angle "+f$string(#angle)+":
"+f$string(imstat(1)) print result
rem system("echo "+result+" >>rcorr.txt") endfor

monitor="on

::::::::::::: RECT2.EAS ::::::::::::: ! Call C code
to handle vector layers etc.  ! Otherwise, start up IW
from Easi

! Variables local
DBin,channels,$in,$prompt,loop,last,calc,dx,dy

dx=512 dy=512

ask "Start IW (y) " text1
if(f$extract(f$lowcase(text1),1,1)="y" or text1="") then
! Start the IW session vd0:="Flow Normalisation
vd0# = 19,dx,dy,2,2,0,0,0,0,0,0,0,0,0,0,0,0
vd00="VDO:

r imageworks ask "Hit <Enter> when the display
comes up" text1
endif

! Loop over image channels ask "Image archive file:
" fili DBin = DBOpen(fili,"r") channels = DBChannels(DBin)

! Set up bitmap colours r dcp "gc 1,196,196,196 r dcp
"gc 2,0,0,0

last=0 for loop=1 to channels/2
system("clear") print

if(last=0) loop=1

$prompt="loop("+f$string(loop)+") :" print $prompt
input "" $in if(f$len($in)<>0) then
if(f$value($in)=0) stop loop=f$value($in)
endif

printf "\nProcessing pair %d\n",loop

if(loop=last) then goto done
endif last=loop

!goto done

! Load Image Frame ! hide work in progress r dcp "ib r dcp
"gb
```

```
file=fili calc=(loop-1)*2+1 dbic=calc,calc+1
vdoc=1,2 dbiw= vdow=

r ivi

! Load Bitmaps dbib=calc+1,calc+2 vdob=1,2 omod="over

r ivb

done: rem continue on

! Unhide display r dcp "ls" r dcp "gd 1,2 r dcp
"io 2,1,2"

endfor

! Cleanup call DBClose(DBin)

:::::::::::::: RECT3.EAS :::::::::::::: ! This version
assumes no WV channel is present.  Therefore no !
PCA product displayed

! Variables local
DBin,channels,$in,$prompt,$ifile,loop,last,calc,dx,dy

dx=512 dy=512

system("clear") ! Loop over image channels ask "Image
archive file: " file $ifile=file DBin = DBOpen(file,"r")
channels = DBChannels(DBin)

ask "Start IW (y) " tex1
if(f$extract(f$lowcase(tex1),1,1)="y" or tex1="") then
! Start the IW session vd0:="Flow Normalisation
vd0# = 19,dx,dy,3,3,0,0,0,0,0,0,0,0,0,0,0
vd00="VD0:

r imageworks ask "Hit <Enter> when the display
comes up: " tex1 print "Initialising..."

! Set up bitmap colours r dcp "gc 1,196,196,196
r dcp "gc 2,0,0,0 r dcp "gd 1,2

! Read Pseudo-colour table file="prepro.pix"
dbpct=5 \! pct to display r ivp r dcp "pd 1,0
file=$ifile
endif

last=0 for loop=1 to channels !       system("clear")
print

if(last=0) loop=1

$prompt="loop("+f$string(loop)+"): " printf
"%s",$prompt input "" $in if(f$len($in)<>0) then
if(f$value($in)=0) stop loop=f$value($in)
endif

printf "Processing image %d -
%s\n",loop,DBReadChanDesc(DBin,loop)

if(loop=last) then goto done
endif last=loop

!goto done

! Load Bitmaps calc=(loop-1)*2+1 dbib=calc+1,calc+2
vdob=1,2 omod="over

r ivb

! Load Image Frame dbic=loop vdoc=1 dbiw= vdow=

r ivi

done: rem continue on endfor

! Cleanup call DBClose(DBin)

:::::::::::::: RECTIFY.EAS :::::::::::::: ! Call C code
to handle vector layers etc.  ! Otherwise, start up IW
from Easi

! Variables local DBin,channels,loop,last,dx,dy

dx=640 dy=640

! Start the IW session
```

```
vd0:="Flow Normalisation vd0# =
19,dx,dy,2,2,0,0,0,0,0,0,0,0,0,0,0 vd00="VD0: r imageworks

! Loop over image channels !ask "Image archive file: "
fili fili="apr94.pix DBin = DBOpen(fili,"r") channels =
DBChannels(DBin)

last=0 for loop=1 to channels by 2
system("clear") print

ask "loop: " vdic if(vdic(1)!=0) then
loop=vdic endif printf "\nProcessing pair
%d\n\n",((loop-1)/2)+1

if(loop=last) then goto done
endif last=loop

! Load Image Frame file=fili dbic=loop,loop+1 vdoc=1,2
dbiw= vdow=

r ivi r dcp "id 2,1,2" r dcp "ls"

! Load Bitmaps dbib=loop+1,loop+2 vdob=1,2 omod="over

r ivb done: rem continue on endfor

! Cleanup call DBClose(DBin)

:::::::::::::: REDUCE.EAS :::::::::::::: rem  Remove all
values in allvec that are not under the flow bitmap

local i

for i=1 to 8 model on "allvec
if(%{i+8}=0) %{i}=0; endmodel
endfor

:::::::::::::: REGBIT.EAS :::::::::::::: model on "subanal
if(%%5=1) then %6=1;
endif; endmodel


:::::::::::::: REGSEQ.EAS ::::::::::::::
%8=.584*(255-%1)-90.1

if(%8>=-5) then %%1=1
else %%1=0
endif

if(%8<-30) then %%2=1
else %%2=0
endif

if(%%1=1) then %3=0
else %3=%2
endif


:::::::::::::: RTARG.EAS :::::::::::::: rem on target
pixels as bitmap two is rotated wrt bitmap one rem
dbic(1)=static channel rem dbic(2)=rotated channel

local #angle,#ch1,#ch2,#count,string result #ch1=dbic(1)
#ch2=dbic(2)

print boxclear(1,1,80,24),@(1,1) system("echo 'date'
>rtarg.txt") system("echo Rotation between planes
"+f$string(#ch1)+" and " \
+f$string(#ch2)+" >>rtarg.txt")

rem count pixels in static channel %6=%%{#ch1+4} dbic=6
dbiw= exclude= arrayop= arrseg=3 dbsn="junk dbsd="used by
rcstats to get the image sum r rcstats #count=imstat(2)

for #angle=-20 to 60 by 5 fili="subanal filo=fili
dbic=6 dboc=7 dbgc= dbiw= angle=#angle rcentre=
resample="near monitor="off

rem copy bitmap into image plane for rotation %6=0
if(%%{#ch2+4}=1) %6=1 r rot

rem define on target pixel as #{bit2*bit1=1}/#bit1
%6=%%{#ch1+4}*%7

dbic=6 vdoc=1 dbiw= vdow= rem   r ivi exclude= arrayop=
arrseg=3 dbsn="junk dbsd="used by rcstats to get the
image sum
```

```
rem count pixels overlap r rcstats
imstat(1)=imstat(2)/#count

result="Angle "+f$string(#angle)+":
"+f$string(imstat(1)) print result system("echo
"+result+" >>rtarg.txt")
endfor system("echo `date` >>rtarg.txt")

monitor="on


:::::::::::::: SEGMENT.EAS :::::::::::::: local #i

file="aprstorm flsz=7,7 mask= thinline="OFF keepvalu=
factor=1 for #i=1 to 16
print "Processing image "+f$string(#i) print
rem print " low cloud removal" print " high and low
cloud removal"
model on "aprstorm rem temperature conversion
rem %{#i+32}=0.584*%{#i}-90.1;
%{#i+32}=0.584*(255-%{#i})-90.1;
rem remove low cloud rem
if(%{#i+32}>=-5) %{#i+32}=0; if(%{#i+32}<-30 or
%{#i+32}>-5) %{#i+32}=0;
endmodel

print " cumulus filtering" rem mode filtering to
remove cumulus puffs dbic=#i+32 dboc=dbic r fmo

print " gradient computation" rem compute gradient
images dboc=#i+16 r fsobel print
endfor


:::::::::::::: SHOWREG.EAS :::::::::::::: local i

for i=1 to 8 file="reg"+f$string(i) dbic=1 vdoc=i r ivi
endfor

:::::::::::::: SOBEDGE.EAS :::::::::::::: ! Calculate
sobel gradients for edge sharpening tracking

! From PPRO2 ! Input Plane:  1 - IR

local $out,i

! Initial low-cloud bitmap (pass 1) print "Creating pass
1 low-cloud mask"

model on "prepro.pix if(%1<109) then \ ! from temperature
LUT = >-5C
%%2=1 else
%%2=0 endif
endmodel

! Pass 1 low cloud from eign2 print "Adjusting low-cloud
regions..."

model on "prepro.pix if(%%2=0) then
%3=%1 else
%3=0 endif
endmodel

! Mode filter low cloud (pass 2) file="prepro.pix dbic=3
dboc=3 flsz=7,7 thinline="off keepvalu= print "   Module
FMO" r fmo

! Pass 2 low-cloud to bitmap print "   Storing results"

model on "prepro.pix if(%3>0) then
%%2=0 else
%%2=1 endif
endmodel

! Sobel edges print "Edge enhancing boundaries..."

factor=1 dboc=7 print "   Module FSOBEL" r fsobel

dbic=7 exclude=0 arrseg=8 r rcstats

! Collect elements of imstat(6)

for i=1 to 6 $out=$out+f$string(imstat(i))+" "
endfor print $out system("echo "+$out+" >>results")

print "DONE!"

:::::::::::::: SOBEL.EAS :::::::::::::: rem x and y
gradient filters local #x[9],#y[9],$fn,#in,#out

#x[1]=-1 #x[2]=-2 #x[3]=-1 #x[4]=0 #x[5]=0 #x[6]=0 #x[7]=1
#x[8]=2 #x[9]=1

#y[1]=-1 #y[2]=0 #y[3]=1 #y[4]=-2 #y[5]=0 #y[6]=2 #y[7]=-1
#y[8]=0 #y[9]=1

rem:  Usage filename start_input start_output
system("clear") print @(BOLD),@(REVERSE),"Sobel gradient
vector computation",@(ALLOFF) print input "Database
filename: " $fn input "Start input channel: " #in input
"Start output channel: " #out

model on $fn

 rem X gradient %{#out}=(%{#in}[-1,-1]*#x[1]+%{#in}[
  0,-1]*#x[2]+%{#in}[ 1,-1]*#x[3]+\
   %{#in}[-1, 0]*#x[4]+%{#in}[ 0, 0]*#x[5]+%{#in}[
   1, 0]*#x[6]+\ %{#in}[-1, 1]*#x[7]+%{#in}[ 0,
   1]*#x[8]+%{#in}[ 1, 1]*#x[9] );

 rem Y gradient %{#out+1}=(%{#in+1}[-1,-1]*#x[1]+%{#in+1}[
  0,-1]*#x[2]+%{#in+1}[ 1,-1]*#x[3]+\
   %{#in+1}[-1, 0]*#x[4]+%{#in+1}[ 0,
   0]*#x[5]+%{#in+1}[ 1, 0]*#x[6]+\ %{#out+1}[-1,
   1]*#x[7]+%{#out+1}[ 0, 1]*#x[8]+%{#out+1}[ 1,
   1]*#x[9] );

 rem Magnitude of gradient vector %{#out+2}=abs(%{#out})
  + abs(%{#out+1});

 rem Angle of gradient vector %{#out+3}=atan(%{#out+1}
  / %{#out});

endmodel


:::::::::::::: SOBS.EAS :::::::::::::: ! Sobel edge
loop prog

! Source: image database of temporal sequence !
of TIR only

! Local variables local DBin, channels, passes, loop local
string filename

system("clear") ask "Image sequence filename: " fili
filename=fili

! Open source and get needed info DBin =
DBOpen(filename,"r") channels = DBChannels(DBin)

printf "Found %d channels in %s\n",channels,filename

passes=channels ask "Proceed? " tex1
if(f$extract(tex1,1,1)<>"y") then
call DBClose(DBin) return
endif system("echo "+filename+" >results") system("date
>>results ; echo >>results")

! -=-=-=-

! set up and process each channel for loop=1 to channels
system("clear") printf "\nProcessing channel
#%d\n\n",loop

! Swap input channels to prepro.pix for enhancement
fili=filename filo="prepro dbic=loop dboc=1 dbiw=
dbow=

r iii

! Run SOBEDGE preprocessing enhancement r sobedge
endfor

! Clean up call DBClose(DBin)

:::::::::::::: SUBREG.EAS :::::::::::::: local i

dbsz=600,600 dbnc=1 upleft=-300,-300 loright=300,300
mapunits="metres dbsd= gcpform="xyxy fili="regions dboc=2
resample="cubic order=1 memsize= pciop="add pcival=1

for i=2 to 7 filo="reg"+f$string(i) file=filo r pcimod

dbic=i dbgc=i r reg
endfor

:::::::::::::: TEMP2.EAS :::::::::::::: model on "test.pix
```

```
if(%%2=1) %2=0 endmodel

:::::::::::::: TEMP3.EAS :::::::::::::: model on "test.pix
if(%3=0) then %%2=1
else %%2=0
endif endmodel

:::::::::::::: TEMP.EAS :::::::::::::: model on "test.pix
if(0.584*(255-%1)-90.1>-5) then %%2=1
else %%2=0
endif endmodel

:::::::::::::: TEMPS.EAS :::::::::::::: local #i,#src,#dest

#src=1 #dest=9

for #i=1 to 8 model on "temps2
%{#i-1+#dest}=0.584*(255-%{#i-1+#src})-90.1
rem %{#i-1+#dest}=(%{#i-1+#src}+90.1)/0.584
endmodel endfor

:::::::::::::: TEMPSEG.EAS :::::::::::::: local i

model on "midseg if(%%3=1) then
if((0.584*(255-%8)-90.1)<-30) then %%9=1;
else %%10=1;
endif else
%%9=0; %%10=0;
endif endmodel

:::::::::::::: TEST.EAS :::::::::::::: local i

i=VDOpen("VD00") call VDClose(i)

:::::::::::::: TOJPG.EAS :::::::::::::: ! Export nov98
storm as jpeg sequence for local display

local i,channels,infile

fili="Enov98 infile=DBOpen(fili,"r")
channels=DBChannels(infile)

for i=1 to channels filo="nov98/"+f$string(i)+".jpg" dbic=i

r fexport endfor

:::::::::::::: TO_LL.EAS :::::::::::::: ! reproject
pixel->PS within metbase.pix ! then PS->lon/lat in
metbasex2.pix

local i

for i=1 to 11 fili="metbase ingeo=1 dbvs=2+i filo= outgeo=
ounits= dbsn=f$string(2411+i) dbsd=

s vecpro r vecpro

dbvs=2+i+11 filo="metbasex2 outgeo=1

s vecpro r vecpro
endfor

:::::::::::::: VEC_5.EAS :::::::::::::: ! Export vecs
with vwrite, run v.5.pl to reduce by 1/2 and re-import
with vread

local i

for i=2 to 12 file="apr96gcp dbvs=i filv="out.vec
dmsform="off fldnme="zcoord

system("rm out.vec") s vwrite r vwrite
!ask "" tex1

system("cat out.vec | perl v.5.pl >in.vec") !
system("head in.vec ; tail in.vec | less") !endfor

filv="in.vec vecunit="pixel file="a96vec
dbsn="24"+f$string(10+i) s vread r vread
!ask "" tex1 endfor

:::::::::::::: VECDISP.EAS :::::::::::::: ! Transfer
vectors to bitmap display

local $infile,DB,seg,flag,vid

input "Data file: "$infile DB=DBOpen($infile,"r")
file=$infile

seg=-1 flag=1 r dcp "gc 1,8 r dcp "gc 2,7 r dcp "gc 3,4

vid=0 while flag>0
seg=DBNextSeg(DB,"VEC",seg) if(seg<>-1) then
dbvs=seg vdob=2 dbvw=0,0,512,512 pclg=
vid=vid+1 vdob=vid polyinfo="both r dcp "gc
"+f$string(vid)+",7" r ivv !    polyinfo="points

s ivv r ivv
!ask "next:" TEX1 else
flag=0 endif
endwhile

!vdob=3 !polyinfo="both !r dcp "gc "+f$string(vid)+",7"
!r ivv

:::::::::::::: VECS.EAS :::::::::::::: local i,a

file="regions dmsform="off fldnme= for i=1 to 8
dbvs=i+1 filv="region"+f$string(i)

s vwrite r vwrite
endfor
```

A-2 Perl Scripts

```
:::::::::::::: addt0.pl :::::::::::::: #!/usr/bin/perl #
Calculate hours relative to t0 and paste into new sheet

chdir "/usr/big/data/work/thesis/finish/indices";

$t0=$ARGV[0]; $base=$ARGV[1]; #$t0=13*24+8;

@stats=`cat $base.sta`; @files=`cat $base.run`;
#@stats=`cat jA.sta`; #@files=`cat jA.run`;
while($i=shift(@files)){
  $i=~/echo (bc.{6})/; $time=`echo $1 | p/times`;
  $time=~/^.+\t.{8}(..) (..):(..)/; $hrs=$1*24+$2+$3/60;
  $hrst0=$hrs-$t0;

  print("$hrst0\t".shift(@stats)); }

:::::::::::::: batch.pl :::::::::::::: # Batch processing
for entire case opendir(DIR,"IMSRC");

$i=1; while($line=readdir(DIR)){
  # ensure it is an image file
  if($line=~/(^bc\w+).(jpg)/i){
    # convert to raw format $fnam=$1; system("convert
    IMSRC/$1.$2 gray:$1.bin"); $stamp=`echo $line |
    ./times`;

    print "\n\nProcessing image $i:\n"; print "\t$stamp\n";
    # Interface to C prog mask.c for computations
    open(RET,"cat $fnam.bin | ./mask 2>&1 1>tp.bin |");

    while($line=<RET>){ print "$line";

      if($line=~/.+read, (\d+) highcloud/){ # grab
highcloud count $hc=$1;
      } if($line=~/.+value: (\d+.\d+)/){
# grab cyclonic index $ci=$1;
      } }

    # store results system("cp tp.bin ANALYSIS/$fnam.$i");
    system("rm $fnam.bin"); system("cp template.hdr
    ANALYSIS/$fnam.$i.hdr"); system("echo $i\t$hc\t$ci
    >>ANALYSIS/results"); $i++;
  } }

:::::::::::::: cp.pl :::::::::::::: while($in=<>){
$in=~/^(.{8}.jpg)\t(.*)$/; print "convert -negate
-normalize -pen black -draw 'fillRectangle 5,10
260,25' -pen yellow -draw 'text 10,10 \"$1 $2\"'
$1 jpg/$1\n";
}

:::::::::::::: dates.pl :::::::::::::: # return day, hour,
minute from times.c output

$dir=$ARGV[0]; chdir $dir;

@times=`ls -1 *.jpg | ../p/times`; @rads=`cat *.rad`;

print("day\thour\tminute\tmean\tstdev\tmax\n");
while($ln=shift(@times)){
  $rd=shift(@rads); $rd=~s/ {1,}/\t/g; chomp($ln);
  $ln=~/.*\t.* .* (..) (..):(..)/; if($ln){
    print("$1\t$2\t$3".$rd); }
}

:::::::::::::: dirlist.pl :::::::::::::: #!/usr/bin/perl #
Dump vec elements from directory

$dir=$ARGV[0];

opendir(DIR,$dir); while($line=readdir(DIR)){
    if($line=~/(\d+Z?\d*)/){ print "$line\n";
    } }

closedir(DIR);

