Machine-Vision Aids for Improved Flight Operations

Final Technical Report
Prepared Under NASA Grant Number NCC2-841
Period of Performance: January 1994 - October 1996
Attn: Dr. B. Sridhar, MS 262-1/ Mr. Philip Smith, MS 210-1
NASA Ames Research Center
Moffett Field, CA 94035-1000

Principal Investigator: Dr. P. K. Menon
Graduate Research Assistant: Gano B. Chatterji
Santa Clara University
Department of Mechanical Engineering
Santa Clara, CA 95053

November 1996

Abstract

The development of machine vision based pilot aids to help reduce night approach and landing accidents is explored in this report. The techniques developed in this report are motivated by the desire to use the available information sources for navigation such as the airport lighting layout, attitude sensors and Global Positioning System to derive more precise aircraft position and orientation information. The fact that airport lighting geometry in known and that images of airport lighting can be acquired by the camera, has lead to the synthesis of machine vision based algorithms for runway relative aircraft position and orientation estimation.

The main contribution of this research is the synthesis of seven navigation algorithms based on two broad families of solutions. The first family of solution methods consist of techniques that reconstruct the airport lighting layout from the camera image and then estimate the aircraft position components by comparing the reconstructed lighting layout geometry with the known model of the airport lighting layout geometry. The second family of methods comprises of techniques that synthesize the image of the airport lighting layout using a camera model and estimate the aircraft position and orientation by comparing this image with the actual image of the airport lighting acquired by the camera. Algorithms I through IV belong to the first family of solutions while Algorithms V through VII belong to the second family of solutions. Algorithms I and II are parameter optimization methods, Algorithms III and IV are feature correspondence methods and Algorithms V through VII are Kalman filter centered algorithms. In order to take advantage of the aircraft dynamics and the multiple images available along the glide path, the position estimates provided by Algorithms I through IV are

used for driving a six–state Kalman filter for providing improved estimates of the aircraft position and inertial velocity components. Algorithms V through VII are Kalman filter centered algorithms and are designed to implicitly utilize the aircraft dynamics and the multiple images available along the glide path. Additionally, Algorithm VI integrates the position information derived from a Global Positioning System receiver.

Results of computer simulation are presented to demonstrate the performance of all the seven algorithms developed in this report. It is shown that all the algorithms meet some or all of the Federal Aviation Administration specified navigation accuracy requirements for various landing categories. These results show that it is feasible to design an accurate machine vision based night landing aid with the currently available technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

The Need for Pilot Aids

Landing is one of the most demanding flight regimes in fixed–wing aircraft operations. This fact is borne out by the statistic that the landing phase of flight alone accounts for 29% of all the aviation accidents. Approach and landing accidents together account for 41% of all aircraft accidents [5]. Research shows that night approach accident rates are about eight times that of the day rate [9]. This is perhaps attributable to difficulties associated with reduced lighting during the nighttime hours. Clearly, out–of–the–window references, navigation aids, and air traffic awareness are significantly impacted during these low–light conditions. Furthermore, the human body is primarily adapted for daytime activity. Night flying places the pilot's eyes, which provide the primary sensory information needed for flight, in an environment for which they are only partially suited. Limitations of the human visual system along with aircraft motion are responsible for numerous static and dynamic illusions which can have dangerous consequences on night landing [40]. Thus, in addition to the usual landing hazards such as winds aloft, and complex approach procedures employed at airports near population centers, night landing can further add to the pilot work load.

Landing aids such as the Instrument Landing System (ILS) and the Microwave Landing System (MLS) can be used to ameliorate the night landing difficulties. Due to their high cost, these systems are likely to be available only at a few

major airports. Given the operational advantages of all weather landing at any airport, large commercial carriers are likely to equip their airplanes with such systems. Currently, ILS systems are routinely used by these air carriers to land their airplanes. Emerging Global Positioning System (GPS) technologies hold the promise for low-cost, high precision landing guidance. GPS-based landing systems are likely to find widespread applications in the aeronautical operations.

Smaller air carriers and general aviation aircraft which are not equipped with INS can only navigate along the Victor Airways or Jet Routes to the destination airport using very-high-frequency omnirange equipment (VOR) and distance measuring equipment (DME) [49]. Once the destination airport is visible, runway lighting is used for obtaining alignment guidance. Visual approach slope indicator (VASI) or precision approach path indicator (PAPI) lights are used for obtaining glide slope information. The objective of the research given in this report is to explore the development of a pilot aid that can help reduce night approach and landing accidents. The research focus is on developing an onboard instrument that complements existing cockpit instrumentation.

The techniques developed during the course of this research are motivated by the desire to use the emerging machine vision techniques along with the existing infrastructure to derive more precise aircraft state information. Decreasing costs of machine vision systems and components places this technology in an attractive position. Even if a highly sophisticated landing system were to become available, runway lighting will continue to be in use. Thus, the machine vision based system will be the ultimate back-up landing system. As and when GPS becomes cheaper and more accurate, the machine vision system can be used to further add value to it. Finally, even though the focus of this report is general aviation application, there is no reason why the algorithms and methods proposed here cannot be used in commercial and military aircraft.

In order to further motivate the development of landing aids, factors that make night landing hazardous are next examined.

1.1 Pilot's Health Condition

The pilot is required to be in good health in order to cope with all the situations encountered during night flying. The following factors are indicated in [40] as symptoms of changing health. Sleeping problems, chronic fatigue, gastric disturbances, shortness of breath, appetite changes, reduced eye–hand coordination or muscle tremor, high blood pressure, and body weight change of more than ten percent when not dieting. Of these, the important ones are sleep disorders and fatigue.

Pilots, like other human beings, experience regular sleep and wakefulness cycles in consonance with the day–night cycles. This is known as circadian rhythm. This rhythm resets the biological activities once every cycle. Pilots are required to stay awake during night flight which conflicts directly with their need to sleep during the nighttime hours. Lack of sleep causes sleep disorders and fatigue. Sleep disorders are also caused if one's sleep hours are shifted to a new time period during the day. For example, transmeridian flights require synchronization of body rhythms to new time zones. Usually, this adjustment is accompanied by loss of appetite and tiredness. Other side effects of sleep deprivation are short–term memory loss, reduced attention span, reduced judgement capability, increased irritability and anxiety, and increased reaction time.

Fatigue can be defined as a general loss of well–being caused by physiological and psychological factors such as inadequate rest or sleep, intense mental activity, limited visibility, seat discomfort, airplane vibration and noise, and excessive radio communications. Pilot response to fatigue is very similar to that caused by sleep deprivation.

Sleep deprivation, fatigue and a number of other factors that impair pilot performance and the physiological and psychological responses to these causal factors are described at length in [40]. Steps needed for preventing and overcoming night pilot health related difficulties are also listed in Reference [40].

1.2 Flight Situational Awareness

In order to maintain flight safety, a correct assessment of aircraft attitude is needed at all times. Although all aircraft include cockpit instrumentation needed for safe flight operations, to a large extent pilots base their sense of orientation on visual, vestibular and somatosensory systems. Often these reflexes give a false sense of attitude. Therefore, a trained pilot consciously suppresses the unwanted vestibular and somatosensory reflexes, and uses only the information that is visually derived [24]. However, a number of potentially dangerous situations may be attributed to the information provided by vestibular and somatosensory systems.

The vestibular system consists of the semicircular canals and the otolith organs of the middle ear. The semicircular canals and otolith organs provide information about angular and linear accelerations, respectively. In addition, the otolith organs also sense the direction of the gravity vector. The information provided by the vestibular system is needed for stabilization of the eyes during head or body motions, which would otherwise result in blurred vision. In the absence of vision, accurate motion and orientation perception when on ground is also provided by the vestibular system. Although, the vestibular system is ideally suited for the ground environment, it is only partially suited for the flight environment. Under certain flight conditions it can generate false perceptions.

The somatosensory system responds to pressure and stretch. It consists of somatosensory sensors that are distributed in several body structures, including, skin, joints and muscles. This system is responsible for the so-called "seat-of-the-pants" sense referred to by pilots [24]. Like the vestibular system, the somatosensory system can also generate false perceptions under certain conditions.

In addition to the vision, vestibular and somatosensory systems, pilots learn to use the auditory system to get a sense of airspeed and attitude based on the wind noise in the cockpit [24].

Compared to the vestibular and somatosensory systems, the visual system

provides more accurate orientation information. In situations such as nighttime flight operations, the visual information is considerably degraded, forcing the pilots to depend on less accurate vestibular and somatosensory systems. In the next section, the impact of reduced lighting flight operations on the visual system is examined.

1.3 Vision at Night

A pilot's vision provides the primary sensory information required for flight. Hence, it is important to examine how the human visual system is impacted during the twilight and nighttime hours. The visual information is combined with other sensory information, memory and domain knowledge via complex mental processes to result in visual perception or understanding of the scene. For correct interpretation of the flight situation, both static and dynamic visual cues are needed.

The cockpit layout around the night aviator plays an important role in providing a frame of reference for the pilot. It is with this reference that the pilot perceives himself or herself to be a fixed component of the aircraft. Static structure of the cockpit aids the pilot in making appropriate control inputs by providing a stable visual reference for judging changes relative to the external environment. Static cockpit structure is so important that excessive head motions have been known to result in a sense of uncertainty about the aircraft attitude.

Static structures external to the cockpit such as the aircraft nose restrict external visibility. To overcome this difficulty to some extent, a design eyepoint is specified for the cockpit to permit optimal internal and external visibility. Pilots are required to be positioned correctly with respect to the design eyepoint. However, over time pilots may have a tendency to slump down into the seat, thus lowering the eye position by a couple of inches, thereby causing a significant deviation from the design eyepoint. This is very significant during night landing since,

deviations from the design eyepoint can result in diminished visual range. This could cause the runway lights during the final approach to appear later than they would have if the visual range were greater.

Spatial reference is also established by the ground plane which provides the horizon. Objects of known size on the ground provide scale and distance information. The motion of the objects in the visual field provides information about groundspeed. During nighttime the horizon and the objects are difficult to see. In some situations this can lead to a complete loss of spatial orientation. Such disorientation causes symptoms of fright, airsickness and dizziness. The recommended procedure in such situations is to switch the pilot's attention to the cockpit instruments.

Due to the greatly reduced visual information during nighttime flight operations, pilots are unable to compensate for perceptual disturbances. A major cause of perceptual disturbances is head motion. During and after rolling and pitching head motions, pilots have reported a feeling that the flight situation may be less safe and secure. This is probably due to conflicting information from vestibular sense organ and the visual system. Due to this reason, the head should be kept as motionless as possible. However, pilots do have to continually scan the external environment and cockpit instruments. Since body motions are deliberately carried out, any apparent motion of cockpit structures, such as window frames, relative to the external environment are attributed to the body motions. All other motions are inferred to be due to aircraft motion. These two types of motion are not easily distinguishable by the night pilot because the visual cues needed for correct interpretation are either lacking or are considerably degraded during nighttime hours.

The combination of reduced lighting, perceptual disturbances and the motion of the outside scene perceived by the pilot give rise to a number of potentially dangerous visual illusions. A few commonly encountered illusions are discussed next. Reference [40] discusses these in further detail.

1.4 Visual Illusions

A visual illusion is a false perception of reality. Often, false perceptions are a consequence of logical interpretation by the observer. Visual illusions can occur when there is differential motion between the outside scene and the aircraft that is perceived within the pilot's field–of–view. They also occur in situations when the outside scene moves across the pilot's field–of–view during relatively stable visual fixation. The commonly known visual illusions that a night pilot is faced with during descent and final approach for landing are described in the following sections. These descriptions are primarily based on [40].

1.4.1 Runway Length/Width Illusion

During the final approach to landing, pilots gauge the aircraft position with respect to the runway and the glide slope by how long and wide the runway appears from their viewing position. During the night, objects of known size and shape on the runway surface are not clearly visible. Therefore, the length/width illusion may arise because of what is observed differs from the pilot's expectation. If the runway width appears to be larger, the pilot perceives the aircraft to be below the normal glide path. A narrower runway on the other hand gives the illusion of being high.The latter can cause the pilot to increase the rate of descent. Since the aircraft is close to the ground, by the time the pilot realizes that the aircraft will land short, there may not be enough lift margin to arrest the rate of descent [24].

1.4.2 Foreshortening Illusion

Foreshortening illusion pertains to when the true shape of the objects such as terrain features appear to be more elliptical or shortened when viewed from a distance along the glide slope.

1.4.3 Sloped Runway Illusion

If the ground surface is not level, as in the case of sloped runways, the visual cues effect pilot's judgement of the aircraft altitude and glide slope. Usually, runways are at the same level as the surrounding terrain. Therefore, the visual information from the terrain reinforces the runway perception. In situations where the runway actually slopes with respect to the ground while pilots expect the runway to be level with the ground, it has been observed that steeper approaches are flown to upsloped runways and shallower approaches to downsloped runways. The illusionary condition causes the pilot to land short of the touchdown point on runways with upslope, and to overshoot the touchdown point on runways with downslope.