:::::::::::::: dirnames.pl :::::::::::::: #!/usr/bin/perl #
Truncate leading path names from dir list

while($in=<>){ if($in=~/.+\/(\w+)\/?$/){
    print "$1\n"; }
}

:::::::::::::: d.pl :::::::::::::: @a=<>;

$i=1; while($a=shift(@a)){
print "$i\t$a"; $i++; system($a);
}

:::::::::::::: mkdate.pl :::::::::::::: #!/usr/bin/perl #
Convert vector filenames to date/time from directory

opendir(DIR,$ARGV[0]);

$ym=$ARGV[0]; $ym=~/\/{1}(\w{3}).*([0-9]{2})\/?$/;
$year=$2; $mon=$1;

@name=("jan","feb","mar","apr","may","jun","jul","aug","sep","oct",
       "nov","dec");

$i=0; for(@name){
  if($name[$i] eq lc($mon)){ last;
  } $i++;
} $mon=$i+1;

# prints year, month, day, hour, minute
while($entry=readdir(DIR)){
  $min=(); if($entry=~/^([0-9]{2})([0-9]{2})$/){
    $day=$1; $hour=$2; $min=0; }
  if($entry=~/^([0-9]{2})([0-9]{2})([0-9]{2})$/){ $day=$1;
    $hour=$2; $min=$3;
  } if($entry=~/^([0-9]{2})([0-9]{2})Z([0-9]{1,2})$/){
    $day=$3; $hour=$1; $min=$2;
  } if($min ne ()){
    $code=$year*100000000+$mon*1000000+$day*10000+$hour*100+$min;
    print "$code\n";
  } }

closedir(DIR);

:::::::::::::: mkenvivec.pl :::::::::::::: #!/usr/bin/perl

@a=<>; # slurp stdin

while($in=shift(@a)){ if(!($in=~/^\w/i)){ # numeric entry
$in=~s/ {1,}/\t/; # convert all space
sequences to single tabs $in=~s/^\t//; #
remove leading tab print $in;
} }

:::::::::::::: mktab.pl :::::::::::::: @a= <>; foreach $i
( @a ) {
  $i=~s/^\t//; # chop leading tabs $i=~s/\t/ \& /g ;
  $i=~s/$/ \\\\/ ; chop($i); # remove \n from end print
  "$i \\hline\n";
}

:::::::::::::: plot.pl :::::::::::::: open(STAT,"stats");

$i=0; while($line=<STAT>){
    $line=~/^([1-3]).([0-9]+).(bc\w+\.jpg)/; $disk=$1;
    $count=$2; $name=$3;

    $num=`echo $name | ./timenum`;

#    $i++; # $p=$i/27846*100; #    print STDERR "$name
$disk\t$p\%\n";
    print "$num $count $name\n"; }

close(STAT);

:::::::::::::: plt4.pl :::::::::::::: @in=<>;
@figs=grep(/\\input{s6stat\/[0-9]+\.tex.*/,@in);
   @times=`grep Hours [0-9]*.tex`;

   while($fig=shift @figs){
       $fig=~/.*s6stat\/([0-9]+)\.tex.*/; $fig=$1;
       if(!grep(/^$fig\.tex.*/,@times)){ # remove
       time correlation vars
   push @ps,$1.".ps"; }
   }

   while(@ps){ $out=join " ",shift
       @ps,shift @ps,shift @ps,shift @ps;
       $fig=sprintf("%02d",++$i); push @run,"mpage
       -o -4 -c -dp ".$out." >fig".$fig.".ps\n";
   }

   open(OUT,">mk4.sh"); print OUT @run; close(OUT);

   system("sh mk4.sh"); # make those figs!

# create the lyx plot file @fourfigs=`ls fig??.ps`; $i=0;
   while($figln=shift @fourfigs){
       chop $figln; #
   $fig=sprintf("Figure %02d",++$i); $fig=" ";
       push @out,"\\begin{figure} {\\centering
```

```
   \\resizebox*{6in}{6in}{".
   "\\includegraphics{s6stat/".
   $figln."}} \\par} \\caption{".$fig."}
   \\end{figure}\n";
   }

   open(OUT,">figs.tex"); print OUT @out;
   close(OUT);

#\begin{figure} {\centering
\includegraphics{s6stat/fig01.ps} \par} \caption{a}
\end{figure}

:::::::::::::: scan.pl ::::::::::::::
opendir(DIR,"cdimgs");

while($line=readdir(DIR)){ if($line=~/^bc/i){
$stamp=`echo $line | ./times`; system("convert
cdimgs/$line gray:temp.bin"); $stats=`./mask
<temp.bin`; chop($stats); unlink("temp.bin");

print STDERR "$stats\t$stamp"; print
"$stats\t$stamp";

# skip 23 images #  $i=0; #       do{ #
$line=readdir(DIR); # if($line=~/^bc/i){ #
$i++; #   } #     } while($i<23);
    } }

closedir(DIR);

:::::::::::::: statrep.pl :::::::::::::: # report on stats
file (allstats.txt)

$statfl="allstats.txt"; # set the input filename

@vars=`grep predicts $statfl`; @intercept=`grep Intercept
$statfl`; @slope=`grep '^xy' $statfl`; @R2=`grep R-Squared
$statfl`; @DF_p=`grep p-value $statfl`;

$i=1; while($vars=shift @vars){
  $vars=~/^.+"(.+) predicts (.+)"/; $vX=$1;$vY=$2;

  $int=shift @intercept;
  $int=~/^.Intercept. +(-?\d+\.\d*e?-?\+?\d*)
  +(-?\d+\.\d*e?-?\+?\d*)/; $int=$1; $interr=$2;

  $sl=shift @slope; $sl=~/^xy.+\] +(-?\d+\.\d*e?-?\+?\d*)
  +(-?\d+\.\d*e?-?\+?\d*)/; $sl=$1; $slerr=$2;

  $r2=shift @R2; $r2=~/^Multiple R-Squared: +(0.\d+)/;
  $r2=$1;

  $df_p=shift @DF_p; $df_p=~/^.* on 1 and (\d+) degrees/;
  $df=$1; $df_p=~/^.*p-value: +(\d+\.?\d*e?-?\d*)/; $p=$1;

  $frmfdef="%0.4f"; # floating point default format
  $frmedef="%0.3e"; # sci notation default format
  $frmsl=$frmfdef; # default format specifier
  $frmint=$frmfdef; # default format specifier
  $frmserr=$frmfdef; # default format specifier
  $frmierr=$frmfdef; # default format specifier
  $frmr2=$frmfdef; # default format specifier
  $frmp=$frmfdef; # default format specifier

  if(abs $sl<1e-3){ # we have small numbers
    $frmsl=$frmedef; # use sci-notation
  } if(abs $int<1e-3){ # we have small numbers
    $frmint=$frmedef; # use sci-notation } if(abs
  $slerr<1e-3){ # we have small numbers
    $frmserr=$frmedef; # use sci-notation } if(abs
  $interr<1e-3){ # we have small numbers
    $frmierr=$frmedef; # use sci-notation } if(abs
  $r2<1e-3){ # we have small numbers
    $frmr2=$frmedef; # use sci-notation } if(abs $p<1e-3){
  # we have small numbers
    $frmp=$frmedef; # use sci-notation }

  $frm="\$%s=(".$frmsl."\\pm".$frmserr.")%s".
"+(".$frmint."\\pm".$frmierr."), ".
"R^{2}=".$frmr2.", ".  "p=".$frmp.", ".
"DF=%d\$\n";

  $result=sprintf($frm,$vY,$sl,$slerr,$vX,$int,$interr,$r2,$p,$df);

  # create a file for each result
  $fnm=sprintf("%d.tex",$i++); system("echo
  >".$fnm." \"\\".$result."\"");
}
```

```
:::::::::::::: vecstrip.pl :::::::::::::: #!/usr/bin/perl #
Strip LINE and POINT for acad format vecs

open(IN,$ARGV[0]); while($line=<IN>){
    if($line=~/ +([0-9]+\.[0-9]+) +([0-9]+\.[0-9]+)/){
print "$1 $2\n";
    } }

closedir(DIR);
```

A-3 C Programs

```
:::::::::::::: ex2c.c :::::::::::::: /*
+--------------------------------------------------------------------
 * |@@@@@@@@@@   @@@| EASI/PACE V6.0, Copyright
 (c) 1995.              | * |@@@@@@ ***     @|
 | * |@@@  *******    @| PCI Inc., 50 West Wilmot Street,
 | * |@@  *********  @@| Richmond Hill, Ontario,
 L4B 1M5, Canada.          | * |@     *******  @@@|
 | * |@       *** @@@@@@| All rights reserved. Not
 to be used, reproduced    | * |@@@     @@@@@@@@@@|
 or disclosed without permission.                    | *
+--------------------------------------------------------------------
 * * $Log: ex2c.c,v $ * Revision 54.3  1997/10/04 16:04:21
 warmerda * Avoid warnings.  * * Revision 54.2  1995/12/19
 04:30:53  warmerda * Update to V6.0 copyright headers.
 * * Revision 54.1  1994/10/28  17:51:13  warmerda *
 Update to V5.4.  * * Revision 53.5  1994/10/15  20:23:00
 warmerda * Ensure that help parses correctly with modern
 'manual' program and * antique helpmake.c.  * * Revision
 53.4  1994/08/31  01:43:57  nathalie * title and keywords.
 * * Revision 53.3  1994/07/05  00:19:31  warmerda *
 Updated IMPStatus() argument list.  * * Revision 53.2
 1993/11/22  22:50:50  shih * Updating copyright message.
 * * Revision 53.1  1993/10/27  21:33:01  shih * Updated
 to V5.3.  * * Revision 1.6  1993/05/10  19:05:03  wilson
 * Minor documentation correction * * Revision 1.5
 1993/04/14  12:57:51  wilson * Reformat comment lines *
 * Revision 1.4  1993/04/09  14:38:38  wilson * Change
 name of program to EX2C in documentation * * Revision
 1.3  1993/03/03  15:05:52  wilson * Updated for V5.1
 Programmer's Tutorial * * Revision 1.2  1992/05/11
 22:00:39  warmerda * Removed <math.h> and fixed up
 comments a bit.  * * Revision 1.1  1992/04/13  14:54:46
 v051 * Initial revision * */

/* C+ C@title{rect}{C code to read display coordinates
and generate vectors} C C The code is called from
RECTIFY.EAS, and cannot be used C on its own.
Please refer to the RECTIFY documentation C C- */

/*
--------------------------------------------------------------------
*/ /*    Include "pci.h" in all C
programs.        */ /*
--------------------------------------------------------------------
*/ #include "pci.h"

RCSID("$Id: ex2c.c,v 54.3 1997/10/04 16:04:21 warmerda
Exp $")

int main( int main_argc, char **main_argv )

{ /*
--------------------------------------------------------------------
*/ /*    Declare parameters for
IMPStatus.  */ /*
--------------------------------------------------------------------
*/
char report[65]; float    vector[16], power[16];
int argcnt[3], i; void    *args[3];

/*
--------------------------------------------------------------------
*/ /*    Initialize argument list
for IMPStatus.         */ /*
--------------------------------------------------------------------
*/
args[0] = (void *) vector; args[1] = (void *)
power; args[2] = (void *) report;

/*
--------------------------------------------------------------------
*/ /*    Get parameters using
IMPStatus.    */ /*
--------------------------------------------------------------------
*/
IMPStatus ("VECTOR, POWER, REPORT;", "R      , R
   , C    ;", "16     , 16    , 64     ;",
   "1    , *   , 1  ;", "EX2C.",
   "FORCE", argcnt, args, main_argc,
   main_argv );

/*
--------------------------------------------------------------------
*/ /*    Print title page header for
report using IMPPage.      */ /*
--------------------------------------------------------------------
*/
IMPPage (Report, 0);

/*
--------------------------------------------------------------------
*/ /*    For each value in VECTOR array,
*/ /*    if value is negative, print warning using
IMPErrReal        */ /*  else print result
of power function calculation.       */ /*
--------------------------------------------------------------------
*/
for (i = 0 ; i < argcnt[0]; i++) {
    if (vector[i] < 0.0) IMPErrReal (11,
ERRTYP_WARNING, "VECTOR", vector[i], i);
    else fprintf (Report, " %f raised to %f power
is %f\n",
vector[i], power[i], pow(vector[i],
power[i])); }

/*
--------------------------------------------------------------------
*/ /*    Exit program using
IMPReturn.      */ /*
--------------------------------------------------------------------
*/
IMPReturn(); }

:::::::::::::: fc.c :::::::::::::: /* compute high cloud
counts and cyclonic index
   ./findcloud {maskfile} <datafile >resultfile */

/*#define NOIMG*/

/* Apply the cyclone tracker mask */

#include <math.h> #include <stdio.h>

#define UBYTE unsigned char

struct bounds{ float x,y;
} *bounds;

FILE *maskfile; UBYTE *frame,*mask; float A,B,L,l,e,D,r,d;
float *index; int IM,SUBIM,amin,amax;

/* Read in the image */

fill() {
  int i,count; UBYTE val; /*  short int val; */

  count=0; for(i=0;i<IM*IM;i++){
    fread(&val,sizeof(UBYTE),1,stdin); /*
    fread(&val,sizeof(short int),1,stdin); */
    frame[i]=(UBYTE)val; if( (int)(frame[i]/152*mask[i])
    ) count++;
  }

#ifdef NOIMG printf("%d\t",count);
#else fprintf(stderr,"Input read, %d highcloud
  pixels\n",count);
#endif }

setup() {
  int i,j;

  A=60*60; B=A/8; L=sqrt(A); l=sqrt(B); e=1.5*L; /* 2*L */
  D=e+l; r=e+l/2; /* 45 deg. in radians */ d=r*sin(0.7854);
  SUBIM=IM-2*D;

  /* threshold of high cloud for frame consideration */
  amin=(0.5*A); /*  amax=(0.9*A);*/ amax=A; /* max cap
  interfering with true results */

#ifndef NOIMG fprintf(stderr,"Image is %d on a
  side\n",SUBIM);
#endif

  frame=(UBYTE *)malloc(IM*IM); if(!frame) exit(0); fill();

  bounds=(struct bounds *)malloc(9*sizeof(struct bounds));
  if(!bounds) exit(0);

  bounds[0].x=D-L/2; bounds[0].y=D-L/2;

  bounds[1].x=D-l/2; bounds[1].y=0;

  bounds[2].x=D+d-l/2; bounds[2].y=D-d-l/2;

  bounds[3].x=2*D-l; bounds[3].y=D-l/2;

  bounds[4].x=D+d-l/2; bounds[4].y=D+d-l/2;
```

```
  bounds[5].x=D-l/2; bounds[5].y=2*D-l;

  bounds[6].x=D-d-l/2; bounds[6].y=D+d-l/2;

  bounds[7].x=0; bounds[7].y=D-l/2;

  bounds[8].x=D-d-l/2; bounds[8].y=D-d-l/2;

#ifndef NOIMG fprintf(stderr,"Definitions calculated\n");
#endif }

int peek(int X, int Y, int size) {
  int i,j,count;

  count=0; for(j=0;j<size;j++)
    for(i=0;i<size;i++)
      count+=(int)(255-frame[(Y+j)*IM+X+i])/152;

  return(count); }

scan() {
  int i,j,k,b[8],padA,padB; float acnt,bmean,maxindex;
  for(j=0;j<SUBIM;j++){
    for(i=0;i<SUBIM;i++){
      acnt=(float)peek(bounds[0].x+i,bounds[0].y+j,L);
      if(acnt>=amin && acnt<amax){ /* do mean and index
      calc */
for(k=1;k<9;k++)
  b[k-1]=peek(bounds[k].x+i,bounds[k].y+j,l);

for(k=0,bmean=0;k<8;k++) bmean+=b[k];
bmean/=8; index[j*SUBIM+i]=acnt/(bmean+1);
      } else index[j*SUBIM+i]=0;
    } }

  for(i=0,maxindex=0;i<SUBIM*SUBIM;i++)
    if(index[i]>maxindex) maxindex=index[i];
  fprintf(stderr,"Maximum index value: %4.2f\n",maxindex);
#ifndef NOIMG
  fprintf(stderr,"Storing image\n"); /*
  for(i=0;i<SUBIM*SUBIM;i++)
    fwrite(&index[i],sizeof(float),1,stdout); */

  acnt=0; padA=(int)((IM-SUBIM)/2); padB=IM-SUBIM-padA;
  /* Store blank border top */ for(i=0;i<padA*IM;i++)
    fwrite(&acnt,sizeof(float),1,stdout); /* Store data
  lines */ for(i=0;i<SUBIM;i++){
    /* Store blank left edge */
    fwrite(&acnt,sizeof(float),padA,stdout);
    /* Store image data */
    fwrite(&index[i*SUBIM],sizeof(float),SUBIM,stdout);
    /* Store blank right edge */
    fwrite(&acnt,sizeof(float),padB,stdout);
} /* Store blank border bottom */ for(i=0;i<padB*IM;i++)
    fwrite(&acnt,sizeof(float),1,stdout); #endif }

main(int argc, char **argv) {
  int i,j;

  IM=512; /* the original image size */

  mask=(UBYTE *)malloc(IM*IM); if(!mask){
    fprintf(stderr,"Error allocating mask memory!\n");
    exit(0);
  } for(i=0;i<IM*IM;i++) mask[i]=1; /* default to all
  pixels used */ if(argc==2){ /* we have a maskfile */
    maskfile=fopen(argv[1],"r"); if(!maskfile){
      fprintf(stderr,"Error reading maskfile!\n"); exit(0);
    } fread(mask,1,IM*IM,maskfile); /* fill mask from file
    */ fclose(maskfile); fprintf(stderr,"Maskfile read\n");
  } else fprintf(stderr,"No masking used on raw data\n");

  setup();

  index=(float *)calloc(SUBIM*SUBIM,sizeof(float));
  if(!index) exit(0);

  /* Calls to peek go here */ scan();

  free(bounds); free(index);

  free(mask); free(frame);
}

::::::::::::: fill.c ::::::::::::: /* fill the damned
image - can't this be done in other software !? */

#include <stdio.h>

#define UBYTE unsigned char

main() {
  int x,y,adr; UBYTE in; UBYTE *mem;

  mem=(UBYTE *)malloc(512*512); for(x=0;x<512*512;x++){
    mem[x]=getchar(); if(mem[x]>0) mem[x]=1; /* set 255
    values to 1 */
  }

  for(x=0;x<512;x++) for(y=0;y<512;y++){ /* fill points
    south */
      adr=y*512+x; if(mem[adr]) mem[adr]=0; else break;
    }

  for(x=0;x<512*512;x++) putchar(mem[x]);

  free(mem); }

::::::::::::: findcloud.c ::::::::::::: /* compute high
cloud counts and cyclonic index
   ./findcloud {maskfile} <datafile >resultfile */

/* #define NOIMG */

/* Apply the cyclone tracker mask */

#include <math.h> #include <stdio.h>

#define UBYTE unsigned char

struct bounds{ float x,y;
} *bounds;

FILE *maskfile; UBYTE *frame,*mask; float A,B,L,l,e,D,r,d;
float *index; int IM,SUBIM,amin,amax;

/* Read in the image */

fill() {
  int i,count;

  count=0; for(i=0;i<IM*IM;i++){
  /* invert the images */ /*
frame[i]=255-getchar(); For negative (raw) images */
    frame[i]=getchar(); if( frame[i]*mask[i]>=152 )
    count++; /* high cloud over water */
  }