1.4.4 False Horizon Illusion

False horizon illusion mainly relates to misinterpretation of the location of the horizon within the field–of–view. One form of this illusion occurs when lights on the ground appear to merge with stars. This results in pilots placing the aircraft in unusual attitudes in an attempt to keep some ground lights above, having perceived them as stars. Another form of this illusion occurs when several lights are seen beyond the runway at a higher elevation. These lights give the impression of a horizon, prompting the pilots to fly below the glide slope.

1.4.5 Vertical Position Illusion

Well lighted objects or terrain features that are farther away from the pilot appear higher on the horizon. This may give the impression that the aircraft is higher on the glide slope than it actually is. This can result in a descent rate increase reaction. Vertical position illusion when combined with false horizon illusion leads to other illusions. One of these occurs when the pilot observes a light located on the ground a small distance ahead and to the side. The pilot may have to look

at a downward angle to observe the light relative to the wings. This gives the impression that the line–of–sight to the ground is more level which may cause the pilot to assume a nose up attitude. At low altitudes, as the pilot looks downward at the ground light when the horizon is invisible, a small bank angle may develop. In this case, the pilot is unable to perceive the development of this dangerous bank angle.

1.4.6 Illusions Caused by Fog and Rain

As pilots descend to the runway, presence of fog causes the runway lights to appear less bright, causing a misperception of the actual distance from the runway. Pilots are led to believe that the aircraft is farther away than it truly is. Refraction caused by heavy rain on the windshield results in ground lights to appear from an apparent location. This may give rise to errors in perceived altitude. Rain can also cause the runway to appear larger in size when compared to clear air conditions and can cause the horizon to appear closer. Heavy rain can often cause the complete disappearance of the horizon. When an approach is made through fog or haze, vertical visibility is better than forward visibility. This causes the ground lights farther ahead to appear less bright leading to the illusion that the aircraft is pitched up.

1.4.7 After–Image Illusion

A visual after–image remains when an observer views a bright light at night. For example, a camera flash bulb leaves a visual after–image subsequent to going off. This after–image results in the illusion that the environment is more static, hence, attitude changes are not perceived during this period. This illusion is encountered specially during approach to a runway, since, high intensity strobe lights placed along the runway approach centerline are used for indicating approach direction. Once a certain altitude is reached, pilots often request the control tower

to turnoff these lights.

1.4.8 Ganzfeld Depth Loss Illusion

Ganzfeld is a German word for a featureless visual scene. This illusion results in loss of depth perception when flying over snow fields, bodies of water or any other featureless terrain. Some features are required on the terrain so that location of one feature is judged to be different than the location of another feature. Without this prerequisite, depth discrimination is impossible. At night, unilluminated areas of the terrain with vastly different features appear continuous. For example, bodies of water smoothly merge with land in the visual scene.

1.4.9 Foreground Occlusion Illusion

This illusion is most often experienced when the ground lights are cutoff by a cloud. In a moonlit night, pilots can detect the cloud by its reflection. However, in a dark night such discrimination is not easy. A more dangerous version occurs during descent at night over mountainous terrain. During a portion of the descent, the lights on the runway are visible to the pilot and the foreground occlusion such as a hilltop lies invisible. At some point along the descent, the lights are cutoff by the hilltop. When such a situation is detected, pilot should climb immediately or else a collision with the terrain would occur. It is easy to see how this illusion could cause confusion in judging whether a hilltop or a cloud is the cause for the foreground occlusion. Detailed terrain knowledge is one of the useful sources of information for correct interpretation.

1.4.10 Up–Sloped Lighted City Illusion

This illusion is experienced when terrain stays level for some distance and then rises to give the impression of two intersecting planes in the pilot's field–of–view. Often there are parallel roads with street lights in a city situated on the upward

sloping terrain. Long rows of street lights appear to converge at a distance giving the impression of a horizon. The runway lights, situated on the level terrain in the foreground, may also appear to converge at a different vanishing point. The pilot can be tricked into believing that the broad upward sloping terrain is level and that the runway is sloped down. This may cause the pilot to increase the descent rate.

1.4.11 Autokinetic Illusion

When an observer stares steadily at a single motionless source of light at night, autokinetic illusion gives the appearance that the source of light is moving around in random directions at varying speeds. Due to this illusion, an isolated motionless ground light may appear to be moving on the ground. One possible erroneous interpretation is that another aircraft is in the vicinity. Autokinetic illusion can also cause a visible star to be misperceived as a moving vehicle on the ground, giving the impression of low pitch attitude to the pilot.

1.4.12 Black Hole Approach Illusion

Black hole approach illusion arises during night approaches where no ground details are visible short of the runway. Four different types of black hole approach situations have been described in [40]. The main factor that causes this illusion is that pilots derive vertical guidance information in the angle between the line–of–sights to the farthest and the nearest light. If an aircraft is flown at a constant altitude, the angle is expected to increase as the aircraft nears the runway. Similarly, the angle should decrease as the aircraft descends. In cases where the pilot is unable to perceive visual angle change, a more rapid descent is flown. Problem occurs in situations where the aircraft descends into the terrain much before the runway.

This concludes the discussion of visual illusions. A few vestibular and so-

matosensory illusions are described next.

1.5 Vestibular and Somatosensory Illusions

Vestibular and somatosensory illusions are caused by the linear and angular accelerations perceived by the pilot.

Somatogravic illusion is a sensation of change of attitude when the otolith organs are subjected to linear acceleration. This illusion occurs in level flight giving the pilot a false cue of being in a nose high attitude during acceleration. The opposite illusion of nose down attitude occurs during deceleration on final approach. A pilot may create a low altitude stall in the process of correcting for this illusion [24]. A variant of this illusion is the inversion illusion in which the G forces acting on the otolith organs give the sensation of being upside down, when the pilot is being subject to negative G forces [24].

During a coordinated turn, the "seat-of-the-pants" sense is misleading because the resultant of the gravitational and centrifugal forces is directed towards the floor of the aircraft, which the pilot falsely perceives as the direction of the vertical [24].

Coriolis illusion occurs during prolonged turns in one plane. The sensation of turn perceived by the semicircular canals in the inner ear at the beginning of the turn subside during the prolonged turn. A sudden head movement causes the canals to sense angular acceleration which gives the impression of rotation in another plane. Attempts to correct for this illusion can place the aircraft in dangerous attitudes [24]. The coriolis illusion is specially hazardous during curved approach because of the aircraft's proximity to ground. Furthermore, it can cause disorientation and can produce intense symptoms of nausea [40].