#ifdef NOIMG printf("%d\t",count);
#else fprintf(stderr,"Input read, %d highcloud
  pixels\n",count);
#endif }

setup() {
  int i,j;

  A=60*60; B=A/8; L=sqrt(A); l=sqrt(B); e=1.5*L; /* 2*L */
  D=e+l; r=e+l/2; /* 45 deg. in radians */ d=r*sin(0.7854);
  SUBIM=IM-2*D;

  /* threshold of high cloud for frame consideration */
  amin=(0.5*A); /* amax=(0.9*A);*/ amax=A; /* max cap
  interfering with true results */

#ifndef NOIMG fprintf(stderr,"Image is %d on a
  side\n",SUBIM);
#endif

  frame=(UBYTE *)malloc(IM*IM); if(!frame) exit(0); fill();

  bounds=(struct bounds *)malloc(9*sizeof(struct bounds));
  if(!bounds) exit(0);

  bounds[0].x=D-L/2; bounds[0].y=D-L/2;

  bounds[1].x=D-l/2; bounds[1].y=0;

  bounds[2].x=D+d-l/2; bounds[2].y=D-d-l/2;

  bounds[3].x=2*D-l; bounds[3].y=D-l/2;

  bounds[4].x=D+d-l/2; bounds[4].y=D+d-l/2;

  bounds[5].x=D-l/2; bounds[5].y=2*D-l;

  bounds[6].x=D-d-l/2; bounds[6].y=D+d-l/2;
```

```
  bounds[7].x=0; bounds[7].y=D-1/2;

  bounds[8].x=D-d-1/2; bounds[8].y=D-d-1/2;

  /* convert to centre-relative (D,D)->(0,0) */
  for(i=0;i<9;i++){
    bounds[i].x-=D; bounds[i].y-=D;
  }

#ifndef NOIMG fprintf(stderr,"Definitions calculated\n");
#endif }

int peek(int X, int Y, int size) {
  int i,j,count,pos;

  count=0; for(j=0;j<size;j++)
    for(i=0;i<size;i++){ pos=(Y+j)*IM+X+i;
      if((X+i)<0||(X+1)>IM||(Y+j)<0||(Y+j)>IM) /* out of
      bounds */
return(-1); if(frame[(Y+j)*IM+X+i]>=152) count++;
    } return(count);
}

scan() {
  int i,j,k,b[8],pkval; float acnt,bmean,maxindex;
  for(j=0;j<IM;j++){
    for(i=0;i<IM;i++){ if(!mask[j*IM+i]){ /* centre over
      landmass */
index[j*IM+i]=0; break;
      } acnt=(float)peek(bounds[0].x+i,bounds[0].y+j,L);
      if(acnt>=amin && acnt<amax){ /* do mean and index
      calc */
for(k=1,pkval=0;k<9 && pkval>=0;k++){
  pkval=peek(bounds[k].x+i,bounds[k].y+j,l);
  if(pkval>=0)
    b[k-1]=pkval; else{
    index[j*IM+i]=0; break;
  } } if(pkval>=0){
  for(k=0,bmean=0;k<8;k++) bmean+=b[k];
  bmean/=8; index[j*IM+i]=acnt/(bmean+1);
} } else index[j*IM+i]=0;
    } }

  for(i=0,maxindex=0;i<IM*IM;i++) if(index[i]>maxindex)
    maxindex=index[i];
  fprintf(stderr,"Maximum index value: %4.2f\n",maxindex);
#ifdef NOIMG
  printf("%f\n",maxindex); #else
  fprintf(stderr,"Storing image\n");
  fwrite(index,sizeof(float),IM*IM,stdout);
#endif }

main(int argc, char **argv) {
  int i,j,k,tst;

  IM=512; /* the original image size */

  mask=(UBYTE *)malloc(IM*IM); if(!mask){
    fprintf(stderr,"Error allocating mask memory!\n");
    exit(0);
  } for(i=0;i<IM*IM;i++) mask[i]=1; /* default to all
  pixels used */ if(argc==2){ /* we have a maskfile */
    maskfile=fopen(argv[1],"r"); if(!maskfile){
      fprintf(stderr,"Error reading maskfile!\n"); exit(0);
    } fread(mask,1,IM*IM,maskfile); /* fill mask from file
    */ fclose(maskfile); fprintf(stderr,"Maskfile read\n");
  } else fprintf(stderr,"No masking used on raw data\n");

  setup();

  index=(float *)calloc(IM*IM,sizeof(float)); if(!index)
  exit(0);

  scan();

  free(bounds); free(index);

  free(mask); free(frame);
}

:::::::::::::: grid.c :::::::::::::: #include <stdio.h>

 #include <stdlib.h>
 #include <ctype.h>

 #define
CH_IN 3 /* Number of allowable digits for grid pair #s */

 struct grid{
float x,y;
 int i,j;
};

 void main();
 int rdgrid(struct grid **);
 void
dumpgrid(struct grid *,int);

 void main()
 {
int xpos,ypos,gridpt,gridlen,chnum;
 char
input[CH_IN];
 struct grid *thegrid;

 gridlen=rdgrid(&thegrid); /* slurp the standard
     grid intersections */


xpos=ypos=0;
 gridpt=0; /* start at first point */

 do{
printf("Pair #%d: ",gridpt+1);

/* Spool input, allowing
for a blank return line */
for(chnum=0;chnum<CH_IN;input[chnum++]=0);

 chnum=0;
 while(
(input[chnum++]=getchar())!='\n'
&& chnum<CH_IN );
 fflush(stdin);
input[chnum]=0; /* Make sure string has
a terminator */

 /* A return on a blank
line */
 if(input[0]=='\n'){
/*  VDCursorPos(video,VD_READ,&xpos,&ypos); */

 thegrid[gridpt].i=xpos;
/* Store mouse locationn */
thegrid[gridpt].j=ypos;
 printf("%d
- (%0.2f,%0.2f) -> (%d,%d)\n" \
 ,gridpt+1,thegrid[gridpt].x,thegrid[gridpt].y
 \
 ,thegrid[gridpt].i,thegrid[gridpt].j);


 if(gridpt<gridlen-1) gridpt++;
}

/* Jump to grid pair */
if(isdigit(input[0])){
sscanf(input,"%2d",&chnum);

 if(chnum<=gridlen){ /* Check
bounds */
gridpt=chnum-1;
 printf("%d
- (%0.2f,%0.2f) ->
(%d,%d)\n" \
 ,gridpt+1,thegrid[gridpt].x
 \
 ,thegrid[gridpt].y,thegrid[gridpt].i
 \
 ,thegrid[gridpt].j);
}
 }
} while(input[0]!='q');

dumpgrid(thegrid,gridlen);
 free(thegrid);
}

 int rdgrid(struct grid **array)
 {
FILE *gridfile;
 struct grid *vals;
 int i,len;
float x,y;

 gridfile=fopen("gridfile.txt","r");
```

```
 if(!gridfile){
puts("Couldn't open standard grid defs");

 exit(0);
}

len=0;
 while(!feof(gridfile))
if(fgetc(gridfile)=='\n') len++;

 rewind(gridfile);

 *array=(struct grid
*)calloc(len,sizeof(struct grid));
 vals=*array;

 if(!(*array)){
puts("Error reserving grid memory");

 exit(0);
}

for(i=0;i<len;i++){
 fscanf(gridfile,"%f%f",&x,&y);

 vals[i].x=x;
 vals[i].y=y;
}

printf("Read %d grid points\n\n",len);
fclose(gridfile);
 return(len);
}

 void dumpgrid(struct grid *grid,int len)
 {
FILE *out;
 int i;

 out=fopen("rectify.gcp","w");
 if(!out){
puts("Error writing output!");
 return;
}

/* Dump GCPs in x1 y1 x2 y2 format */
for(i=0;i<len;i++)
fprintf(out,"%d %d %0.2f
%0.2f\n",grid[i].i,grid[i].j \
 ,grid[i].x,grid[i].y);

 fclose(out);
}


:::::::::::::: hist.c :::::::::::::: #include <stdio.h>

main() {
  int i,vec[256]; unsigned char in;

  for(i=0;i<256;) vec[i++]=0;

  while(!feof(stdin)){ in=getchar(); vec[in]++;
  }

  for(i=0;i<256;i++) printf("%d: %d\n",i,vec[i]);
}

:::::::::::::: mask.c :::::::::::::: /*#define NOIMG*/

/* Apply the cyclone tracker mask */

#include <math.h> #include <stdio.h>

#define UBYTE unsigned char

struct bounds{ float x,y;
} *bounds;

UBYTE *frame; float A,B,L,l,e,D,r,d; float *index; int
IM,SUBIM,amin,amax;

/* Read in the image */

fill() {
  int i,count; UBYTE val; /*  short int val; */

  count=0; for(i=0;i<IM*IM;i++){
    fread(&val,sizeof(UBYTE),1,stdin); /*
    fread(&val,sizeof(short int),1,stdin); */
    frame[i]=(UBYTE)val; if((int)(255-frame[i])/152)
    count++;
  }

#ifdef NOIMG printf("%d\t",count);
#else fprintf(stderr,"Input read, %d highcloud
  pixels\n",count);
#endif }

setup() {
  int i,j;

  A=60*60; B=A/8; L=sqrt(A); l=sqrt(B); e=1.5*L; /* 2*L */
  D=e+l; r=e+l/2; /* 45 deg. in radians */ d=r*sin(0.7854);
  SUBIM=IM-2*D;

  /* threshold of high cloud for frame consideration */
  amin=(0.5*A); /*  amax=(0.9*A);*/ amax=A; /* max cap
  interfering with true results */

#ifndef NOIMG fprintf(stderr,"Image is %d on a
  side\n",SUBIM);
#endif

  frame=(UBYTE *)malloc(IM*IM); if(!frame) exit(0); fill();

  bounds=(struct bounds *)malloc(9*sizeof(struct bounds));
  if(!bounds) exit(0);

  bounds[0].x=D-L/2; bounds[0].y=D-L/2;

  bounds[1].x=D-l/2; bounds[1].y=0;

  bounds[2].x=D+d-l/2; bounds[2].y=D-d-l/2;

  bounds[3].x=2*D-l; bounds[3].y=D-l/2;

  bounds[4].x=D+d-l/2; bounds[4].y=D+d-l/2;

  bounds[5].x=D-l/2; bounds[5].y=2*D-l;

  bounds[6].x=D-d-l/2; bounds[6].y=D+d-l/2;

  bounds[7].x=0; bounds[7].y=D-l/2;

  bounds[8].x=D-d-l/2; bounds[8].y=D-d-l/2;

#ifndef NOIMG fprintf(stderr,"Definitions calculated\n");
#endif }

int peek(int X, int Y, int size) {
  int i,j,count;

  count=0; for(j=0;j<size;j++)
    for(i=0;i<size;i++)
      count+=(int)(255-frame[(Y+j)*IM+X+i])/152;

  return(count); }

scan() {
  int i,j,k,b[8],padA,padB; float acnt,bmean,maxindex;
  for(j=0;j<SUBIM;j++){
    for(i=0;i<SUBIM;i++){
      acnt=(float)peek(bounds[0].x+i,bounds[0].y+j,L);
      if(acnt>=amin && acnt<amax){ /* do mean and index
      calc */
for(k=1;k<9;k++)
  b[k-1]=peek(bounds[k].x+i,bounds[k].y+j,l);

for(k=0,bmean=0;k<8;k++) bmean+=b[k];
bmean/=8; index[j*SUBIM+i]=acnt/(bmean+1);
      } else index[j*SUBIM+i]=0;
    } }

  for(i=0,maxindex=0;i<SUBIM*SUBIM;i++)
    if(index[i]>maxindex) maxindex=index[i];
  fprintf(stderr,"Maximum index value: %4.2f\n",maxindex);
#ifndef NOIMG
  fprintf(stderr,"Storing image\n"); /*
  for(i=0;i<SUBIM*SUBIM;i++)
    fwrite(&index[i],sizeof(float),1,stdout); */

  acnt=0; padA=(int)((IM-SUBIM)/2); padB=IM-SUBIM-padA;
  /* Store blank border top */ for(i=0;i<padA*IM;i++)
    fwrite(&acnt,sizeof(float),1,stdout); /* Store data
  lines */ for(i=0;i<SUBIM;i++){
    /* Store blank left edge */
    fwrite(&acnt,sizeof(float),padA,stdout);
```

```
   /* Store image data */
   fwrite(&index[i*SUBIM],sizeof(float),SUBIM,stdout);
   /* Store blank right edge */
   fwrite(&acnt,sizeof(float),padB,stdout);
} /* Store blank border bottom */ for(i=0;i<padB*IM;i++)
   fwrite(&acnt,sizeof(float),1,stdout); #endif }

main() {
  int i,j;

  IM=512; /* the original image size */ setup();

  index=(float *)calloc(SUBIM*SUBIM,sizeof(float));
  if(!index) exit(0);

  /* Calls to peek go here */ scan();

  free(bounds); free(index);

  free(frame); }

:::::::::::::: noland.c :::::::::::::: /* remove land
masses from raw images
   Assumes 512x512

   ./noland datafile <maskfile >resultfile */

#include <stdio.h> #define UBYTE unsigned char

main(int argc,char **argv) {
  int i; FILE *infile; UBYTE mask,data;

  infile=fopen(argv[1],"r"); if(!infile){
    puts("File read error"); exit(0);
  }

  for(i=0;i<512*512;i++){ mask=getchar();
    data=fgetc(infile);
if(!mask) putchar(70); /* high is (255-x)/152 ;
x=255 -> 0/152 = low */ else putchar(data);
  }

  close(infile); }

:::::::::::::: rectgdb.c :::::::::::::: /* +
C@title{RECTIFY}{Flow Normalisation for Cyclonic
Development} C C       The code is called from RECTIFY.EAS,
and cannot be used C    on its own. Please refer to
the RECTIFY documentation C C- */

/*
---------------------------------------------------------------------
*/ /*    Include "pci.h" in all C
programs.        */ /*
---------------------------------------------------------------------
*/ #include "pci.h" #include "string.h"

int dumpcount=0;

struct grid { float x,y; float i,j;
};

#define CH_IN 3 /* Number of allowable digits for grid
pair #s */

int rdgrid(struct grid **); /* void dumpgrid(struct grid
*,int); */ void dumpgcps(FILE *dbase,struct grid *grid,int
len,unsigned char *name);

RCSID("$Id: ex2c.c,v 54.3 1997/10/04 16:04:21 warmerda
Exp $")

int main( int main_argc, char **main_argv ) {
FILE *dbase_in,*dbase_out; int
xpos,ypos,gridpt,gridlen,seg,nvert,i; char
input[CH_IN]; struct grid *thegrid; GDBLayer
layer; GDBShape *shape; GDBShapeId shape_id;
GDBVertex *verts;

unsigned char
gridverts[]={1,5,9,13,17,23,29,37,43,51,57};
unsigned char
ptverts[]={63,64,65,66,67,18,30,44,58}; char
seg_name[30];

VInfo_t vecinfo; int *nvertex,*type; int32
*group,*attribute; double **vertices; double
*buffer; int buffer_size;
```

```
/*
---------------------------------------------------------------------
*/ /*    Declare parameters for
IMPStatus.  */ /*
---------------------------------------------------------------------
*/
char report[65]; float   vector[16], power[16];
int argcnt[3]; void    *args[3];

/*
---------------------------------------------------------------------
*/ /*    Initialize argument list
for IMPStatus.        */ /*
---------------------------------------------------------------------
*/
args[0] = (void *) vector; args[1] = (void *)
power; args[2] = (void *) report;

/*
---------------------------------------------------------------------
*/ /*    Get parameters using
IMPStatus.    */ /*
---------------------------------------------------------------------
*/
IMPStatus (";", ";", ";", ";", "RECTGDB.", "FORCE",
   argcnt, args, main_argc, main_argv );

/*
---------------------------------------------------------------------
*/ /*    Print title page header for
report using IMPPage.     */ /*
---------------------------------------------------------------------
*/
IMPPage (Report, 0);

/* Program code here */

dbase_in=IDBOpen("apr94","r+"); /* The Source
Database */ if(!dbase_in){
  puts("Error linking to source database!");
  return(0);
}

dbase_out=IDBOpen("rec94","r+"); /* The Destination
Database */ if(!dbase_out){
  puts("Error linking to destination database!");
  return(0);
}

gridlen=rdgrid(&thegrid); /* slurp the standard
     grid intersections */

/* Body */

seg=GDBSegNext(dbase_out,SEG_VEC,0); /* get first
vec segment */

do{ /* read all vector segments */ /*
  Find out the name of the segment */
  IDBSegInfoIO(dbase_out,seg,IDB_READ,NULL,NULL,&seg_name,NULL,NULL);

  dumpcount++; /* make sure count = segment # */
  layer=GDBGetLayer(dbase_out,seg); /* get layer
  handle */ if(GDBGetNumShapes(layer)!=10){ /*
  wrong count of shapes */
    printf("Vector #%d - wrong shape
    count\n",dumpcount); goto skip;
  }

  /* shape 0 vertices directly from grid
  alignment */ shape_id=GDBGetFirst(layer);
  /* get first shape ID */
  verts=GDBGetVertices(layer,shape_id,&nvert);
  if(nvert!=68){ /* wrong count of points */
    printf("Vector #%d - wrong vertex count in
    shape 0\n"); goto skip;
  }

  for(i=0;i<11;i++){ /* original coords */
    thegrid[i].x=verts[gridverts[i]-1].x;
    thegrid[i].y=verts[gridverts[i]-1].y;
  } HFree(verts); /* deallocate vertex memory */

  for(i=0;i<9;i++){ /* read comma head and flow
    width */ shape_id=GDBGetNext(layer,shape_id);
    verts=GDBGetVertices(layer,shape_id,&nvert);
    thegrid[11+i].x=verts[0].x;
    thegrid[11+i].y=verts[0].y; HFree(verts);
  }
```

```
  /* Write to PCIDSK GCP Segment */ /* GCPs must
   go to SOURCE database */
  dumpgcps(dbase_in,thegrid,gridlen,seg_name); /*
dumpgrid(thegrid,gridlen); */

skip: } while((seg=GDBSegNext(dbase_out,SEG_VEC,seg))!=-1);
/* next segment */

free(thegrid); IDBClose(dbase_out);

/*
---------------------------------------------------------------------
*/ /*    Exit program using
IMPReturn.      */ /*
---------------------------------------------------------------------
*/
IMPReturn(); }

int rdgrid(struct grid **array) {
FILE *gridfile; struct grid *vals; int n,len;
float i,j; /* gcps2x and gcps2y - the transformed
coords */

gridfile=fopen("gridfile.txt","r"); if(!gridfile){
puts("Couldn't open standard grid defs");
IMPReturn();
}

len=0; while(!feof(gridfile))
if(fgetc(gridfile)=='\n') len++;
rewind(gridfile);

*array=(struct grid *)calloc(len,sizeof(struct
grid)); vals=*array; if(!(*array)){
puts("Error reserving grid memory");
IMPReturn();
}

for(n=0;n<len;n++){ fscanf(gridfile,"%f%f",&i,&j);
vals[n].i=i; vals[n].j=j;
}

printf("Read %d grid points from standard
grid.\n\n",len); fclose(gridfile); return(len);
}

void dumpgrid(struct grid *grid,int len) {
FILE *out; unsigned char file[30]="rectify00.gcp";
int i;

file[7]=(int)dumpcount/10+'0';
file[8]=dumpcount-((int)dumpcount/10)*10+'0';
out=fopen(file,"w"); if(!out){
puts("Error writing output!"); return;
}

printf("Creating file %s\n",file);

/* Dump GCPs in x1 y1 x2 y2 format */
for(i=0;i<len;i++)
fprintf(out,"%0.2f %0.2f %0.2f
 %0.2f\n",grid[i].i,grid[i].j \
 ,grid[i].x,grid[i].y);

fclose(out); }

void dumpgcps(FILE *dbase,struct grid *grid,int
len,unsigned char *name) {
  int i,j; static int segment; static int new_seg=1;
  unsigned char create;

  /* IDBGcpIo parameters */ int gcpids[255]; double
  gcps1x[255],gcps1y[255],gcps2x[255],gcps2y[255]; double
  blanks[255]; /* for elevations */

  /* find or create next gcp segment */ create=0; /* flag
  */ if(new_seg==1){ /* First time, try to find existing
  segment */
    if((segment=GDBSegNext(dbase,SEG_GCP,0))==-1) create=1;
  } else /* try to find the next segment */
    if((segment=GDBSegNext(dbase,SEG_GCP,segment))==-1)
    create=1;

  if(create){ /* We need to add a segment */
    segment=IDBSegCreate(dbase,name,"Flow
    Normalisation",SEG_GCP,0);

    if(!segment){ puts("Error creating new segment!
      Aborting!"); IMPReturn();
    } } else{
    printf("Processing transform GCP segment
    #%d\n",new_seg); /* Re-name GCP segment
    to match source vector segment name */
    IDBSegInfoIO(dbase,segment,IDB_WRITE,NULL,NULL,name,NULL,NULL);
  }

  /* Fill parameter arrays */ for(i=0,j=0;i<len;i++){ /*
  (x,y) -> (i,j) for transform */
    if(grid[i].x>=0 && grid[i].y>=0){ /* only add positive
      GCPs */ gcpids[j]=i+1; gcps1x[j]=grid[i].x;
      gcps1y[j]=grid[i].y; gcps2x[j]=grid[i].i;
      gcps2y[j]=grid[i].j; blanks[j]=0;

      j++; }
  }

  /* Write the GCP */
  IDBGcpIO(dbase,IDB_WRITE,1,segment,&j,"PIXEL","PIXEL","PIXEL","PIXEL",
   gcpids,gcps1x,gcps1y,blanks,gcps2x,gcps2y,blanks);

  /* Increase the transform GCP segment number */
  new_seg++;
}

::::::::::::: rectify.c ::::::::::::: /* +
C@title{RECTIFY}{Flow Normalisation for Cyclonic
Development} C C       The code is called from RECTIFY.EAS,
and cannot be used C    on its own. Please refer to
the RECTIFY documentation C C- */

/*
---------------------------------------------------------------------
*/ /*    Include "pci.h" in all C
programs.       */ /*
---------------------------------------------------------------------
*/ #include "pci.h"

struct grid{ float x,y; int i,j;
};

#define CH_IN 3 /* Number of allowable digits for grid
pair #s */

int rdgrid(struct grid **); void dumpgrid(struct grid
*,int);

RCSID("$Id: ex2c.c,v 54.3 1997/10/04 16:04:21 warmerda
Exp $")

int main( int main_argc, char **main_argv ) {
FILE *video; int xpos,ypos,gridpt,gridlen,chnum;
char input[CH_IN]; struct grid *thegrid;

/*
---------------------------------------------------------------------
*/ /*    Declare parameters for
IMPStatus.  */ /*
---------------------------------------------------------------------
*/
char report[65]; float   vector[16], power[16];
int argcnt[3], i; void    *args[3];

/*
---------------------------------------------------------------------
*/ /*    Initialize argument list
for IMPStatus.        */ /*
---------------------------------------------------------------------
*/
args[0] = (void *) vector; args[1] = (void *)
power; args[2] = (void *) report;

/*
---------------------------------------------------------------------
*/ /*    Get parameters using
IMPStatus.    */ /*
---------------------------------------------------------------------
*/
IMPStatus (";", ";", ";", ";", "RECT.", "FORCE",
   argcnt, args, main_argc, main_argv );

/*
---------------------------------------------------------------------
*/ /*    Print title page header for
report using IMPPage.     */ /*
---------------------------------------------------------------------
*/
IMPPage (Report, 0);
```

```
/* Program code here */

video=VDOpen("VD00");

/* Body */

gridlen=rdgrid(&thegrid); /* slurp the standard
     grid intersections */

xpos=ypos=0; gridpt=0; /* start at first point
*/ do{
printf("Pair #%d: ",gridpt+1);

/* Spool input, allowing
for a blank return line */
for(chnum=0;chnum<CH_IN;input[chnum++]=0);
chnum=0; while(
(input[chnum++]=getchar())!='\n'
&& chnum<CH_IN ); fflush(stdin);
input[chnum]=0; /* Make sure string has
a terminator */

/* A return on a blank line */
if(input[0]=='\n'){
VDCursorPos(video,VD_READ,&xpos,&ypos);
thegrid[gridpt].i=xpos;
/* Store mouse locationn */
thegrid[gridpt].j=ypos; printf("%d
- (%0.2f,%0.2f) -> (%d,%d)\n" \
 ,gridpt+1,thegrid[gridpt].x,thegrid[gridpt].y
 \
 ,thegrid[gridpt].i,thegrid[gridpt].j);

if(gridpt<gridlen-1) gridpt++; }

/* Skip to next grid pair */
if(input[0]=='n' && gridpt<gridlen-1){
gridpt++; printf("%d -
(%0.2f,%0.2f) -> (%d,%d)\n" \
 ,gridpt+1,thegrid[gridpt].x \
 ,thegrid[gridpt].y,thegrid[gridpt].i
 \ ,thegrid[gridpt].j);
}

/* Jump to grid pair */
if(isdigit(input[0]) || input[0]=='n'){
sscanf(input,"%2d",&chnum);
if(chnum<=gridlen){ /* Check
bounds */
gridpt=chnum-1; printf("%d
- (%0.2f,%0.2f) ->
(%d,%d)\n" \
 ,gridpt+1,thegrid[gridpt].x
 \
 ,thegrid[gridpt].y,thegrid[gridpt].i
 \ ,thegrid[gridpt].j);
} }

} while(input[0]!='q');

/* Finish */ dumpgrid(thegrid,gridlen); free(thegrid);

VDClose(video);

/*
---------------------------------------------------------------------
*/ /*    Exit program using
IMPReturn.      */ /*
---------------------------------------------------------------------
*/
IMPReturn(); }

int rdgrid(struct grid **array) {
FILE *gridfile; struct grid *vals; int i,len;
float x,y;

gridfile=fopen("gridfile.txt","r"); if(!gridfile){
puts("Couldn't open standard grid defs");
exit(0);
}

len=0; while(!feof(gridfile))
if(fgetc(gridfile)=='\n') len++;
rewind(gridfile);

*array=(struct grid *)calloc(len,sizeof(struct
grid)); vals=*array; if(!(*array)){
puts("Error reserving grid memory");
exit(0);
}

for(i=0;i<len;i++){ fscanf(gridfile,"%f%f",&x,&y);
vals[i].x=x; vals[i].y=y;
}

printf("Read %d grid points\n\n",len);
fclose(gridfile); return(len);
}

void dumpgrid(struct grid *grid,int len) {
FILE *out; int i;

out=fopen("rectify.gcp","w"); if(!out){
puts("Error writing output!"); return;
}

/* Dump GCPs in x1 y1 x2 y2 format */
for(i=0;i<len;i++)
fprintf(out,"%d %d %0.2f
 %0.2f\n",grid[i].i,grid[i].j \
 ,grid[i].x,grid[i].y);

fclose(out); }

::::::::::::: rectvec.c ::::::::::::: /* +
C@title{RECTIFY}{Flow Normalisation for Cyclonic
Development} C C       The code is called from RECTIFY.EAS,
and cannot be used C    on its own. Please refer to
the RECTIFY documentation C C- */

/*
---------------------------------------------------------------------
*/ /*    Include "pci.h" in all C
programs.       */ /*
---------------------------------------------------------------------
*/ #include "pci.h"

struct grid { float x,y; float i,j;
};

#define CH_IN 3 /* Number of allowable digits for grid
pair #s */

int rdgrid(struct grid **); void dumpgrid(struct grid
*,int);

RCSID("$Id: ex2c.c,v 54.3 1997/10/04 16:04:21 warmerda
Exp $")

int main( int main_argc, char **main_argv ) {
FILE *video; int xpos,ypos,gridpt,gridlen,chnum;
char input[CH_IN]; struct grid *thegrid;

VInfo_t vecinfo; int *nvertex,*type; int32
*group,*attribute; double **vertices; double
*buffer; int buffer_size;

/*
---------------------------------------------------------------------
*/ /*    Declare parameters for
IMPStatus.  */ /*
---------------------------------------------------------------------
*/
char report[65]; float    vector[16], power[16];
int argcnt[3], i; void    *args[3];

/*
---------------------------------------------------------------------
*/ /*    Initialize argument list
for IMPStatus.       */ /*
---------------------------------------------------------------------
*/
args[0] = (void *) vector; args[1] = (void *)
power; args[2] = (void *) report;

/*
---------------------------------------------------------------------
*/ /*    Get parameters using
IMPStatus.    */ /*
---------------------------------------------------------------------
*/
IMPStatus (";", ";", ";", ";", "RECTVEC.", "FORCE",
   argcnt, args, main_argc, main_argv );

/*
---------------------------------------------------------------------
*/ /*    Print title page header for
```

```
report using IMPPage.     */ /*
---------------------------------------------------------------------
*/
IMPPage (Report, 0);

/* Program code here */

video=VDOpen("VD00"); if(!video){
  puts("Error linking to display!"); return(0);
}

/* Body */

gridlen=rdgrid(&thegrid); /* slurp the standard
     grid intersections */

/* Wait for display to be set up */ puts("Align
layer 1, add points 12 through 20, then hit
return:"); getchar();

/* Just read in the vector segment from the display
   and dump the verticies */

  nvertex =   (int *) HMalloc( sizeof(int) *
  vecinfo.NStructure ); type =     (int *) HMalloc(
  sizeof(int) * vecinfo.NStructure ); group =   (int *)
  HMalloc( sizeof(int) * vecinfo.NStructure ); attribute
  = (int *) HMalloc( sizeof(int) * vecinfo.NStructure
  ); vertices = (double **) HMalloc( sizeof(double
  *)*vecinfo.NStructure ); buffer = (double *) HMalloc(
  sizeof(double) * vecinfo.NVertex * 2 );

  VDVectorInfo(video,VD_READ,1,&vecinfo);
  i=VDVectorIO(&vecinfo,VD_READ,0,vecinfo.NStructure,
   nvertex,type,group,attribute,vertices,buffer,vecinfo.NVertex*2);

  if(!i){ puts("Problems on segment read!"); return(0);
  } printf("Structures: %d\tVertices:
  %d",vecinfo.NStructure,vecinfo.NVertex);

  for(i=0;i<vecinfo.NVertex;i++) printf("#%d - %lf
    %lf\n",i+1,buffer[2*i],buffer[2*i+1]);

/* Finish */

  HFree(nvertex); HFree(type); HFree(group);
  HFree(attribute); HFree(vertices); HFree(buffer);

  /* dumpgrid(thegrid,gridlen);*/ free(thegrid);

VDClose(video);

/*
---------------------------------------------------------------------
*/ /*    Exit program using
IMPReturn.      */ /*
---------------------------------------------------------------------
*/
IMPReturn(); }

int rdgrid(struct grid **array) {
FILE *gridfile; struct grid *vals; int i,len;
float x,y;

gridfile=fopen("gridfile.txt","r"); if(!gridfile){
puts("Couldn't open standard grid defs");
exit(0);
}

len=0; while(!feof(gridfile))
if(fgetc(gridfile)=='\n') len++;
rewind(gridfile);

*array=(struct grid *)calloc(len,sizeof(struct
grid)); vals=*array; if(!(*array)){
puts("Error reserving grid memory");
exit(0);
}

for(i=0;i<len;i++){ fscanf(gridfile,"%f%f",&x,&y);
vals[i].x=x; vals[i].y=y;
}

printf("Read %d grid points from standard
grid.\n\n",len); fclose(gridfile); return(len);
}

void dumpgrid(struct grid *grid,int len) {
FILE *out; int i;
```

```
out=fopen("rectify.gcp","w"); if(!out){
puts("Error writing output!"); return;
}

/* Dump GCPs in x1 y1 x2 y2 format */
for(i=0;i<len;i++)
fprintf(out,"%0.2f %0.2f %0.2f
%0.2f\n",grid[i].i,grid[i].j \
 ,grid[i].x,grid[i].y);

fclose(out); }

:::::::::::::: timenum.c :::::::::::::: #include <time.h>
#include <math.h> #include <stdio.h> #include <ctype.h>
#include <string.h>

main() {
/* decode first line of stdin base 36 min since
70/01/01 */ char code[80]; int i,digit,num;

num=0; code[0]=0; fscanf(stdin,"%s",&code);
if(strlen(code)<12) exit(-1);

for(i=0;i<5;i++){ digit=code[i+2];
if(isalpha(digit))
digit-='a'-10; else
digit-='0';

num+=(digit*pow(36,(4-i))); }

num*=60; /* convert minutes to seconds */
printf("%d",num);
}

:::::::::::::: times.c :::::::::::::: #include <time.h>
#include <math.h> #include <stdio.h> #include <ctype.h>
#include <string.h>

int chop(char *code) {
  int clen,offset,i;

  clen=strlen(code);

  if(clen>12 || clen<5) return(-1); if(clen>5) return(2);
  else return(0);
}

void decode(char *code,int offset) {
  int i,digit,num;

  num=0; for(i=0;i<5;i++){
digit=code[i+offset]; if(isalpha(digit))
  if(isupper(digit)) digit-='A'-10;
  else digit-='a'-10;
else digit-='0';

num+=(digit*pow(36,(4-i))); }

num*=60; /* convert minutes to seconds */
printf("%s\t%s",code,ctime(&num));
}

main() {
  int offset;

/* decode stdin base 36 min since 70/01/01 */
char code[80];

do{ code[0]=0; fscanf(stdin,"%s",&code);
  offset=chop(code);

  if(offset>=0) decode(code,offset); }
while(!feof(stdin));
}

:::::::::::::: to16.c :::::::::::::: /* Read an 8-bit
image and dump a 16-bit */

#include <stdio.h>

main() {
unsigned char in; unsigned short int out;

while(!feof(stdin)){ in=getchar(); out=(unsigned
short int)in; fwrite(&out,sizeof(unsigned
short int),1,stdout);
} }
```

```
:::::::::::::: to_img.c :::::::::::::: #include <stdio.h>

main() {
int i,j;

for(i=0;i<253;i++) for(j=0;j<512;j++)
fprintf(stdout,"%c",getchar()); }

:::::::::::::: to_text.c :::::::::::::: #include <stdio.h>

main() {
int i,j=0;

while(!feof(stdin)){ i=fgetc(stdin);
printf("%d\n",i); j++;
} fprintf(stderr,"Count: %d\n",j);
}

:::::::::::::: txt16.c :::::::::::::: /* Dump an
ImageMagick 16bit gray binary as text */

#include <stdio.h>

main() {
  int cnt; unsigned short in; unsigned char out;

  cnt=0; while(fread(&in,sizeof(unsigned short),1,stdin)){
    out=(unsigned char)in; printf("%d ",out); cnt++;
    if(cnt>19){
      puts("");cnt=0; }
  } }
```

A-4 R/S-Plus Programs

```
::::::::::::: f.accel.R ::::::::::::: "f.accel"_
  function(tab=rep.pres,dist,mult=1,interval=1){
    vel_f.vel(tab,distraw=dist,mult,interval)
    velx_vel$num.hours vely_vel$vel len_length(velx)
    delvel_vely[2:len]-vely[1:(len-1)]
    delint_velx[2:len]-velx[1:(len-1)]

    accel_as.data.frame(cbind(velx[2:len],delvel/delint))
    attr(accel,"names")_c("num.hours","accel")

    return(accel) }

::::::::::::: f.allrep.R ::::::::::::: "f.allrep"_
  function(){ # Call f.report for each case in sequence
    cat("9604\n") f.report("p9604","^v96","9604")
    f.replot(fig=1) cat("\n9701a\n")
    f.report("p9701a","^v970113","9701a") f.replot(fig=2)
    cat("\n9701b\n") f.report("p9701b","^v97012","9701b")
    f.replot(fig=3) cat("\n9711S\n")
    f.report("p9711S","^v9711","9711S") f.replot(fig=4)
    cat("\n9803\n") f.report("p9803","^v9803","9803")
    f.replot(fig=5)

    # Update 4 figure plots with mpage print("Updating
    4-up plots") system("cd report;sh 4/mk4.sh")
  }

::::::::::::: f.bearing.R ::::::::::::: "f.bearing"_
  function(tab=rep.pres,interval=1){
    len_length(tab$easting)
    A_cbind(tab$easting[1:(len-1)],tab$northing[1:(len-1)])
    B_cbind(tab$easting[2:len],tab$northing[2:len])
    Bear_rep(0,len-1)

    del_B-A for(i in 1:(len-1)){
      if(del[i,2]>=0){ # Y is positive (Q1 & Q4)
        Bear[i]_atan(del[i,1]/del[i,2])*360/2/pi
      } else{
        if(del[i,1]>=0){ # X is positive, Y negative
          (Q2) Bear[i]_( ((-1)*atan(del[i,2]/del[i,1]))
          +pi/2)*360/2/pi
        } else{ # X is negative, Y is negative (Q3)
          Bear[i]_( ((-1)*atan(del[i,1]/del[i,2]))
        -pi/2)*360/2/pi }
      } }

    tseq_tab$num.hours[2:len]
#    trange_tseq[len-1]-tseq[1] #
spl_approx(tseq,Bear,xout=seq(tseq[1],tseq[length(tseq)],by=interval))
#
spl_spline(tseq,Bear,n=round(trange/interval)+1,method="natural")

    bearing_as.data.frame(cbind(tseq,Bear))
    attr(bearing,"names")_c("Hour","Bearing")

    return(bearing) }

::::::::::::: f.bomb.R ::::::::::::: "f.bomb"_
  function(){ # correct to baselat latitude (42.5 is
    Roebber's mean value) baselat_60

    hours_rep.pres$num.hours mb_rep.pres$mb
    num_length(rep.pres$mb)

    max.deep_as.data.frame(matrix(rep(0,(5*4)),ncol=5,byrow=T))
    attr(max.deep,"names")_c("Bergeron","reqMBpH","meanlat","delMB","delH")
    num.loop_length(mb)

    for(i in 1:num.loop){ j_i+1 while(j<(num.loop+1)){
        hsep_hours[j]-hours[i] mbsep_mb[i]-mb[j]