"Leans" is a common illusion caused by rapid roll maneuvers to correct for roll angle developed by subtle bank. For example, if a subtle bank angle develops to the left such that the vestibular system is unable to detect it, the pilot eventually

detects the roll on the attitude indicator and corrects it by a rapid roll to the right. The pilot gets the false sense of only having rolled to the right.

Giant hand illusion gives the impression that an external force is pushing on the aircraft or holding it at a certain attitude. This is caused by vestibular and somatosensory inputs that interfere with pilot's conscious control of the aircraft. If the disorientation is about the pitch axis during aircraft acceleration, the aircraft appears to resist pilot efforts to pull the nose up because the natural reflex is to push the nose down [24]. This illusion also occurs when the disorientation is about the roll axis as in the "Leans" illusion. In these cases, the aircraft seems to resist roll efforts by the pilot.

In addition to the illusions described here, a number of vestibular and somatosensory illusions can occur in high performance aircraft during maneuvers such as graveyard spin, graveyard spiral and rapid aileron rolls. These are described in further detail in Reference [24].

With the background of sensory illusions that the night pilots often experience, the potential use of machine vision systems in ameliorating the impact of these illusions is examined next.

1.6 Machine Vision Systems As Pilot Aids

Based on the preceding discussion of the human perceptual system and how prone it is to visual, vestibular and somatosensory illusions, this report attempts to answer the question: Can a machine vision system augment the pilot's perception sufficiently to avoid these illusions?

Before an attempt is made to answer this question, it is necessary to establish the underlying causes for the various illusions described earlier. Closer analysis reveals that they can be classified into three distinct groups based on the underlying causal factors. Those that occur because of imprecise knowledge of geometry, those due to conflicting information from the vestibular, somatosensory and vi-

sual systems, and those arising from the limitations of the human eye. Runway length/width illusion, foreshortening illusion, sloped runway illusion, false horizon illusion, vertical position illusion, Ganzfeld depth loss illusion, foreground occlusion illusion, up–sloped lighted city illusion, and black hole approach illusion, all have their root in imprecise knowledge of the terrain, runway and lighting geometry. Autokinetic illusion and the various vestibular and somatosensory illusions have their roots in conflicting information received from the vestibular, somatosensory and vision systems. The third group of illusions, which includes after–image illusion and fog and rain caused illusions, has its basis in the physical limitations of the human eye. After–image illusion is caused due to saturation of the visual receptors in the eye. Fog and rain cause the runway lights to appear diffused. In this situation, the eye has no mechanism for enhancing the appearance of these lights.

In order to examine how a machine vision system could be functionally superior to its human counterpart, consider the analogy between the human perceptual system and the machine vision system shown in Figures 1.1 and 1.2.

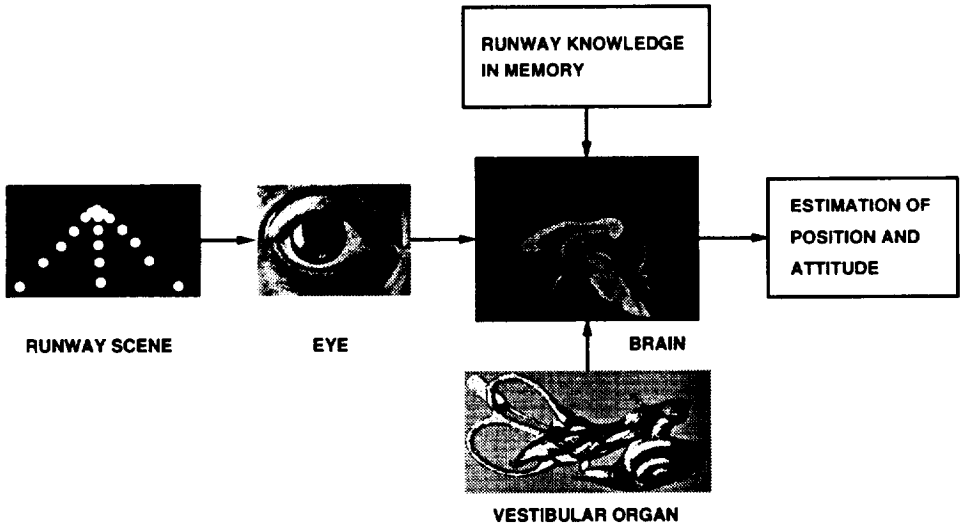

Figure 1.1: Human perceptual system.

The human perceptual system is mainly driven by three sources for the land-

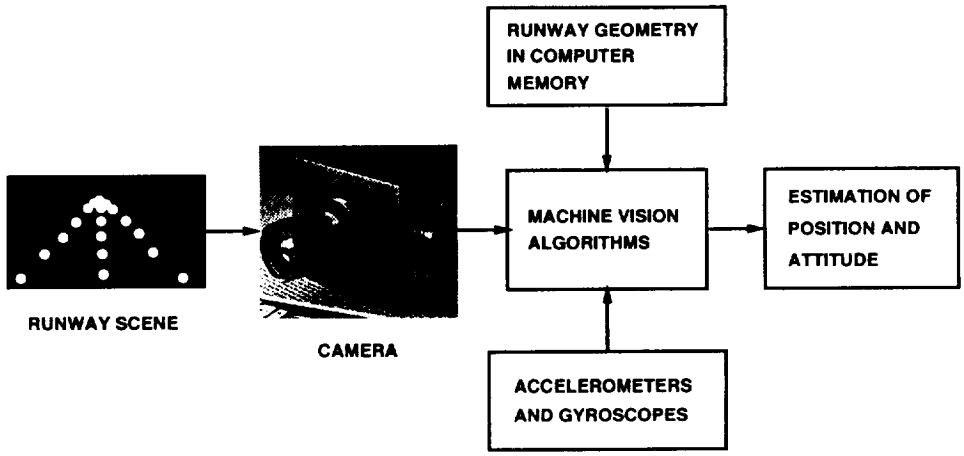

Figure 1.2: Proposed machine vision system.

ing task: runway lighting as seen by the eye, motion sensed by the vestibular and somatosensory systems, and runway knowledge learnt and stored in the memory. An analogous machine vision system could also be driven by equivalent sources: runway lighting seen by the camera, motion sensed by accelerometers and gyroscopes, and knowledge of runway geometry encoded in the computer memory. In the human perception system the brain integrates the input information. In the machine vision system this function can be accomplished by computer–resident algorithms. Thus, according to this analogy, the camera can be considered equivalent to the human eye, accelerometers and gyros equivalent to the vestibular organ, and the geometry information available from digital memory can be considered to be equivalent to the domain knowledge in the human brain.

The exact geometric information encoded in the digital memory of the computer is precise when compared with the approximate runway geometry knowledge stored in the human brain. For the machine vision system, this fact offers the immunity to visual illusions caused by imprecise knowledge of the runway geometry.