        # calculate Bergeron value
        meanlat_mean(c(rep.pres$northing[i],rep.pres$northing[j]))
        ber.hr_sin(2*pi*meanlat/360)/sin(2*pi*baselat/360)
        rate_mbsep/hsep/ber.hr # store maxima if(hsep==6
        && rate>max.deep$Bergeron[1]){
          max.deep$Bergeron[1]_rate
          max.deep$reqMBpH[1]_ber.hr
          max.deep$meanlat[1]_meanlat
          max.deep$delMB[1]_mbsep max.deep$delH[1]_hsep
        } if(hsep==12 && rate>max.deep$Bergeron[2]){
          max.deep$Bergeron[2]_rate
          max.deep$reqMBpH[2]_ber.hr
          max.deep$meanlat[2]_meanlat
          max.deep$delMB[2]_mbsep max.deep$delH[2]_hsep
        } if(hsep==24 && rate>max.deep$Bergeron[3]){
          max.deep$Bergeron[3]_rate
          max.deep$reqMBpH[3]_ber.hr
          max.deep$meanlat[3]_meanlat
          max.deep$delMB[3]_mbsep max.deep$delH[3]_hsep
        }

        j_j+1 }
    }

    # calculate best interpolated pressure drop
    spln_(hours[length(hours)]-hours[1])/6+1
    interp_spline(hours,mb,n=spln,method="natural")
    intervals_length(interp$x)
    sy_spline(1:num,rep.pres$northing,n=intervals,method="natural")
    iy_sy$y for(i in 1:intervals){
      j_i+1 while(j<(intervals+1)){
        hsep_interp$x[j]-interp$x[i]
        mbsep_interp$y[i]-interp$y[j]

        # calculate Bergeron value
        meanlat_mean(c(iy[i],iy[j]))
        ber.hr_sin(2*pi*meanlat/360)/sin(2*pi*baselat/360)
        rate_mbsep/hsep/ber.hr # store maxima if(hsep>=6
        && rate>max.deep$Bergeron[4]){
          max.deep$Bergeron[4]_rate
          max.deep$reqMBpH[4]_ber.hr
          max.deep$meanlat[4]_meanlat
          max.deep$delMB[4]_mbsep
          max.deep$delH[4]_paste(signif(hsep,digits=3),"i",sep="")
        } j_j+1
      } }

# re-order the table for presentation -
    the full table is kept for debugging purposes
    ret_as.data.frame(cbind(max.deep$delH,max.deep$delMB,max.deep$Bergeron))
    # move 6i row to the top ret_rbind(ret[4,],ret[1:3,])
    attr(ret,"names")_c("Time Interval","Delta
    mb","Bergeron") attr(ret,"row.names")_c("","","","")

    return(ret) }

::::::::::::: f.bounds.R ::::::::::::: "f.bounds" <-
function(legx=-175,legy=60){
  # collect storm centres vcent_rep.vecs
  # collect non-grid based vec centres
  vc2_get(paste("VN",basename,sep=""),pos=1)
  vcent_rbind(vcent,vc2)[,6:7]
  pcent_cbind(rep.pres$easting,rep.pres$northing,rep.pres$mb)
  vbnd_c(min(vcent[,1]),max(vcent[,1]),min(vcent[,2]),max(vcent[,2]))
  pbnd_c(min(pcent[,1]),max(pcent[,1]),min(pcent[,2]),max(pcent[,2]))
  all_rbind(vbnd,pbnd)

  plot(x=c(-180,-130),y=c(25,65),type="n",main="Pacific
Storms Locator",xlab="Degrees East",ylab="Degrees North")
# add probability of development area rectanges and lines
  rect(-170,30,-160,50,lty=2)
  lines(x=c(-190,-120),y=c(30,30),lty=2)
  lines(x=c(-190,-120),y=c(50,50),lty=2) points(vcent)
  lines(vcent) points(pcent,pch=19) lines(pcent)
  text(pcent,labels=seq(1:dim(pcent)[1]),pos=4)
  legend(legx,legy,c("Image Centres","Low Pressure
  Centres"),pch=c(1,19))

# add dashed lines connecting vec
  pt. 1 to pres. line at same time
  ptime_rep.pres$num.hours vtime_rep.vecs$num.hours
  plen_length(ptime) vlen_length(vtime)
  sx_splinefun(x=ptime,y=pcent[,1]) # ptime predicts
  p$easting sy_splinefun(x=ptime,y=pcent[,2]) #
  ptime predicts p$northing if(vtime[1]>=ptime[1] &&
  vtime[1]<=ptime[plen]){ # lower range in bounds
    p1_cbind(sx(vtime[1]),sy(vtime[1]))
    l1_rbind(vcent[1,],p1) lines(l1,lty=3,lwd=2)
  } if(vtime[vlen]<=ptime[plen] && vtime[vlen]>=ptime[1]){
  # upper range in bounds
    p2_cbind(sx(vtime[vlen]),sy(vtime[vlen]))
    l2_rbind(vcent[vlen,],p2) lines(l2,lty=3,lwd=2)
  } # add times on the pressure obs
  lbl_paste(seq(1:dim(rep.pres)[1]),"-",rep.pres$hour,"Z",rep.pres$day,sep="")
  text(pcent,labels=lbl,pos=4)
# add times to first and last vector obs
  lbl_paste(rep.vecs$hour,"Z",rep.vecs$day,sep="")
  text(x=vcent[1,1],y=vcent[1,2],labels=lbl[1],pos=3)
  text(x=vcent[vlen,1],y=vcent[vlen,2],labels=lbl[vlen],pos=3)

# place an X at the location of max
  deepening (actually a square encloses
  it) p_get("rep.pres",pos=1) pmin_min(p$mb)
  plogic_p$mb==pmin # T where pcent$mb is at minimum
  eastmin_p$easting[plogic] northmin_p$northing[plogic]
  print(cbind(pmin,eastmin,northmin)) bsize_1.2
```

```
  rect(eastmin-bsize,northmin-bsize,eastmin+bsize,northmin+bsize)
#  text(x=eastmin,y=northmin,labels="X")

  mtext(text=get("casename",pos=1),side=1,outer=T) #
  annotate plot

  frame_data.frame(all[,1],all[,2],all[,3],all[,4])
  attr(frame,"names")_c("min_east","max_east","min_north","max_north")
  return(frame)
}

:::::::::::::: f.casecor.R :::::::::::::: "f.casecor"_
  function(){ tabin_allcaselvl
    cnames_colnames(tabin) len_length(colnames(tabin))
    out_as.data.frame(matrix(4*len*2346,ncol=4))
    # 2346=69 choose 2 combos
    attr(out,"names")_c("Var1","Var2","DF","Correlation")

    count_1 for(i in 1:(len-1)){ # columns
      for(j in (i+1):len){
        corcol_na.omit(as.data.frame(cbind(tabin[,i],tabin[,j])))
        df_dim(corcol)[1]-2 cors_cor(corcol)[2]
        line_c(cnames[i],cnames[j],df,cors)
        out[count,]_line print(paste(count,"of",2346))
        count_count+1
      } }

    return(out) }

:::::::::::::: f.cenplot.R :::::::::::::: "f.cenplot"_
function(delxpos=2.5,ypos=52){ # plot pressure positions
  vcent_cbind(rep.vecs$easting,rep.vecs$northing)
  pcent_cbind(rep.pres$easting,rep.pres$northing)
  all_rbind(vcent,pcent) minx_min(all[,1])
  miny_min(all[,2]) maxx_max(all[,1]) maxy_max(all[,2])
  # scale factor to allow for plot text sfactor_0.05
  bnds_rbind(c(minx+sfactor*minx,miny-sfactor*miny),c(maxx-sfactor*maxx,maxy+sfactor*maxy))

  plot(bnds,type="n",main="Low
  Centres",xlab="Degrees East",ylab="Degrees
  North") points(pcent,pch=19) lines(pcent)
  text(pcent,labels=seq(1:dim(pcent)[1]),pos=4)

# plot vector positions points(vcent) lines(vcent)
# first vector label
  vct1_paste(formatC(rep.vecs$hour[1],width=2,flag="0"),formatC(rep.vecs$minute[1],width=2,flag="0"),"Z",formatC(rep.vecs$day[1],width=2,flag="0"),sep="")
  text(x=rep.vecs$easting[1],y=rep.vecs$northing[1],labels=vct1,pos=4)
# last vector label vtlast_rep.vecs[length(rep.vecs$hour),]
  vct2_paste(formatC(vtlast$hour,width=2,flag="0"),formatC(vtlast$minute,width=2,flag="0"),"Z",formatC(vtlast$day,width=2,flag="0"),sep="")
  text(x=vtlast$easting,y=vtlast$northing,labels=vct2,pos=3)

# Legend legend(bnds[1,1]+delxpos,ypos,c("Image
  Centres","Low Pressure Centres"),pch=c(1,19))

# dashed joins for first and last vec obs
  ptime_rep.pres$num.hours vtime_rep.vecs$num.hours
  plen_length(ptime) vlen_length(vtime)
  sx_splinefun(x=ptime,y=pcent[,1]) # ptime predicts
  p$easting sy_splinefun(x=ptime,y=pcent[,2]) #
  ptime predicts p$northing if(vtime[1]>=ptime[1] &&
  vtime[1]<=ptime[plen]){ # lower range in bounds
    p1_cbind(sx(vtime[1]),sy(vtime[1]))
    l1_rbind(vcent[1,],p1) lines(l1,lty=3,lwd=2)
  } if(vtime[vlen]<=ptime[plen] && vtime[vlen]>=ptime[1]){
  # upper range in bounds
    p2_cbind(sx(vtime[vlen]),sy(vtime[vlen]))
    l2_rbind(vcent[vlen,],p2) lines(l2,lty=3,lwd=2)
  } # add times on the pressure obs
  lbl_paste(seq(1:dim(rep.pres)[1]),"-",rep.pres$hour,"Z",rep.pres$day,sep="")
  text(pcent,labels=lbl,pos=4)
  mtext(text=get("casename",pos=1),side=1,outer=T) #
  annotate plot
}

:::::::::::::: f.combine.R :::::::::::::: # combine first
and second derivative output for bounds calc "f.combine"_
function(tab){
  x1_f.derivative(tab,recurse=1) ymat1_f.derivative(tab)
  x2_f.derivative(tab,order=2,recurse=1)
  ymat2_f.derivative(tab,order=2)

  tmin_min(c(x1,x2)) tmax_max(c(x1,x2))

  c.all_NULL # combine all columns for(i in 1:4){
    c.local_c(ymat1[,i],ymat2[,i]) c.all_c(c.all,c.local)
  }

  rmin_min(c.all) rmax_max(c.all)
  return(cbind(c(tmin,tmax),c(rmin,rmax))) }

:::::::::::::: f.derivative.R :::::::::::::: "f.derivative"
<-
  function(tab,order=1,times=0){ # interpolation on 1/2
    hour interval if(times!=0){
      tseq_tab[,5] tab_tab[,1:4] trange_length(tseq)
    } else{
      tseq_rep.vecs$num.hours trange_round(-2*tseq[1]+1)
    } rate_as.data.frame(matrix(1:((trange-1)*5),ncol=5))

    for(i in 1:4){ # Check for NA's
      if(!complete.cases(t(tab[,i]))){
        for(j in 1:length(tab[,i])){ if(is.na(tab[j,i])){
            tab[j,i]_0 }}}

      spl_spline(tseq,tab[,i],n=trange,method="natural")
      rate[,i]_(spl$y[2:trange]-spl$y[1:(trange-1)])/(spl$x[2:trange]-spl$x[1:(trange-
    } rate[,5]_spl$x[2:length(spl$x)]

    # put in the labels
    attr(rate,"names")_c("Drot","Drlen","Dhlen","Djwid","Time")

    if(order==1){ # return result return(rate)
    } else{
      # recurse until level
      f.derivative(tab=rate,order=order-1,times=1)
    } }

:::::::::::::: f.do2Cor.R :::::::::::::: "f.tmp" <-
  function(Xname,Yname,sep=0,spl=T,fix=F){ # compute
    correlations for variables spline matched using f.dospl

    Y_f.getvar(Yname) if(fix){
      Y[,2]_Y[,2]+90 } if(Xname=="Hours"){ # append Y's
    hours as the predictor
      return(na.omit(Y)) } else{
      X_f.getvar(Xname) }
    Yhat_f.dospl(Y,F,X[,1]+sep,1,2,spl=spl)
    xy_na.omit(as.data.frame(cbind(X[,2],Yhat)))
    attr(xy,"names")_c(colnames(X)[2],colnames(Y)[2])

    f.Varplot(xy,1,2) }

:::::::::::::: f.dospl.R :::::::::::::: "f.dospl"_
  function(intab,hrcol,varcol,tseq,spl=T){
    # handle the interpolations for f.mknorm
    spltab_na.omit(as.data.frame(cbind(intab[,hrcol],intab[,varcol])))
    hours
      hrs_rep(1,length(spltab[,1])) } if(spl){
      spf_splinefun(spltab[,1]/hrs,spltab[,2],method="natural")
    } else{
      spf_approxfun(spltab[,1]/hrs,spltab[,2]) }
    spl_spf(tseq) truth_tseq<=spltab[1,1]/hrs |
         tseq>=spltab[length(spltab[,1]),1]/hrs
    spl[truth]_NA # remove out of time range extrapolations
    return(spl)
  }

:::::::::::::: f.FourUp.R :::::::::::::: "f.FourUp"_
  function(){ # Make sure plots are annotated for 4-up
    printing

    case_switch(get("basename",pos=1), "9604"="Case
           I", "9701a"="Case II", "9701b"="Case III",
           "9711S"="Case IV", "9803"="Case V")

    par(oma=c(2,0,0,0)) # 2 text lines for bottom label
    assign("casename",case,pos=1) # store casename

#    mtext(text=get("casename",pos=1),side=1,outer=T) #
  annotate plot }

:::::::::::::: f.getcyc.R :::::::::::::: "f.getcyc"_
  function(dirname){
    tab_read.table(paste(dirname,"/results",sep=""))[,2:3]
    return(tab)
  }

:::::::::::::: f.getres.R :::::::::::::: "f.getres" <-
  function(dirname,skip=4){ system(paste("tail
    +",skip," ",dirname,"/results >res.tmp",sep=""))
    tab_read.table("res.tmp") system("rm res.tmp")
    return(tab)
  }

:::::::::::::: f.getvar.R :::::::::::::: "f.getvar" <-
  function(var){ # return required variable
```

```
objs_c("Hours","Pressure","Chart.Velocity","Image.Velocity","Chart.Bearing",
       "Image.Bearing","High.Cloud","Cyclonic.Index","Edge.Mean","Edge.StDev","Edge.Max",
       "Jet.Tilt","Head.Length","Comma.Bulge","Baroclinic.Width")
basenms_c("a96","jA","jB","n97","m98") #
names used in highcloud and cyclonic tables
t0s_c(598,320,698,186,152.5) # t0 for each case in
day*24+hour+min/60

i_as.integer(substring(var,nchar(var),nchar(var)))
# trim case name var_substring(var,1,(nchar(var)-1))
# trim case number j_grep(var,objs) # find which var
print(paste("Finding",objs[j],"for case",i))

if(j==2){ # Pressure
  intab_get(paste("P",caselst[i],".rep",sep="")) # must
  "attach()" correct dir first (with the data in it)
  out_cbind(intab[grep("num.hours",colnames(intab))],
              intab[grep("mb",colnames(intab))] ) }

if(j==3){ # Chart.Velocity
  intab_get(paste("P",caselst[i],".rep",sep="")) # must
  "attach()" correct dir first (with the data in it)
  out_f.vel(intab,get(paste("RP",caselst[i],sep=""))[,1])
}

if(j==4){ # Image.Velocity
  intab_rbind(get(paste("V",caselst[i],".rep",sep="")),get(paste("VN",caselst[i],sep="")))
  out_f.vel(intab,get(paste("RV",caselst[i],sep=""))[,1])
}

if(j==5){ # Chart.Bearing
  intab_get(paste("P",caselst[i],".rep",sep="")) # must
  "attach()" correct dir first (with the data in it)
  out_cbind(intab[2:dim(intab)[1],4],get(paste("RP",caselst[i],sep=""))[,2])
}

if(j==6){ # Image.Bearing
  intab_rbind(get(paste("V",caselst[i],".rep",sep="")),get(paste("VN",caselst[i],sep="")))
  out_cbind(intab[2:dim(intab)[1],5],get(paste("RV",caselst[i],sep=""))[,2])
}

if(j==7){ # High.Cloud
  intab_get(paste("c",caselst[i],sep=""))
  out_intab[,1:2]
}

if(j==8){ # Cyclonic.Index
  intab_get(paste("c",caselst[i],sep=""))
  out_cbind(intab[,1],intab[,3])
}

if(j==9){ # Edge.Mean
  intab_get(paste("e",caselst[i],sep=""))
  # proper dir must be "attach()"ed
  out_cbind(intab$day*24+intab$hour+intab$minute/60-t0s[i],intab[,4])
  # Add t0 times
}

if(j==10){ # Edge.StDev
  intab_get(paste("e",caselst[i],sep=""))
  # proper dir must be "attach()"ed
  out_cbind(intab$day*24+intab$hour+intab$minute/60-t0s[i],intab[,5])
  # Add t0 times
}

if(j==11){ # Edge.Max
  intab_get(paste("e",caselst[i],sep=""))
  # proper dir must be "attach()"ed
  out_cbind(intab$day*24+intab$hour+intab$minute/60-t0s[i],intab[,6])
  # Add t0 times
}

if(j==12){ # Jet.Tilt
  out_cbind(get(paste("V",caselst[i],".rep",sep=""))[,5],
             get(paste("G",caselst[i],sep=""))[,1]) }

if(j==13){ # Head.Length
  out_cbind(get(paste("V",caselst[i],".rep",sep=""))[,5],
             get(paste("G",caselst[i],sep=""))[,3]) }

if(j==14){ # Comma.Bulge
  out_cbind(get(paste("V",caselst[i],".rep",sep=""))[,5],
             get(paste("G",caselst[i],sep=""))[,2]) }

if(j==15){ # Baroclinic.Width
  out_cbind(get(paste("V",caselst[i],".rep",sep=""))[,5],
             get(paste("G",caselst[i],sep=""))[,4]) }

out_as.data.frame(out)
```

```
attr(out,"names")_c("Raw.Hours",paste(objs[j],i,sep=""))
return(out)
}

::::::::::::: f.makeplot.R ::::::::::::: "f.makeplot"_
function(Xname,Yname,sep=0,do.ord=F,spl=T){ # each
  object contains Time, and Variable Y_f.getvar(Yname)
  if(Xname=="Hours"){ # append Y's hours as the predictor
    X_as.data.frame(cbind(Y[,1],Y[,1]))
    attr(X,"names")_rep(attr(Y,"names")[1],2)
  } else{
    X_f.getvar(Xname) }

  if(do.ord){ ord_order(X[,2]) # sort order by X's
    variable values X_X[ord,] # sorted X
  } else{
    print("Data will not be ordered") }

  Yhat_F if(dim(X)[1]==dim(Y)[1]){
    if(X[,1]==Y[,1]){ # no need to interpolate!
      Yhat_Y[,2]
    } } if(!Yhat){
    Yhat_f.dospl(Y,F,X[,1]+sep,1,2,spl=spl)
  # call spline/interpolation function }
  xy_na.omit(as.data.frame(cbind(X[,2],Yhat)))
  # combine predictor and response variables
  attr(xy,"names")_c(colnames(X)[2],colnames(Y)[2])
  reg_lm(xy[,2] ~ xy[,1]) plot(xy,type="p") # scatterplot
  of vars lines(x=xy[,1],y=fitted(reg),lty=2) # fit
  regression line

  return(summary(reg)) }

::::::::::::: f.mk2cor.R ::::::::::::: "f.mk2cor" <-
function(sep=0,spl=T){ # produce observation level
  correlation table using the f.getvar paired variable
  method
  ncols_(length(varnames)-1)*5+1 # 5xvars + times ncols_4
  # Cor, DF, Xname, Yname

  tabout_as.data.frame(matrix(rep(0,ncols*2485),ncol=ncols))
  # 71 choose 2 possible values
  attr(tabout,"names")_c("Correlation","D.F.","XName","Yname")

  count_1 print("Time correlations") # do Time
  correlations for(i in 2:length(varnames)){
    for(j in 1:5){ # Cases
      Yname_paste(varnames[i],j,sep="")
      rtab_f.do2Cor("Hours",Yname,sep=sep,spl=spl)
      r_cor(rtab)[2] df_dim(rtab)[1]-2
      tabout[count,]_c(r,df,"Hours",Yname) count_count+1
    } }

  # now correlation within cases, all vars but time
  print("Within-case correlations") for(i in 1:5){ #
  Case-level loop
    print(paste("Case",i))
    casevars_paste(varnames,i,sep="") # Column names of
    Case-n variables for(j in 2:(length(casevars)-1)){
      for(k in (j+1):length(casevars)){
        print(paste("Comparing pair",j,k,"-
        vars",casevars[j],casevars[k]))
        rtab_f.do2Cor(casevars[j],casevars[k],sep=sep,spl=spl)
        if(!dim(rtab[1])){ # No overlap between
        variables!
          r_cor_NA } else{
          r_cor(rtab)[2] } df_dim(rtab)[1]-2
        tabout[count,]_c(r,df,casevars[j],casevars[k])
        count_count+1
      } }
  }

  # finally, correlations between cases on the
  same variable print("Between-case correlations")
  casevars_varnames[2:length(varnames)] # hack to merge
  with old f.mkcor code for(i in 1:length(casevars)){
  # compare same variables for each case
    for(j in 1:4){ for(k in (j+1):5){ # compare varX
      for case-j and case-k
        print(paste("Comparing ",casevars[i],j,"
        and ",casevars[i],k,sep=""))
        rtab_f.do2Cor(paste(casevars[i],j,sep=""),paste(casevars[i],k,sep=""),
                      sep=sep,spl=spl) if(!dim(rtab[1])){
        # No overlap between variables!
          r_cor_NA } else{
          r_cor(rtab)[2] } df_dim(rtab)[1]-2
        tabout[count,]_c(r,df,paste(casevars[i],j,sep=""),
                         paste(casevars[i],k,sep=""))
```

```
        count_count+1
      } }
    }

    return(tabout[1:(count-1),]) }

:::::::::::::: f.mkallplots.R ::::::::::::::

"f.mkallplots"_ function(){
    # do what it says.  First all case levels with
    f.Varplot on tab allcaselvl # then all obs level with
    f.obsreg - only do those pairs that are of interest
    please

    count_1 # Case level pairs pairs_rbind(
    c("EndZone","B24"),
                c("PIV-PII","Pmin"), c("PIV-PII","B24"),
                c("Pmin","Pmean"), c("Pmean","B24"),
                c("CVmax","Pmin"), c("CVmax","B24") )

    for(i in 1:(dim(pairs)[1]) ){
      print(paste(pairs[i,1],"predicts",pairs[i,2]))
      x_grep(pairs[i,1],colnames(caselvl)) # get this var's
      column number y_grep(pairs[i,2],colnames(caselvl))
      # get this var's column number
      print(f.Varplot(caselvl,x,y)) # plot them

      f.psdump(paste("stats/",count,".ps",sep=""))
      count_count+1
    }

    # Time cor Obs level pairs pairs_rbind(
    c("Hours","Chart.Velocity2"),
                c("Hours","Chart.Velocity3"),
                c("Hours","Chart.Velocity5"),
                c("Hours","Image.Bearing1"), #(?)
                c("Hours","Image.Bearing2"),
                c("Hours","Image.Bearing4"),
                c("Hours","Image.Bearing5"),
                c("Hours","Image.Velocity2"),
                c("Hours","Image.Velocity5"),
                c("Hours","Head.Length4"),
                c("Hours","Head.Length5"),
                c("Hours","Comma.Bulge1"),
                c("Hours","Comma.Bulge4"),
                c("Hours","Comma.Bulge5"),
                c("Hours","Jet.Tilt1"),
                c("Hours","Jet.Tilt2"),
                c("Hours","Jet.Tilt3"),
                c("Hours","Jet.Tilt4"),
                c("Hours","Jet.Tilt5"),
                c("Hours","Baroclinic.Width1"),
                c("Hours","Baroclinic.Width3"),
                c("Hours","Baroclinic.Width5"),
                c("Hours","Edge.Mean2"),
                c("Hours","Edge.Mean3"),
                c("Hours","Edge.Mean4"),
                c("Hours","Edge.StDev3"),
                c("Hours","Edge.StDev4") )

    for(i in 1:(dim(pairs)[1]) ){
      print(paste(pairs[i,1],"predicts",pairs[i,2]))
      print(f.obsreg(pairs[i,1],pairs[i,2]))

      f.psdump(paste("stats/",count,".ps",sep=""))
      count_count+1
    }

    # Other obs level pairs pairs_rbind(
    c("Pressure1","Chart.Bearing1"),
                c("Pressure2","Chart.Bearing2"),
                c("Pressure4","Chart.Bearing4"),
                c("Pressure5","Chart.Bearing5"),
                c("Pressure1","Image.Bearing1"),
                c("Pressure4","Image.Bearing4"),
                c("Pressure2","Image.Velocity2"),
                c("Pressure3","Image.Velocity3"),
                c("Pressure5","Image.Velocity5"),
                c("Pressure2","High.Cloud2"),
                c("Pressure4","High.Cloud4"),
                c("Pressure2","Cyclonic.Index2"),
                c("Pressure2","Edge.Mean2"),
                c("Edge.Mean1","Comma.Bulge1"),
                c("Edge.Mean2","Comma.Bulge2"),
                c("Edge.Mean3","Comma.Bulge3"),
                c("Edge.Mean4","Comma.Bulge4"),
                c("Edge.Mean5","Comma.Bulge5"),
                c("Edge.Mean1","Jet.Tilt1"),
                c("Edge.Mean3","Jet.Tilt3"),
                c("Edge.Mean4","Jet.Tilt4"),
                c("Edge.Mean5","Jet.Tilt5"),
                c("Jet.Tilt1","Comma.Bulge1"),
                c("Jet.Tilt3","Comma.Bulge3"),
                c("Jet.Tilt4","Comma.Bulge4"),
                c("Jet.Tilt5","Comma.Bulge5"),
                c("High.Cloud1","Edge.Mean1"),
                c("High.Cloud2","Edge.Mean2"),
                c("High.Cloud3","Edge.Mean3"),
                c("High.Cloud4","Edge.Mean4"),
                c("Cyclonic.Index1","Edge.Mean1"),
                c("Cyclonic.Index3","Edge.Mean3") )

    for(i in 1:(dim(pairs)[1]) ){
      print(paste(pairs[i,1],"predicts",pairs[i,2]))
      print(f.obsreg(pairs[i,1],pairs[i,2]))

      f.psdump(paste("stats/",count,".ps",sep=""))
      count_count+1
    }

    # PII->PIII case level pairs pairs_rbind(
    c("Image.BearingIII.IImax","Pmin"),
                c("Baroclinic.WidthIII.IImin","Pmin"),
                c("Edge.MaxIII.IIdiff","Pmin"),
                c("Image.VelocityIII.IImean","B24"),
                c("Image.BearingIII.IIdiff","B24"),
                c("Image.BearingIII.IImax","B24"),
                c("Jet.TiltIII.IIdiff","B24") )

    for(i in 1:(dim(pairs)[1]) ){
      print(paste(pairs[i,1],"predicts",pairs[i,2]))
      x_grep(pairs[i,1],colnames(allcaselvl))
      # get this var's column number
      y_grep(pairs[i,2],colnames(allcaselvl)) # get this
      var's column number print(f.Varplot(allcaselvl,x,y))
      # plot them

      f.psdump(paste("stats/",count,".ps",sep=""))
      count_count+1
    }

    # forgotten variables (Let's not have to re-order
    all eqn's in the text!  # Time cor Obs level pairs
    pairs_rbind( c("Hours","Chart.Bearing1"),
                c("Hours","Chart.Bearing2"),
                c("Hours","Chart.Bearing4"),
                c("Hours","Chart.Bearing5"),
                c("Hours","Pressure1"),
                c("Hours","Pressure2"),
                c("Hours","Pressure4"),
                c("Hours","Pressure5") )

    for(i in 1:(dim(pairs)[1]) ){
      print(paste(pairs[i,1],"predicts",pairs[i,2]))
      print(f.obsreg(pairs[i,1],pairs[i,2]))

      f.psdump(paste("stats/",count,".ps",sep=""))
      count_count+1
    }

}

:::::::::::::: f.mkcor.R :::::::::::::: "f.mkcor"_
  function(tabin){ # calculate na.omit correlations
    vars_dim(tabin)[2] ncols_6 cnames_colnames(tabin)
    tabout_as.data.frame(matrix(rep(0,ncols*2485),ncol=ncols))
    # 71 choose 2 possible values
    attr(tabout,"names")_c("Correlation","D.F.","XName","Yname","Row","Col")

    # fill first row manually
    cortab_na.omit(cbind(tabin[,1],tabin[,2]))
    tabout[1,]_c(cor(cortab)[1,2],dim(cortab)[1]-2,
                cnames[1],cnames[2],1,2)

    # do time correlations first count_2 print("Time
    correlations") for(i in 3:vars){ # skip time column
    and first column (already done)
      print(paste(i,"in",vars))
      cortab_na.omit(cbind(tabin[,1],tabin[,i]))
      line_c(cor(cortab)[1,2],dim(cortab)[1]-2,cnames[1],cnames[i],1,i)
      for(j in 1:ncols){
        tabout[count,j]_line[j] } count_count+1
    }

    # now correlation within cases, all vars but time
    print("Within-case correlations") for(i in 1:5){ #
    Case-level loop
      print(paste("Case",i))
```

```
    casevars_grep(as.character(i),cnames) #
    Column indices of Case-n variables for(j in
    1:(length(casevars)-1)){
      for(k in (j+1):length(casevars)){
        print(paste("Comparing pair",j,k,"-
        vars",cnames[casevars][j]
                  ,cnames[casevars][k]))
        cortab_na.omit(cbind(tabin[,casevars[j]],tabin[,casevars[k]]))
        line_c(cor(cortab)[1,2],dim(cortab)[1]-2,
              cnames[casevars][j],cnames[casevars][k],i,j)
        for(l in 1:ncols){
          tabout[count,l]_line[l] } count_count+1
      } }
  }

  # finally, correlations between cases
  on the same variable print("Between-case
  correlations") casevars_cnames[grep("1",cnames)]
  # find all variable names for case 1
  casevars_substring(casevars,1,nchar(casevars)-1)
  # remove case number from variable names for(i in
  1:length(casevars)){ # compare same variables for
  each case
    for(j in 1:4){ for(k in (j+1):5){ # compare varX
      for case-j and case-k
        print(paste("Comparing ",casevars[i],j,"
        and ",casevars[i],k,sep=""))
        jcol_grep(paste(casevars[i],j,sep=""),cnames)
        kcol_grep(paste(casevars[i],k,sep=""),cnames)
        cortab_na.omit(cbind(tabin[,jcol],tabin[,kcol]))
        line_c(cor(cortab)[1,2],dim(cortab)[1]-2,
            paste(casevars[i],j,sep=""),paste(casevars[i],k,sep=""),jcol,kcol)
        for(l in 1:ncols){
          tabout[count,l]_line[l] } count_count+1
      } }
  }

  return(tabout[1:count-1,]) }

:::::::::::::: f.mknorm.R :::::::::::::: "f.mknorm" <-
 function(normal=T,spl=T){ # create a normalised
   "global" table for each case lst_get("caselst",pos=1)
   objs_c("Normalised.Hours","Pressure","Chart.Velocity","Image.Velocity","Chart.Bearing",
         "Image.Bearing","High.Cloud","Cyclonic.Index","Edge.Mean","Edge.StDev","Edge.Max",
         "Jet.Tilt","Head.Length","Comma.Bulge","Baroclinic.Width")
   t0s_c(598,320,698,186,152.5) # t0
   for each case in day*24+hour+min/60
   basenms_c("a96","jA","jB","n97","m98") # names used
   in highcloud and cyclonic tables

   if(normal){ hrs_c(24,21.5,25.5,21.5,26.2) #
     time normalisation: time of PhaseIV - tPhaseII
     tseq_seq(-1,2.1,0.01) print("Calculating tPIV-tPII
     normalised table")
   } else{ # use raw time values
     hrs_c(1,1,1,1,1) # use these for non-normalised data
     values tseq_seq(-26,55,0.5) # approximate time range
     for all cases objs[1]_"Raw.Hours" # correct column
     name print("Calculating raw-time based table")
   }

   labs_rep(0,(length(objs)-1)*5) # leave off time column
   for now for(i in 1:length(labs)){
     # add index number to column names
     labs[i]_paste(objs[as.integer((i-1)/5+2)],(i-1)%%5+1,sep="")
   } # now add the time column labs_c(objs[1],labs)

   rows_length(tseq) cols_length(labs)
   tab_as.data.frame(matrix(rep(0,rows*cols),ncol=cols))
   attr(tab,"names")_labs tab[,1]_tseq

   for(i in 1:5){ print(paste("Filling
     Case",i,"variables")) # Fill vars for each
     case # ======================= # Pressure
     col_grep(paste("Pressure",i,sep=""),labs)
     intab_get(paste("P",caselst[i],".rep",sep="")) # must
     "attach()" correct dir first (with the data in it)
     tab[,col]_f.dospl(intab,hrs[i],tseq,grep("num.hours",colnames(intab))
                     ,grep("mb",colnames(intab)),spl=spl)

     # Chart.Velocity
     col_grep(paste("Chart.Velocity",i,sep=""),labs)
     tab[,col]_f.dospl(f.vel(intab,get(paste("RP",caselst[i],sep=""))[,1]),
                     hrs[i],tseq,1,2,spl=spl)

     # Chart.Bearing
     col_grep(paste("Chart.Bearing",i,sep=""),labs)
     tab[,col]_f.dospl(cbind(intab[2:length(intab[,5]),],
                         get(paste("RP",caselst[i],sep=""))[,2]),hrs[i],tseq,4,8

     # Image.Velocity
     col_grep(paste("Image.Velocity",i,sep=""),labs)
     intab_rbind(get(paste("V",caselst[i],".rep",sep="")),get(paste("VN",caselst[i],
     tab[,col]_f.dospl(f.vel(intab,get(paste("RV",caselst[i],sep=""))[,1]),
                     hrs[i],tseq,1,2,spl=spl)

     # Image.Bearing
     col_grep(paste("Image.Bearing",i,sep=""),labs)
     tab[,col]_f.dospl(cbind(intab[2:length(intab[,5]),],
                         get(paste("RV",caselst[i],sep=""))[,2]),hrs[i],tseq,5,8

     # High.Cloud
     col_grep(paste("High.Cloud",i,sep=""),labs) if(i>1){
     # 9604 case read separately (imagefile names don't
     code hours)
       system(paste("i/addt0.pl",t0s[i],basenms[i],">ctab"))
       intab_read.table("ctab")
     } else{
       intab_get("c9604",pos=1) }
     tab[,col]_f.dospl(intab,hrs[i],tseq,1,2,spl=spl)

     # Cyclonic.Index
     col_grep(paste("Cyclonic.Index",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,1,3,spl=spl)

     # Edge.Mean
     col_grep(paste("Edge.Mean",i,sep=""),labs)
     intab_get(paste("e",lst[i],sep=""))
     # proper dir must be "attach()"ed
     intab_cbind(intab$day*24+intab$hour+intab$minute/60-t0s[i],
               intab[,4:6]) # Add t0 times
     tab[,col]_f.dospl(intab,hrs[i],tseq,1,2,spl=spl)

     # Edge.StDev
     col_grep(paste("Edge.StDev",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,1,3,spl=spl)

     # Edge.Max col_grep(paste("Edge.Max",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,1,4,spl=spl)

     # Jet.Tilt col_grep(paste("Jet.Tilt",i,sep=""),labs)
     intab_cbind(get(paste("V",lst[i],".rep",sep="")),
               get(paste("G",lst[i],sep="")) )
     tab[,col]_f.dospl(intab,hrs[i],tseq,5,8,spl=spl)

     # Head.Length
     col_grep(paste("Head.Length",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,5,10,spl=spl)

     # Comma.Bulge
     col_grep(paste("Comma.Bulge",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,5,9,spl=spl)

     # Baroclinic.Width
     col_grep(paste("Baroclinic.Width",i,sep=""),labs)
     tab[,col]_f.dospl(intab,hrs[i],tseq,5,11,spl=spl)

   }

   return(tab) }

:::::::::::::: f.obsreg.R :::::::::::::: "f.obsreg"_
function(xvar,yvar,do.ord=F){
spl_f.do2Cor(xvar,yvar)
f.Varplot(spl,1,2,do.ord=do.ord)
}

:::::::::::::: f.order.R :::::::::::::: "f.order"_
  function(tab,na.remove=T,minDF=F,sortcol=1,DFcol=2,largefirst=T){
   if(na.remove){
     isvar_!is.na(as.numeric(splcor[,1])) t1_tab[isvar,]
   } else{
     t1_tab }

   if(minDF){ # remove less than DF deg. of freedom rows
     t1_t1[as.numeric(t1[,DFcol])>=minDF,]

   t2_t1[order(t1[,sortcol]),] # ordered smallest to
   largest if(largefirst){
     return(t2[dim(t2)[1]:1,]) # flip sort order } else{
   return(t2) }
  }

:::::::::::::: f.pderiv.R :::::::::::::: "f.pderiv"_
  function(tab=rep.pres,order=1){ # produce 1st and
```

```
    2nd derivative in hourly intervals for pressures
    mb_tab$mb plen_length(mb) tseq_tab$num.hours
    trange_tseq[plen]-tseq[2]+1 # we lose the first obs in
    taking diff of [2]-[1] delmb_mb[2:plen]-mb[1:(plen-1)]
    deltime_tseq[2:plen]-tseq[1:(plen-1)]
    dermb_delmb/deltime

    spl_spline(tseq[2:plen],dermb,n=trange,method="natural")
    dmb_as.data.frame(spl)
    attr(dmb,"names")_c("num.hours","mb") if(order==1){
      return(dmb) } else{
      f.pderiv(tab=dmb,order=order-1) }
  }

:::::::::::::: f.Pdiff.R :::::::::::::: "f.Pdiff"_
  function(){ # find the Phase III-II difference between
    vars PIIs_c(-10,-7,-9.5,-4.5,-6.2)

# define the output matrix:  cols Case1 .. Case
5 # rows:  <varX>III.II{diff|max|mean|min}
# 5 cases, 4 treatments, 14 variables
results_as.data.frame(matrix(5*4*14,ncol=5))

    for(i in 1:5){ # case level count_1
      for(j in 2:length(varnames)){
vname_paste(varnames[j],i,sep="")
var_f.getvar(vname) vtime_var[,1]
if(!length(vtime[vtime==PIIs[i]])){ # No obs at PII
  vtime_c(vtime,PIIs[i]) # tack it on the end }
if(!length(vtime[vtime==0])){ # No t0 obs
  vtime_c(vtime,0) } vtime_vtime[order(vtime)] #
sort the new entries vspl_f.dospl(var,F,vtime,1,2)
vout_na.omit(cbind(vtime,vspl))

vtest_var for(k in 1:dim(vout)[1]){
  if(!dim(var[ (var[,1]==vout[k,1]), ])[1]){
    # spline entries not found in original var
    vtest_rbind(vtest,vout[k,]) # append entry to
    a copy
  } } vout_vtest[order(vtest[,1]),] # and sort
everything