Accelerometers and gyros are precision instruments which far exceed the capabilities of the human vestibular and somatosensory systems. In addition, these sensors provide true motion of the aircraft. In situations where the pilot moves relative to the airplane, the vestibular and somatosensory systems sense a combi-

nation of the aircraft motion and the pilot's motion. Thus, it is difficult for the pilot to differentiate between self motion and aircraft motion.

Unlike the human eye, the camera sensor elements can be designed to have optimal sensitivity to runway lights. Additionally, optical filters can be used for reducing or eliminating certain frequencies from the visible spectrum. They can also be designed to avoid saturation of the photosensitive elements. Thus, after-image illusion can be effectively eliminated in a machine vision system.

So far, information sources which have the potential of providing superior quality information to the machine vision system have been discussed. However, the critical component of a machine vision system is the algorithm for estimating runway relative position and attitude of the aircraft. The point of view adopted in this research is that two categories of algorithms based on sound physical and mathematical principles are needed for algorithm development. Firstly, methods for conditioning the image output from the camera are required. Secondly, methods for integrating information from the image, motion sensors, and the stored runway geometry, for runway relative position and attitude determination need to be developed. Both categories are topics in the Computer Vision or Machine Vision literature. Assuming that such algorithms can be designed, the next issue relates to what is available in the literature and what has been accomplished so far. The following chapter provides a brief summary of Computer Vision and the literature relevant to the design of such algorithms.

The preliminary discussion in this section provides a glimpse at the possibilities offered by a machine vision system. Although current generation imaging sensor technology is adequate for the design of a machine vision system, future improvements will only enhance the capability of such a system.

1.7 Summary

The complexity of the landing task and the hazards of night operations were discussed in this chapter. Pilot health issues, flight situational awareness, vision at night, visual illusions, vestibular and somatosensory illusions were discussed at length. A study of these issues indicated that a low cost, machine vision based position and orientation instrument was required for general aviation. Since human pilots are able to fly the aircraft along the descent path purely by visual reference, it was argued by drawing an analogy between the human perceptual system and the machine vision system that a machine vision system could be designed for deriving runway relative position information during approach and landing without being subject to optical illusions. Reasons were given for expecting higher reliability of the machine vision system as opposed to the human perceptual system, specially in cases where precise knowledge of runway geometry is required. It was pointed out that the algorithms are the key component of the machine vision system.

Chapter 2

The Machine Vision Technology

Machine or Computer Vision technology deals with algorithms and methods for two dimensional digital signal processing, pattern classification, image segmentation, geometric modeling, and relational structures. Text books in this area [6, 27, 47, 53, 78] cover many of the topics of machine vision. Many of these text books have an Artificial Intelligence flavor, focusing on heuristics of machine vision technology, Reference [6] being an example of this approach. A few examine the issues from a signal processing point of view, while others are motivated by statistical decision theory. Representative examples of these two approaches are References [53] and [27]. These references cover most of the tools and techniques used in machine vision system development and research.

The range of topics addressed in machine vision technology can be organized as a sequence of representations from early or low-level vision to cognitive interpretation [6]. Starting with an image generated by the camera, early or low-level vision algorithms are used for image conditioning such as filtering, edge detection, and optical flow computation. The output of this process is usually encoded into an image format, often called intrinsic or generalized image [6]. Higher level algorithms use these intrinsic images as inputs and gather regions within the image that are likely to be associated with objects being viewed. For example, neighboring pixels in the image which have the same color can be grouped together to

represent an object. Higher-level algorithms also assign geometric properties such as shape, area, eccentricity, compactness, and boundary length to regions within the image. This representation is natural if a database containing all the shapes making up the scene are available. Shape properties can be used for matching the image with models in the database, permitting the derivation of the location and orientation of the observer. Finally, higher level algorithms may use rules of logic to infer about what is seen. Clearly, this function is very much dependent on the domain. With this brief introduction to machine vision, modern solid state imaging systems are examined next. Some of the low-level and high-level functions are examined.

2.1 Modern Solid State Imaging Systems

Electronic imaging technology has changed considerably since the introduction of photoemissive storage tubes which use incident light to emmit electrons in a pattern corresponding to the brightness of the scene. The Iconoscope was the first practical device of this type. This was soon replaced by Image Orthicon. The low signal-to-noise ratio of these devices led to the development of photoconductive devices. Photoconductive devices are based on principle of change in electrical resistance of a photoconductor when exposed to light. Vidicon, Plumbicon and later Saticon are devices of this type. More recently, solid-state devices called charge-coupled devices (CCDs) have found an increased use in the consumer electronics. These devices provide good signal-to-noise ratio along with the advantages of small size, low power consumption and low cost. A modern solid state CCD camera unit is shown in Figure 2.1.

Cameras convert electro-magnetic radiation received within a certain field-of-view into electrical signals encoded to form a two dimensional array. This general definition is applicable to visible-range and infra-red camera systems. Thus, one way to classify an imaging system is by its operating range within the electro-

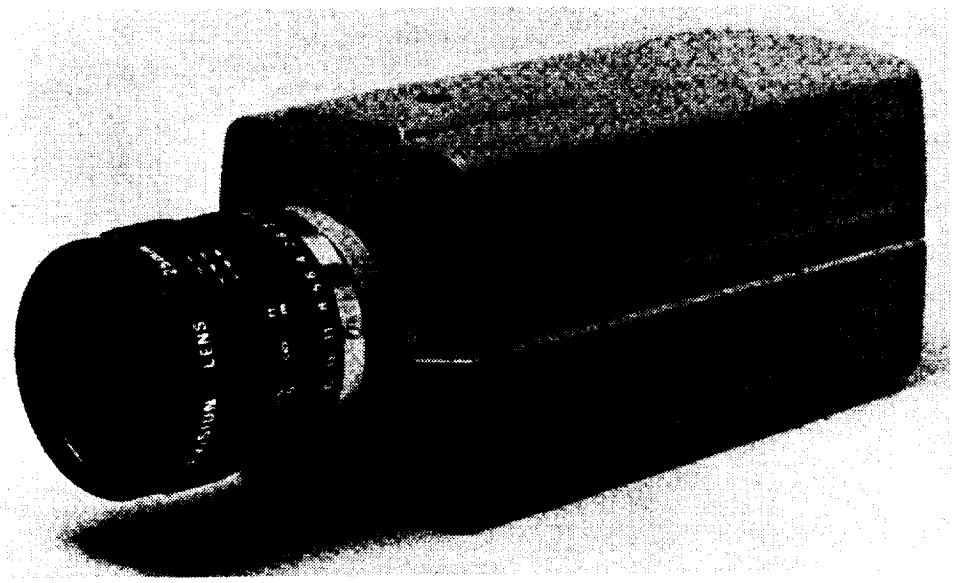

Figure 2.1: A Modern Solid State Camera.

magnetic spectrum. This report will be mainly concerned with image sequences generated by Television (TV) cameras, although some of the algorithms are applicable to the infra–red systems as well.

Major components of a solid state TV camera such as the one shown in Figure 2.1 are the lens, iris, shutter, photosensitive sensor array and camera electronics. In addition to these, color cameras employ beam splitters. A compound lens is used for adjusting the focal length for projecting the image of the viewed scene on to the photosensitive sensor array. Iris controls the amount of light that is allowed to reach the photosensitive sensor array. Photosensitive sensor array is the sensing element that converts images into electrical signals. Imaging systems are often classified by the type of photosensitive sensor used, charge coupled devices and charge induced devices (CID) being two examples. The camera electronics provides timing signals for shuttering and downloading the signal from the photosensitive sensor array, noise removal, signal conditioning, pre–amplification, amplification, image encoding and several other functions that are needed for generating acceptable images.

A CCD sensor array consists of rows and columns of photosensitive elements arranged in a rectangular array on a silicon substrate. Pixel size is the term used for describing the size of an individual photosensitive element. These elements collect and store electrical charge in proportion to the intensity of the light incident on their surface. The charges are electronically transferred to the device output to form the output video signal.

The resolution of a CCD camera depends on the number of photosensitive elements. The number of rows determine the vertical resolution and the number of columns determine the horizontal resolution of the camera. Typically, a National Television Systems Committee (NTSC) format CCD camera is designed with 492 rows and 510 columns [52].

Three commonly used architectures for CCD transfer and readout are: (1) frame transfer, (2) interline transfer and (3) frame and interline transfer [52]. Frame transfer architecture uses a sensor array, a storage register array and a horizontal output register. The sensor array is allowed to collect charge for a complete frame. Commanded by a clock, the charges in each column are transferred vertically from element to element to a corresponding column of the storage array. This process empties the sensor array for the next frame. The charges from the storage array are transferred one row at a time to the output register in synchronism with clock commands. The output register generates the video signal.

The main disadvantage of this architecture is that the photodiodes saturate if exposed to bright light for a long duration. To overcome this problem, a mechanical shutter is employed to permit the CCD sensor array to be exposed to bright light for a short duration. Clearly, this introduces a mechanical element into an otherwise all electro–optic device. This is specially of concern if the camera is to be used in a high vibration environment.

In the interline architecture, every photosensitive element in the sensor array is connected to a neighboring storage element. The storage elements are arranged in columns next to sensor element columns. Once charge is collected, commanded

by a clock, charge is transferred horizontally from each sensor element to the neighboring storage element which frees up the sensor element for the next frame. Charge is then transferred vertically from storage element to storage element in each storage column. Finally, under a clock command, the charges from the storage array are shifted one row at a time into the horizontal output register, similar to the frame transfer architecture discussed earlier. The video signal is then read out from the output register.

This architecture has the advantage of being resistant to blooming and smear because of the rapid transfer from the sensor array to the storage array. The main disadvantage is that placement of storage elements next to sensor elements causes the sensor to have lower pixel density.

Frame interline architecture employs a row of selection gates between the sensor and storage elements so that excess charges are drained from the system before being transferred to the storage columns. This architecture is similar to the frame transfer architecture with the added advantage of being resistant to blooming and smear.

Blooming occurs when a CCD sensor element saturates and spills charge to the neighboring elements. This gives the appearance of a large bright spot in the image. This effect may be seen in the image of the runway lighting, acquired by a CCD camera, shown in Figure 2.2. In order to overcome this problem, more expensive CCD sensor elements are designed with built in anti-blooming gates which remove the excess charge.

Sensitivity is an important measure of camera performance. Camera sensitivity is defined as the amount of light that is needed to produce a video signal of certain magnitude. For example, sensitivity can be characterized by the amount of light in units of Candle Power required to produce a gray-level of 255 in an 8 bit system. A more sensitive camera requires less amount of light to produce the same output as a less sensitive camera. Camera sensitivity can be adjusted to a certain extent by increasing the video gain in the camera. The disadvantage

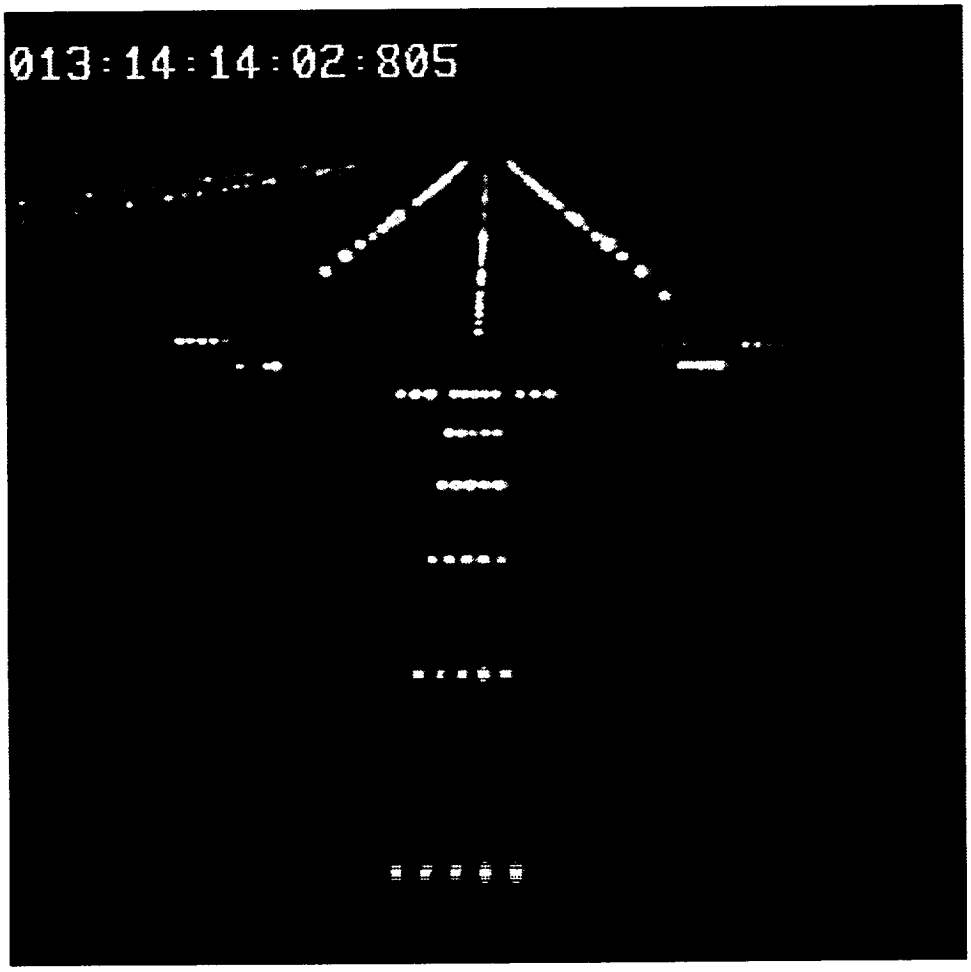

Figure 2.2: A CCD camera image during night landing illustrating the "blooming" effect.

in increasing sensitivity is that it decreases the signal–to–noise ratio. For CCD cameras, signal–to–noise ratio is directly related to the camera sensitivity.