# fill the matrix rngnums_1:dim(vout)[1] # get
number of rows rngnums_rngnums[vout[,1]>=PIIs[i]
& vout[,1]<=0] rngvar_vout[rngnums,]

results[count,i]_rngvar[dim(rngvar)[1],2]-
 rngvar[1,2] # difference PIII.II
results[count+1,i]_max(rngvar[,2])
results[count+2,i]_mean(rngvar[,2])
results[count+3,i]_min(rngvar[,2])

if(i==1){ # fill row.names the first time
resbase_paste(varnames[j],"III.II",sep="")
attr(results,"row.names")[count]_paste(resbase,"diff",sep="")
attr(results,"row.names")[count+1]_paste(resbase,"max",sep="")
attr(results,"row.names")[count+2]_paste(resbase,"mean",sep="")
attr(results,"row.names")[count+3]_paste(resbase,"min",sep="")
}

count_count+4 }
      }

return(results) }

:::::::::::::: f.plot2geom.R :::::::::::::: # See
f.plottabs func # Plot first and second geom derivatives
"f.plot2geom"_
  function(geom){ # collect rates d1_f.derivative(geom)
    d2_f.derivative(geom,order=2)

    # increase outer margin to 3 chars par(oma=c(2,0,3,0))
    # four plots: Rot, Rlen, Hlen, Jwid par(mfcol=c(2,2))
#     titles_c("Rotation","Radial Length","Head Length","Jet
    Width") titles_c("Jet Tilt from N-E Rates","Comma
    \"Bulge\" Radius Rates","Comma Head Length Rates"
     ,"Baroclinic Cloud Width Rates")

    for(i in 1:4){ # set up scaling D_c(d1[,i],d2[,i])
      T_c(d1[,5],d2[,5]) Dmin_min(D) Dmax_max(D) if(i>1){
yl_"Corrected Kilometres/{Hr|Hr^2}" } else{
yl_"Degrees of Arc/{Hr|Hr^2}" }
      plot(cbind(c(min(T),max(T)),c(Dmin,Dmax)),type="n",main=titles[i],xlab="
      hours from t0",ylab=yl)
      lines(x=c(min(T),max(T)),y=c(0,0))
      lines(x=d1[,5],y=d1[,i],type="o")
      lines(x=d2[,5],y=d2[,i],type="b",lty=2)
    }
```

```
    mtext(text=get("casename",pos=1),side=1,outer=T)
    # annotate plot # restore original parameters
    par(mfcol=c(1,1)) par(oma=c(2,0,0,0))

  }

:::::::::::::: f.plot2pres.R :::::::::::::: "f.plot2pres"_
  function(delpos=0){ par(mfcol=c(1,2))

    if(delpos==0){ f.presplot() f.plotpderiv()
    } else{
      f.presplot(delx=delpos[1],dely=delpos[2])
      f.plotpderiv(legx=delpos[3],legy=delpos[4])
    }

    mtext(text=get("casename",pos=1),side=1,outer=T) #
    annotate plot par(mfcol=c(1,1))
  }


:::::::::::::: f.plot4VB.R :::::::::::::: "f.plot4VB"_
  function(legpos=0,FourUp=NULL){
    basename_get("basename",pos=1) par(mfrow=c(2,2))

    f.FourUp()

    Pvelbear_get(paste("RP",basename,sep=""),pos=1)
    Vvelbear_get(paste("RV",basename,sep=""),pos=1)
    vecs_rbind(get(paste("V",basename,".rep",sep=""),pos=1),
      get(paste("VN",basename,sep=""),pos=1))
    xbnd_range(c(rep.pres$num.hours,rep.vecs$num.hours))

    Pvel_f.vel(distraw=Pvelbear[,1])
    Vvel_f.vel(tab=vecs,distraw=Vvelbear[,1])
    Paccel_f.accel(dist=Pvelbear[,1])
    Vaccel_f.accel(tab=vecs,dist=Vvelbear[,1])
    ybvel_range(c(Pvel[,2],Vvel[,2],Paccel[,2],Vaccel[,2]))
    ybbear_range(c(Pvelbear[,2],Vvelbear[,2]))

    velbnd_cbind(xbnd,ybvel) bearbnd_cbind(xbnd,ybbear)

    if(legpos==0){
      f.plotvel(dist=Pvelbear[,1],bnds=velbnd,sub="1")
      f.plotvel(tab=vecs,dist=Vvelbear[,1],bnds=velbnd,interval=0.5,
chart="Images",sub="2") } else{
      f.plotvel(dist=Pvelbear[,1],bnds=velbnd,legx=legpos[1],
legy=legpos[2],sub="1")
      f.plotvel(tab=vecs,dist=Vvelbear[,1],bnds=velbnd,interval=0.5,
legx=legpos[3],legy=legpos[4],chart="Images",sub="2")
    }

    f.plotbearing(Bear=Pvelbear[,2],bnds=bearbnd,sub="3")
    f.plotbearing(Bear=Vvelbear[,2],bnds=bearbnd,
  tab=vecs,interval=0.5,chart="Images",sub="4")

    mtext(text=get("casename",pos=1),side=1,outer=T) #
    annotate plot par(mfrow=c(1,1))
  }

:::::::::::::: f.plotbearing.R ::::::::::::::
"f.plotbearing"_
  function(tab=rep.pres,Bear,bnds=0,legx=-14,legy=0.5,interval=1,chart="Charts",sub="
#     B_f.bearing(tab,interval)
    hrs_tab$num.hours len_length(hrs) tseq_hrs[2:len]
    B_as.data.frame(matrix(c(tseq,Bear),ncol=2))
    attr(B,"names")_c("Hour","Bearing") # Interpolation
#     tseq_B$Hour Bear_B$Bearing
    trange_tseq[length(tseq)]-tseq[1]
    spl_spline(tseq,Bear,n=round(trange/interval)+1,method="natural")

    title_paste("Ground Track Bearing - ",chart,sep="")
    if(bnds==0){
      bnds_spl } plot(bnds,main=title,xlab="Hours
    from t0",ylab="Degrees Clockwise from
    North",type="n",sub=sub) points(B) lines(spl)
  }

:::::::::::::: f.plotedge.R :::::::::::::: "f.plotedge"_
  function(legx1=0,legx2=0){ basename_get("basename",pos=1)
    obj_get(paste("e",basename,sep=""),pos=1)

    yrng_range(obj[,c(4,5)])
    hrs_obj$day*24+obj$hour+obj$minute/60-get("t0",pos=1)
    xrng_range(hrs)
    ext_cbind(c(xrng[1],xrng[2]+(xrng[2]-xrng[1])*0.4),yrng)
    par(oma=c(2,0,3,0)) par(mfrow=c(2,1))

    # mean, st.dev and highest val
    plot(ext,type="n",main="Edge Sharpening",xlab="Hours
```

```
    from t0",ylab="Sobel Gradient Magnitude")
    points(x=hrs,y=obj$mean,type="b",pch=19)
    points(x=hrs,y=obj$stdev,type="b",pch=22)
    points(x=hrs,y=obj$stdev,type="b",pch=22)
    legend(legx1,yrng[1]+(yrng[2]-yrng[1])*0.75,c("Mean","St.Dev"),pch=c(19,22))

    yrng_range(obj$max)
    plot(x=ext[,1],yrng,type="n",main="Edge
    Sharpening",xlab="Hours from t0",ylab="Sobel Gradient
    Magnitude") points(x=hrs,y=obj$max,type="b",pch=23)
    legend(legx2,yrng[1]+(yrng[2]-yrng[1])*0.65,"Image
    Max.",pch=23)
    mtext(text=get("casename",pos=1),side=1,outer=T) #
    annotate plot

    par(mfrow=c(1,1)) par(oma=c(2,0,0,0))
  }

:::::::::::::: f.plotgeom.R :::::::::::::: # Plot raw
geometric values "f.plotgeom" <- function(geom){
    # increase outer margin to 3 chars par(oma=c(2,0,3,0))

    # four plots: Rot, Rlen, Hlen, Jwid par(mfcol=c(2,2))
#    titles_c("Rotation","Radial Length","Head Length","Jet
    Width") titles_c("Jet Tilt from N-E","Comma \"Bulge\"
    Radius","Comma Head Length","Baroclinic Cloud Width")

    for(i in 1:4){ # Check for NA's
      if(!complete.cases(t(geom[,i]))){
for(j in 1:length(geom[,i])){ if(is.na(geom[j,i])){
    geom[j,i]_0 }}}

      pl_cbind(rep.vecs$num.hours,geom[,i])
      nm_round(-2*rep.vecs$num.hours[1]+1)
      sp_spline(pl,n=nm,method="natural") if(i>1){
#   yl_"Geographic Degrees" yl_"Kilometres"
      } else{
yl_"Degrees of Arc" }
      plot(sp,type="n",main=titles[i],xlab="Interpolated
      hours from t0",ylab=yl) points(pl) lines(sp)
    }

    mtext(text=get("casename",pos=1),side=1,outer=T)
    # annotate plot # restore original parameters
    par(mfcol=c(1,1)) par(oma=c(2,0,0,0))

  }

:::::::::::::: f.plotpderiv.R :::::::::::::: # Plot first
and second pressure derivatives "f.plotpderiv"_
  function(legx=-14,legy=-1.5){ # collect rates
    d1_f.pderiv() d2_f.pderiv(order=2)

    # set up scaling P_c(d1$mb,d2$mb)
    T_c(d1$num.hours,d2$num.hours)
    plot(cbind(c(min(T),max(T)),c(min(P),max(P))),type="n",main="1st
    and 2nd Derivatives",xlab="Interpolated hours
    from t0",ylab="Millibars/{Hr|Hr^2}",sub="2")
    lines(x=d1$num.hours,y=d1$mb,type="o")
    lines(x=d2$num.hours,y=d2$mb,type="b",lty=2,pch=2)

    legend(legx,legy,c("1st Deriv.","2nd
  Deriv."),pch=c(1,2)) }

:::::::::::::: f.plotrad.R :::::::::::::: "f.plotrad"_
  function(basename){
    hcloud_get(paste("c",basename,sep=""),pos=1)
    edges_get(paste("e",basename,sep=""),pos=1)
    hours_edges[,1]*24+edges[,2]+edges[,3]/60-get("t0",pos=1)
    par(mfcol=c(1,2))
#    split.screen(c(1,2)) #    screen(1)
    plot(x=hours,y=hcloud[,1],type="b",main="High Cloud
Pixel Count",xlab="Hours from t0",ylab="Number of Pixels")
#    screen(2)
    plot(x=hours,y=hcloud[,2],type="b",main="Cyclonic
    Index",xlab="Hours from t0",ylab="Index Value")

    mtext(text=get("casename",pos=1),side=1,outer=T) #
annotate plot #    close.screen(all=T)
    par(mfcol=c(1,1)) }

:::::::::::::: f.plotres.R :::::::::::::: "f.plotres"_
function(resfile){
  close.screen() plots_dim(resfile[2])
  split.screen(c(ceiling(sqrt(plots)),floor(sqrt(plots))))
  for(i in 1:plots){
    screen(i) plot(resfile[,i],type="o")
  } close.screen()
}
```

```
:::::::::::::: f.plottabs.R :::::::::::::: # plot output
from derivatives of tables "f.plottabs"_ function(tab){
# plot tables and data ranges
  x1_f.derivative(tab,recurse=1) ymat1_f.derivative(tab)
  x2_f.derivative(tab,order=2,recurse=1)
  ymat2_f.derivative(tab,order=2)

  tmin_min(c(x1,x2)) tmax_max(c(x1,x2))

  c.all_NULL # combine all columns for(i in 1:4){
    c.local_c(ymat1[,i],ymat2[,i]) c.all_c(c.all,c.local)
  }

  rmin_min(c.all) rmax_max(c.all)
  plot(cbind(c(tmin,tmax),c(rmin,rmax)),type="n")

  pntsym_c(19,22,23,24) for(i in 1:4){
    # use different line types par(lty=i) # first deriv.
    points(x=x1,y=ymat[,i],pch=22) lines(x=x1,y=ymat[,i])
    # second deriv.  points(x=x2,y=ymat2[,i],pch=23)
  }

}

:::::::::::::: f.plotvel.R :::::::::::::: "f.plotvel"_
  function(tab=rep.pres,dist,bnds=0,mult=1,legx=-14,legy=0.5,interval=1,chart="Charts'
    vel_f.vel(tab,dist,mult,interval)
    accel_f.accel(tab,dist,mult,interval)

    # scaling if(bnds==0){
      D_c(vel$vel,accel$accel)
      T_c(vel$num.hours,accel$num.hours)
      bnd_cbind(c(min(T),max(T)),c(min(D),max(D)))
    } else{
      bnd_bnds } title_paste("Velocity
    and Acceleration - ",chart)
    plot(bnd,type="n",main=title,xlab="Interpolated hours
    from t0",ylab="Corrected Kilometres/{Hr|Hr^2}",sub=sub)
    lines(x=vel$num.hours,y=vel$vel,type="o")
    lines(x=accel$num.hours,y=accel$accel,type="b",lty=2,pch=2)

#
    legend(legx,legy,c("Velocity","Acceleration"),pch=c(1,2))
    legend(legx,legy,c("V.","A."),pch=c(1,2))
  }

:::::::::::::: f.presplot.R :::::::::::::: "f.presplot"
<- function(delx=0.2,dely=2){ # Pressure centres
  ptime_rep.pres$num.hours pval_rep.pres$mb

# plot pressure
  prng_c(min(ptime),max(ptime),min(pval),max(pval))
  pbnd_matrix(c((prng[1]-delx*.2*abs(prng[1])),(prng[2]+delx*abs(prng[2])),(prng[3]-de
  plot(pbnd,type="n",main="Pressure
  Observations",xlab="Hours from t0",ylab="Pressure
  (mb)",sub="1") points(ptime,pval)
  lines(spline(ptime,pval,method="natural"))
  lbl_paste(seq(1:dim(rep.pres)[1]),"-",rep.pres$hour,"Z",rep.pres$day,sep="")
  text(ptime,pval,labels=lbl,pos=4)
}

:::::::::::::: f.psdump.R :::::::::::::: "f.psdump"_
function(filename){
  dev.print(device=postscript,paper="letter",horizontal=F)
  system(paste("mv Rplots.ps",filename))
}

:::::::::::::: f.Ptest.R :::::::::::::: "f.Pdiff"_
  function(){ # find the Phase III-II difference between
    vars PIIs_c(-10,-7,-9.5,-4.5,-6.2)

# define the output matrix:  cols Case1 .. Case
5 # rows:  <varX>III.II{diff|max|mean|min}
# 5 cases, 4 treatments, 14 variables
results_as.data.frame(matrix(5*4*14,ncol=5))

    for(i in 1:5){ # case level count_1
      for(j in 2:length(varnames)){
vname_paste(varnames[j],i,sep="")
var_f.getvar(vname) vtime_var[,1]
vspl_f.dospl(var,F,vtime,1,2)
#  vlogic_vtime>=PIIs[i] &
vtime<=0 # remove out of bounds values #
vout_cbind(vtime[vlogic],vspl[vlogic]) #HERE
vout_cbind(vtime,vspl)

# fill the matrix rngnums_1:dim(vout)[1] # get
```

```
number of rows rngnums_rngnums[vout[,1]>=PIIs[i]
& vout[,1]<=0] rngvar_vout[rngnums,]

results[count,i]_rngvar[dim(rngvar)[1],2]-
 rngvar[1,2] # difference PIII.II
results[count+1,i]_max(rngvar[,2])
results[count+2,i]_mean(rngvar[,2])
results[count+3,i]_min(rngvar[,2])

if(i==1){ # fill row.names the first time
resbase_paste(varnames[j],"III.II",sep="")
attr(results,"row.names")[count]_paste(resbase,"diff",sep="")
attr(results,"row.names")[count+1]_paste(resbase,"max",sep="")
attr(results,"row.names")[count+2]_paste(resbase,"mean",sep="")
attr(results,"row.names")[count+3]_paste(resbase,"min",sep="")
}

count_count+4 }
      }

# return(results) }

::::::::::::: f.readist.R ::::::::::::: "f.readist"_
  function(rangefile="cenout"){ tab_read.table(rangefile)
    attr(tab,"names")_c("Range","Bearing") return(tab)
  }

::::::::::::: f.replot.R ::::::::::::: f.replot_
  function(args=0, fig=0){ if(fig>0){ # replot pre-defined
    selection
      switch(fig,
      f.replot(args=c(-175,60,23,21,1.5,1,-12,-1.7,-20,240,-21,100)),
      f.replot(args=c(-175,60,32,30,0.7,1,10,-1.8,-12,-60,10,-60)),
      f.replot(args=c(-175,60,13,12,1,1,2,-0.5,-7,100,18,100)),
      f.replot(args=c(-152,65,47,44,0.3,1,29,-1.5,1,75,20,125)),
      f.replot(args=c(-175,60,65,60,0.3,0.1,26,-1.5,1,-40,15,-40))
     ) return()
    }

    if(length(args)!=12){ print("Requires 12 format
      arguments"); return();
    }

    f.bounds(legx=args[1],legy=args[2]);
    f.psdump(paste("report/",basename,"/bounds.ps",sep=""))
    f.plotedge(legx1=args[3],legx2=args[4]);
    f.psdump(paste("report/",basename,"/edges.ps",sep=""))
    f.plot2pres(delpos=args[5:8]);
    f.psdump(paste("report/",basename,"/pressure.ps",sep=""))
    f.plot4VB(legpos=args[9:12]);
    f.psdump(paste("report/",basename,"/velbear.ps",sep=""))
  }

::::::::::::: f.report.R ::::::::::::: "f.report"_
  function(spres,vecpat,basename){ # report
    on pressure centres pres_get(spres,pos=1)
    pyear_1900+as.numeric(substr(spres,2,3))
    pdates_pres[,1] phour_pres[,2]
    phours_(pres[,1]-min(pres[,1]))*24+pres[,2]
    pmb_pres[,3]+900 pmb[pmb<930]_pmb[pmb<930]+100 #
    pres. >= 1000mb peast_pres[,4]*(-1) pnorth_pres[,5]
    pframe_data.frame(pyear,pdates,phour,phours,pmb,peast,pnorth)
    attr(pframe,"names")_c("year","day","hour","num.hours","mb","easting","northing")

    # collect data on vector centres
    vdates_f.vdates(vecpat) vcentres_f.vcentres(vecpat)
    vframe_cbind(vdates,vcentres) cat("The last vector
    entry is:\n") print(vframe[dim(vframe)[1],])
#    t0line_as.integer(readline("Enter
    t0 entry number: ")) t0line_dim(vframe)[1]
    t0_vframe$day[t0line]*24+vframe$hour[t0line]+(vframe$minute[t0line]/60)
    assign("t0",t0,pos=1) #
    save t0 as obj for other functions
    vframe$num.hours_vframe$day*24+vframe$hour+(vframe$minute/60)-t0
    pframe$num.hours_pframe$day*24+pframe$hour-t0

    # ask user about storage name #
basename_readline("Enter case's base id: ")

    # create new objs with results
    assign(paste("P",basename,".rep",sep=""),pframe,pos=1)
    assign(paste("V",basename,".rep",sep=""),vframe,pos=1)

    # store vanila copy for other functions
    to use assign("rep.pres",pframe,pos=1)
    assign("rep.vecs",vframe,pos=1)

    # store basename assign("basename",basename,pos=1)

    # set plot params and casename f.FourUp()

    # create base report directory system(paste("mkdir
    report/",basename,sep=""))

    # calls to reporting functions

    print("Dates") D_matrix(1:20,ncol=4)
    D[,1]_c(as.integer(substr(spres,2,3)),as.integer(substr(spres,4,5)),pframe[1,2],p
    D[,2]_c(D[1:2,1],pframe[length(pframe[,1]),2],pframe[length(pframe[,1]),3],0)
    vlst_ls(patt=vecpat,pos=1)
    D[,3]_c(as.integer(substr(vlst[1],2,3)),vframe[1,1],vframe[1,2],vframe[1,3],vfram
    vlen_length(vlst)
    D[,4]_c(as.integer(substr(vlst[vlen],2,3)),vframe[vlen,1],vframe[vlen,2],vframe[v
    Dout_as.data.frame(D) attr(Dout,"names")_c("First
    Chart","Last Chart","First Image","Last Image")
    attr(Dout,"row.names")_c("Year","Month","Day","Hour","Minute")
    assign(paste("D",basename,sep=""),Dout,pos=1)

    print("Bounds") bounds_f.bounds()
    f.psdump(paste("report/",basename,"/bounds.ps",sep=""))

    print("Pressure") f.plot2pres()

    f.psdump(paste("report/",basename,"/pressure.ps",sep=""))

    print("Velocity and Bearing") f.plot4VB()
    f.psdump(paste("report/",basename,"/velbear.ps",sep=""))

    print("Centres") f.cenplot()
    f.psdump(paste("report/",basename,"/centres.ps",sep=""))

    print("Bergeron") deepening_f.bomb()
    assign(paste("B",basename,sep=""),deepening,pos=1)

    print("Geometry") #    geom_f.table(vecpat)
    geom_f.table(paste("^d",substring(vecpat,2),sep=""))
    # use vecs projected into km distances
    assign(paste("G",basename,sep=""),geom,pos=1)

    print("Geometry Plots") # raw values f.plotgeom(geom)
    f.psdump(paste("report/",basename,"/geom1.ps",sep=""))
    # 1st and 2nd derivatives f.plot2geom(geom)
    f.psdump(paste("report/",basename,"/geom2.ps",sep=""))

    print("Radiometric") f.plotrad(basename)
    f.psdump(paste("report/",basename,"/radiometric.ps",sep=""))

    print("Edges") f.plotedge()
    f.psdump(paste("report/",basename,"/edges.ps",sep=""))

    # Now create output file # Dates f.tabout("D",basename)
    # Bounds f.textout(basename,"bounds.ps") #
    Probe f.textout(basename,"manual interp") #
    Pressure f.textout(basename,"pressure.ps") #
    Positions f.textout(basename,"centres.ps") # Bergeron
    f.tabout("B",basename) # Raw Rad f.tabout("G",basename)
    # Geom plots f.textout(basename,"geom1.ps")
    f.textout(basename,"geom2.ps") # Radiometric
    f.textout(basename,"radiometric.ps") #
    Edge f.textout(basename,"edges.ps") # PROBE
    f.textout(basename,"PROBE")

::::::::::::: f.static.R ::::::::::::: "f.static" <-
function (vec) { # Calculate static measures # rotation,
radial length, head length, jet width

  A_vec[17,] B_vec[57,]
  C_as.data.frame(c(A[1]+0.5*(B[1]-A[1]),A[2]+0.5*(B[2]-A[2])))
  # Degrees Rotation (rot) & Head Length (hl) # Slope AB <
  90 deg.  if(A[1]<B[1]){
    run_as.data.frame(c((B[1]-A[1]),(B[2]-A[2])))
    rot_atan(run[2]/run[1])
    hl_sqrt(run[1]^2+run[2]^2) r <- 3 * hl/sqrt(20)
    R_as.data.frame(c(C[1]+r*sin(rot),C[2]-r*cos(rot)))
  } # Slope AB > 90 deg.  else{
    run_as.data.frame(c((A[1]-B[1]),(B[2]-A[2])))
    rot_pi/2-atan(run[2]/run[1])
    hl_sqrt(run[1]^2+run[2]^2) r <- 3 * hl/sqrt(20)
    R_as.data.frame(c(C[1]+r*cos(rot),C[2]+r*sin(rot))) #
    return real rotation and run rot_pi-atan(run[2]/run[1])
    run_as.data.frame(c((B[1]-A[1]),(B[2]-A[2])))
  }

  # Mean Radial Length rads <- 69:73 rsum <- 0 for (i in
  69:73) {
    rads[i] <- sqrt((vec[i,1]-R[1])^2+(vec[i,2]-R[2])^2)
```

```
    rsum <- rsum + rads[i]
  } rmean <- rsum/5

  # Mean jet width m_run[2]/run[1] b_A[2]-m*A[1] jsum <-
  0 jpts <- 74:77 for (i in 1:4) {
    y_vec[jpts[i],2] xint_(y-b)/m
    dx_sqrt((vec[jpts[i],1]-xint)^2) w_dx*sin(rot) jsum <-
    jsum + w
  } jmean <- jsum/4 # remove NA's where jet not specified
  (I think ?)  if(!is.numeric(jmean$V2)){
    jmean_0 }

  result_as.data.frame(c(rot*360/2/pi, rmean-(2*r/3), hl,
  jmean$V2)) attr(result,"names")_c("Deg.","RL","HL","JW")
  return(result)
}

:::::::::::::: f.statplot.R :::::::::::::: "f.statplot"_
function(vecpat){
  tb_table(vecpat) par(mfrow=c(2,2))

  xlabs_c("Rotation (deg)","Radial Length","Head
  Length","Jet Width") for(i in 1:4){
    plot(tb[,i],xlab=xlabs[i],ylab="") lines(tb[,i])
  }

  par(mfrow=c(1,1)) }

:::::::::::::: f.table.R :::::::::::::: "f.table" <-
function (vpat) {
    vars <- ls(pat = vpat, pos = 1) # produce a matrix
    output for all static measures ln_length(vars) tab <-
    as.data.frame(matrix(1:(4 * ln), ncol = 4)) j <- 0 for
    (i in vars) {
j <- j + 1 cat(paste(" ",j," of ",ln,"\n")) obj <-
get(i, pos = 1) tab[j, ] <- f.static(obj)
    }

    # put in the labels
    attr(tab,"names")_c("Rotation","Radial Length","Head
    Length","Jet Width") return(tab)
}

:::::::::::::: f.tabout.R :::::::::::::: "f.tabout"_
  function(lead,base){
    obj_get(paste(lead,base,sep=""),pos=1) # bomb table
    has some non-numerics if(lead!="B"){
      obj_round(obj,digits=2) } else{
      B_obj$Bergeron # is this really necessary
      just to round a data.frame entry?
      obj$Bergeron_round(as.numeric(as.vector(B)),digits=2)
    }
    write.table(obj,file="tabout.txt",quote=FALSE,sep="\t")
    system(paste("cat tabout.txt | perl mktab.pl
    >>","report/",base,"/report",sep=""))
    system(paste("cat tabout.txt | perl mktab.pl
    >","report/",base,"/",lead,"tab.tex",sep=""))
  }

:::::::::::::: f.textout.R :::::::::::::: "f.textout"_
  function(base,file){ system(paste("echo
    report/",base,"/",file,"
    >>report/",base,"/report",sep=""))
  }

:::::::::::::: f.tp.R :::::::::::::: "f.tp"_ function(var){
  # interpol trange_rep.vecs$hours[length(rep.vecs$hours)]
  inter_spline(rep.vecs$hours,var,n=trange*6,method="natural")
  ln_length(inter$x) ret_1:ln for(i in 1:(ln-1)){
    rate_(inter$y[i+1]-inter$y[i])/(inter$x[i+1]-inter$x[i])*60
    ret[i]_rate
  } return(ret)
}

:::::::::::::: f.Varplot.R :::::::::::::: "f.Varplot" <-
  function(tab,xcol,ycol,sep=0,do.ord=F,spl=T){
    X_tab[,xcol] Y_tab[,ycol]

    if(do.ord){ ord_order(X) # sort order by X's variable
      values X_X[ord] # sorted X Y_Y[ord] # sorted Y
      (same order as X) print("Data WILL be ordered")
    } else{
      print("Data will NOT be ordered") }

    xy_na.omit(as.data.frame(cbind(X,Y)))
    # combine predictor and response variables
    attr(xy,"names")_c(colnames(tab)[xcol],colnames(tab)[ycol])
    reg_lm(xy[,2] ~ xy[,1]) plot(xy,type="p") # scatterplot
    of vars lines(x=xy[,1],y=fitted(reg),lty=2) # fit
    regression line

    return(summary(reg)) }

:::::::::::::: f.vcentres.R :::::::::::::: "f.vcentres"_
  function(vecpatt){

# collect storm centres vecs_ls(patt=vecpatt,pos=1)
  vcent_matrix(nrow=length(vecs),ncol=2) j_1 for (i
  in vecs){
    tcent_as.matrix(get(i,pos=1)) vcent[j,]_tcent[17,]
    j_j+1
  }

  result_as.data.frame(vcent)
  attr(result,"names")_c("easting","northing")
  return(result)
}

:::::::::::::: f.vdates.R ::::::::::::::
"f.vdates"_function(vpat){
  vecnm_ls(patt=vpat,pos=1)
  dtmat_matrix(seq(1:(4*length(vecnm))),ncol=4) j_1 for(i
  in vecnm){
    mon_as.integer(substr(i,4,5))
    day_as.integer(substr(i,6,7))
    hour_as.integer(substr(i,8,9))
    min_as.integer(substr(i,10,11))
    dtmat[j,]_c(mon,day,hour,min) j_j+1
  }
  hours_(dtmat[,2]-dtmat[1,2])*24+dtmat[,3]-dtmat[1,3]+(dtmat[,4]-dtmat[1,4])/60
  dtmat_cbind(dtmat,hours)

  dtf_as.data.frame(dtmat)
  attr(dtf,"names")_c("month","day","hour","minute","num.hours")
  return(dtf)
}

:::::::::::::: f.veccenpl.R :::::::::::::: "f.veccenpl"_
function(vecpatt){
  # collect storm centres vecs_ls(patt=vecpatt,pos=1)
  vcent_matrix(nrow=length(vecs),ncol=2) j_1 for (i
  in vecs){
    tcent_as.matrix(get(i,pos=1)) vcent[j,]_tcent[17,]
    j_j+1
  }

# vector times vtm_f.vdates(vecpatt)

# set bounds vbnd_c(min(vcent[,1])-1,min(vcent[,2])-1)
  vbnd_rbind(vbnd,c(max(vcent[,1])+1,max(vcent[,2])+1))
  plot(vbnd,type="n")

# plot vector positions points(vcent,pch=23) lines(vcent)
  vt1_paste(formatC(vtm[1,3],width=2,flag="0"),formatC(vtm[1,4],width=2,flag="0"),"Z"
  text(x=vcent[1,1],y=vcent[1,2],labels=vt1,pos=4)
  vplast_vcent[dim(vcent)[1],] vtlast_dim(vtm)
  vt2_paste(formatC(vtm[vtlast[1],3],width=2,flag="0"),formatC(vtm[vtlast[1],4],width
  text(x=vplast[1],y=vplast[2],labels=vt2,pos=3)

}

:::::::::::::: f.vectransin.R ::::::::::::::
"f.vectransin"_
  function(){ # return vectors in azimuthal projection
    vlst_list.files(path="vectrans",pattern="^dv") for(i
    in 1:length(vlst)){
      vpath_paste("vectrans/",vlst[i],sep="") #
print(vpath)
      assign(vlst[i],read.table(vpath),pos=1) #
print(read.table(vpath))
    } }

:::::::::::::: f.vectransout.R ::::::::::::::
"f.vectransout"_
  function(vpat){ # dump vector pattern for azimuthal
    projection in IDL vlst_ls(patt=vpat,pos=1)

    for(i in 1:length(vlst)){
      obj_as.matrix(get(vlst[i],pos=1))
      len_dim(obj)[1] obout_rbind(obj[17,],obj)
      fnam_paste("vectrans/",vlst[i],sep="")
      write.table(obout,quote=F,fnam)
    } }

:::::::::::::: f.vel.R :::::::::::::: "f.vel"_
  function(tab=rep.pres,distraw,mult=1,interval=1){
    # calculate ground track velocity between
```

```
    interval tseq_tab$num.hours len_length(tseq)
    trange_tseq[len]-tseq[2]+1
#    x1_tab$easting[1:(len-1)] #    x2_tab$easting[2:len]
#    y1_tab$northing[1:(len-1)] #    y2_tab$northing[2:len]

#    distraw_( (x2-x1)^2 + (y2-y1)^2 )^0.5
    times_tab$num.hours[2:len]-tab$num.hours[1:(len-1)]
#return()
    velspl_spline(tseq[2:len],distraw/times*mult,n=round(trange/interval),method="natural")
    velspl_as.data.frame(velspl)
    attr(velspl,"names")_c("num.hours","vel")
    return(velspl)
  }
```

A-5 IDL Programs

```
::::::::::::: dumptif.pro ::::::::::::: PRO dumptif
 envi_open_file, "thesis/apr96/aprIR.img", r_fid=fid
 envi_file_query, fid, bnames=nme for i=0,20 do begin
print, "writing " + nme[i]
envi_output_to_external_format,
dims=[0,0,511,0,511],fid=fid,
out_name="thesis/apr96/"+nme[i], pos=i, /tiff
 endfor END

::::::::::::: edgeloop.pro ::::::::::::: FUNCTION
edgeloop, indir
 jpgs=findfile(indir+'/*.jpg') msk=read_bmp('mask.bmp')
;set_plot, 'PS' for i=0,n_elements(jpgs)-1 do begin
  e=edges(jpgs[i],msk) plot,
  histogram(e),yrange=[0,5000],xrange=[0,1000]
  st=[mean(e),stddev(e),max(e)] print, st if (i eq 0)
  then begin
   sts=st endif else begin
   sts=[[sts], [st]] endelse
;  print, 'Key for next' ;  n=get_kbrd(1)
 endfor ;device, /close
 return, sts END

::::::::::::: edges.pro ::::::::::::: FUNCTION edges,
fname, msk ;PRO edges, fname ; Initialise image ; if not
dialog_read_image(file=fname) then stop
 read_jpeg, fname, imsrc

; compute edges ; imfilt=sobel(median(imsrc,7))*msk
 imfilt=sobel(median(imsrc*(imsrc ge 109),7))

 tvscl, imfilt*msk ; n=get_kbrd(1) ; non-zero elements
found with WHERE
 out=imfilt[where(imfilt)]

return, out END

::::::::::::: erun.pro ::::::::::::: PRO erun
 print, 'a96' s=edgeloop('indices/a96/../src')
 print, 'jA' s=edgeloop('indices/jA') print,
 'jB' s=edgeloop('indices/jB') print, 'n97'
 s=edgeloop('indices/n97') print, 'm98'
 s=edgeloop('indices/m98')
END

::::::::::::: groundtrack.pro ::::::::::::: PRO
groundtrack, srcdest ;+ Calculate great circle distances
between source and destination ;   by calls to map_2points.
;  Return Distance in km and bearing in deg. east of north.
;-

for i=0,n_elements(srcdest.id)-1 do begin
 gclen=map_2points(srcdest.e0[i],srcdest.n0[i], $
  srcdest.e1[i],srcdest.n1[i],/meters)
 bear=map_2points(srcdest.e0[i],srcdest.n0[i], $
  srcdest.e1[i],srcdest.n1[i]) print, gclen/1000, bear[1]
endfor

END

::::::::::::: projvec.pro ::::::::::::: PRO projvec
; project all v9* files to azimuthal through vectrack
function

; load template for read_ascii restore, 'vectpl.dat'
; get list of files vecfls=findfile('v9*')

; loop across each file for i=0,n_elements(vecfls)-1
 do begin
; recover a vector
  vin=read_ascii(vecfls[i],template=vectpl)
; project a vector vout=vectrack(vin)
; report output outname=strjoin(['d',vecfls[i]]) print,
  outname
; write file openw, 1, outname printf, 1, vout close, 1

 endfor END

::::::::::::: vectrack.pro ::::::::::::: FUNCTION
vectrack, vecsrc ;+ Calculate great circle distances
between point N and all N+n ; by calls to map_2points.
N is a duplicate of vec pt 17, the ;   storm centre.  ;
Return Distance in km x, y ;-

; store copy of centre point as centre of projection
centre=[vecsrc.e[0],vecsrc.n[0]]

; loop across all other points for
i=1,n_elements(vecsrc.id)-1 do begin
 gclen=map_2points(centre[0],centre[1], $
  vecsrc.e[i],vecsrc.n[i],/meters)
 az=map_2points(centre[0],centre[1], $
  vecsrc.e[i],vecsrc.n[i])
 bear=az[1] r=gclen/1000

; convert bearing to trig degrees ang=bear ; remove
 negative bearings if (ang lt 0) then ang=ang+360 ;
 reverse direction of angle ang=360-ang ; change quadrant
 ang=(ang+90) mod 360

 pi=3.141593 ; convert to (x,y)[km]
 if (ang le 90) then begin x=r*cos(ang/360*2*pi)
  y=r*sin(ang/360*2*pi)
 endif else $ if (ang le 180) then begin
  x=-r*cos((180-ang)/360*2*pi) y=r*sin((180-ang)/360*2*pi)
 endif else $ if (ang le 270) then begin
  x=-r*cos((ang-180)/360*2*pi) y=-r*sin((ang-180)/360*2*pi)
 endif else begin x=r*cos((360-ang)/360*2*pi)
  y=-r*sin((360-ang)/360*2*pi)
 endelse

; accumulate results ; if (i eq 1) then
res=[1,vecsrc.e[i],vecsrc.n[i],r,bear,ang,x,y] $ ;  else
res=[[res],[i,vecsrc.e[i],vecsrc.n[i],r,bear,ang,x,y]]
 if (i eq 1) then res=[x,y] $ else res=[[res],[x,y]]
endfor

; return results return, res

END
```

B-1 PWC MSC Infrared to Radiance Code

```
#define PWC_LowerBound -90.15
#define PWC_UpperBound 56.85
#define PWC_DoglegTemp -31.15
#define PWC_DoglegIntensity 0.6165
#define PWC_LowerDegrees ( PWC_LowerBound - PWC_DoglegTemp )
#define PWC_UpperDegrees ( PWC_UpperBound - PWC_DoglegTemp )
#define PWC_LowerSlope \
( ( 1.0 - PWC_DoglegIntensity ) / PWC_LowerDegrees )
#define PWC_UpperSlope   ( PWC_DoglegIntensity / PWC_UpperDegrees )


/*---------------------------------------------------------------------
-------
  PROCEDURE CERadianceToCelcius Kevin Carson   January 16, 1996
  -------------------------------------------------------------------
-------
  Converts an infrared pixel or radiance value to a degree value.  The
mapping
  is based upon the PWC standard for IR mapping of the GOES-GVAR
satellites.
  -------------------------------------------------------------------
-------*/
double CERadianceToCelcius
(
card8 radiance
)
{
double  intensity;

intensity = radiance / 255.0;

if ( intensity >= PWC_DoglegIntensity )
return
(
( intensity - PWC_DoglegIntensity )
/ PWC_LowerSlope
+ PWC_DoglegTemp
);

else /* if ( radiance < PWC_DoglegIntensity ) */
return PWC_UpperBound - intensity / PWC_UpperSlope;
} /* CERadianceToCelcius() */
```