Dynamic range of a CCD camera is a measure of its range of useful operation. It is defined as the ratio of the number of electrons required for maximum charge to the number of electrons that accumulate at the charge site if no light is incident on it. This ratio is often expressed in decibels (dB). As an example, if 80,000 electrons represent full charge and 20 electrons represent the dark current, the dynamic range is 72 dB.

Integration time is defined as the duration in which charge is allowed to accumulate in the charge sites of the CCD array. The integration time is controlled by electronic shuttering or by the selection of the readout pulse width.

Gamma is another term commonly associated with TV cameras. An image gamma of unity means that the system correctly reproduces the gray–levels of the scene [52]. If gamma is greater than unity, the image appears sharper but the scene contrast range is reduced. Reduction of gamma makes the image appear washed out [52]. Since the CCD is a nearly linear device, its output signal is directly proportional to the scene illumination. However, the phosphors used in display monitors behave nonlinearly. Typically, they have lower gain for dark signals and higher gain for bright signals. To compensate for this, higher gain is used for dark signals and lower gain for bright signals in the video output to produce a linear response from the monitor. This intentional incorporation of nonlinear gain is called gamma correction. The disadvantage of gamma correction for dark signals is that the noise is also amplified. Gamma correction is seldom used for cameras used in image processing applications.

2.2 Low–level Vision

The digital image generated by the camera can be considered to be a two dimensional function $f(x, y)$. In order to restrict the memory requirements, it is

customary to represent images as an integer function of integer variables. Such an image with function values between 0 and 255, known as gray–levels, is shown in Figure 2.3. This figure is a daytime image of a runway taken by a camera

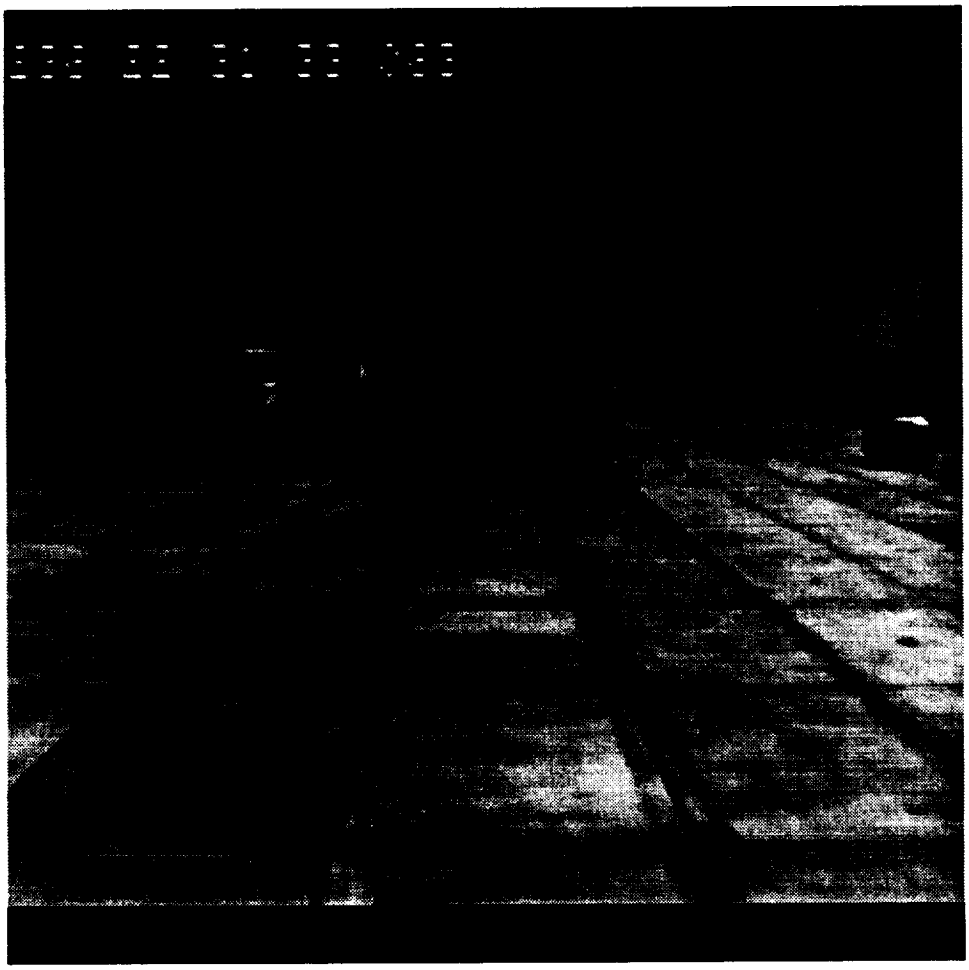

Figure 2.3: Vehicles parked on a runway.

mounted on the nose of a rotorcraft. Considering the image as a two–dimensional signal permits the application of various signal processing techniques. Techniques such as low–pass, band–pass and high–pass filtering, histogram equalization, and interpolation can all be used. Mathematical tools of transform theory such as two–dimensional Fourier transform, sine transform, cosine transform, singular value decomposition, and Radon transform can all be applied to enhance the information content in an image. Ideas from the theory of vector spaces can also be applied if an

image is conceptualized as a matrix. Images can also be processed using stochastic signal processing tools such as covariance models and autoregressive models. Application of some of these techniques are discussed at length in Reference [53].

2.2.1 Filtering

As an example of low-level vision processing, a low-pass filtered version of Figure 2.3 is shown in Figure 2.4. It is hard to tell the difference between this

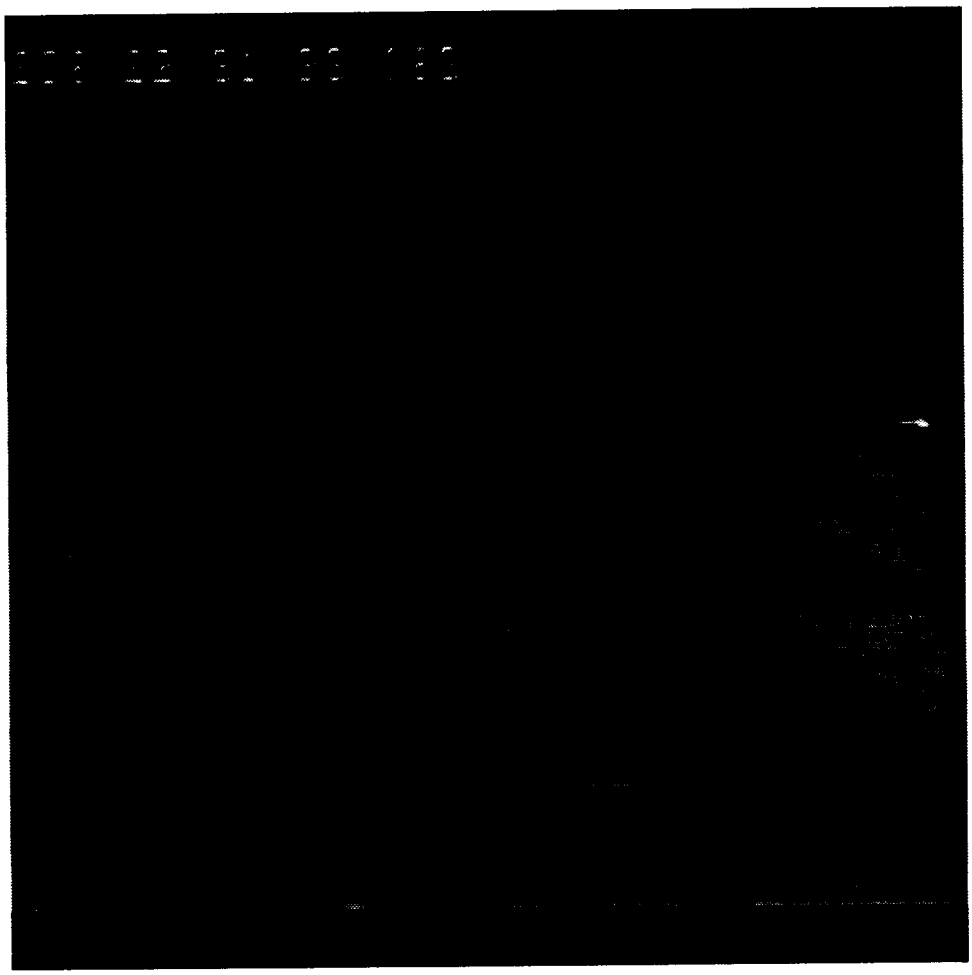

Figure 2.4: Low-pass filtered image.

image and the original image given in Figure 2.3, except for a slight reduction in the gray-level bandwidth. However, the difference is clear when examined in the

histogram domain shown in Figure 2.5. The histogram summarizes the frequency with which a certain gray-level appears in an image. Comparison of the two histograms in Figure 2.5 reveals that the gray-levels in the image shown in Figure 2.4 vary much more smoothly.

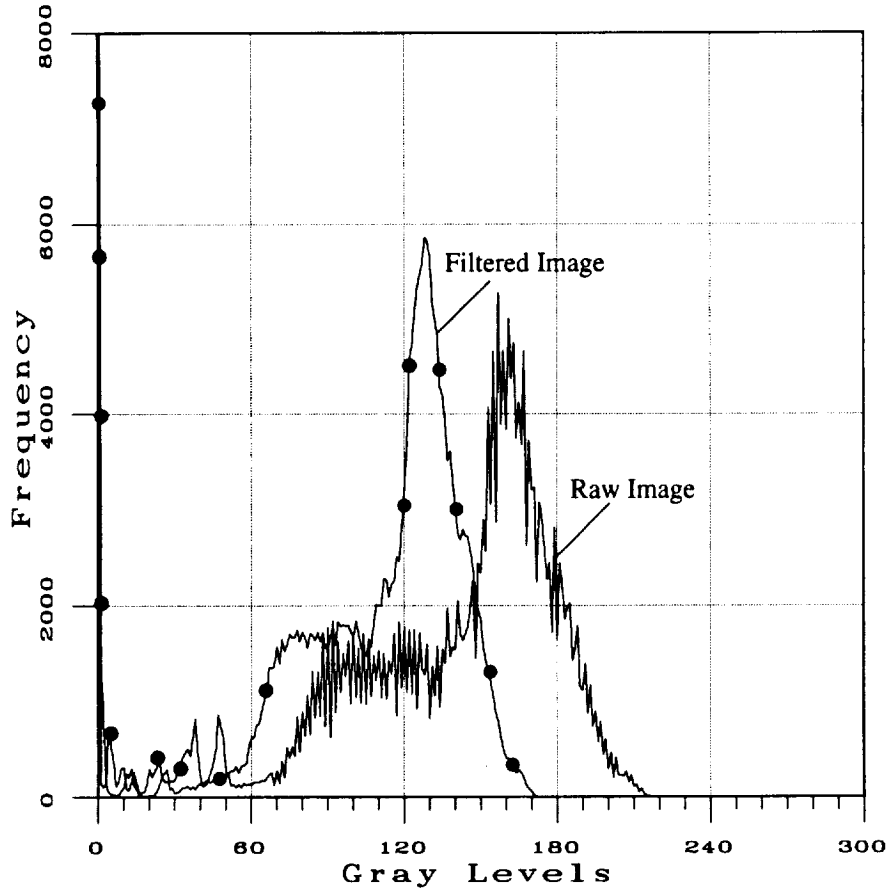

Figure 2.5: Histograms of original image and low-pass filtered image.

2.2.2 Edge Detection

An important early vision processing function is the edge detection. Edges in an image occur at locations of large gray-level changes. These changes can be characterized as step, ramp or a combination of step and ramp functions. Rather

than attempting to locate the edges directly from the gray-level image, a gradient operation followed by the thresholding operation is usually employed for edge detection. Edge operators lie in the following three classes: (1) operators that approximate the mathematical gradient operation, (2) template matching methods that use many templates at different orientations, and (3) operators that use parametric edge models for fitting local intensities [6]. Machine vision literature abounds with edge operators [78]. An example of edge operation on the image in Figure 2.3 is given in Figure 2.6. In this case a Sobel edge operator [6] was used.

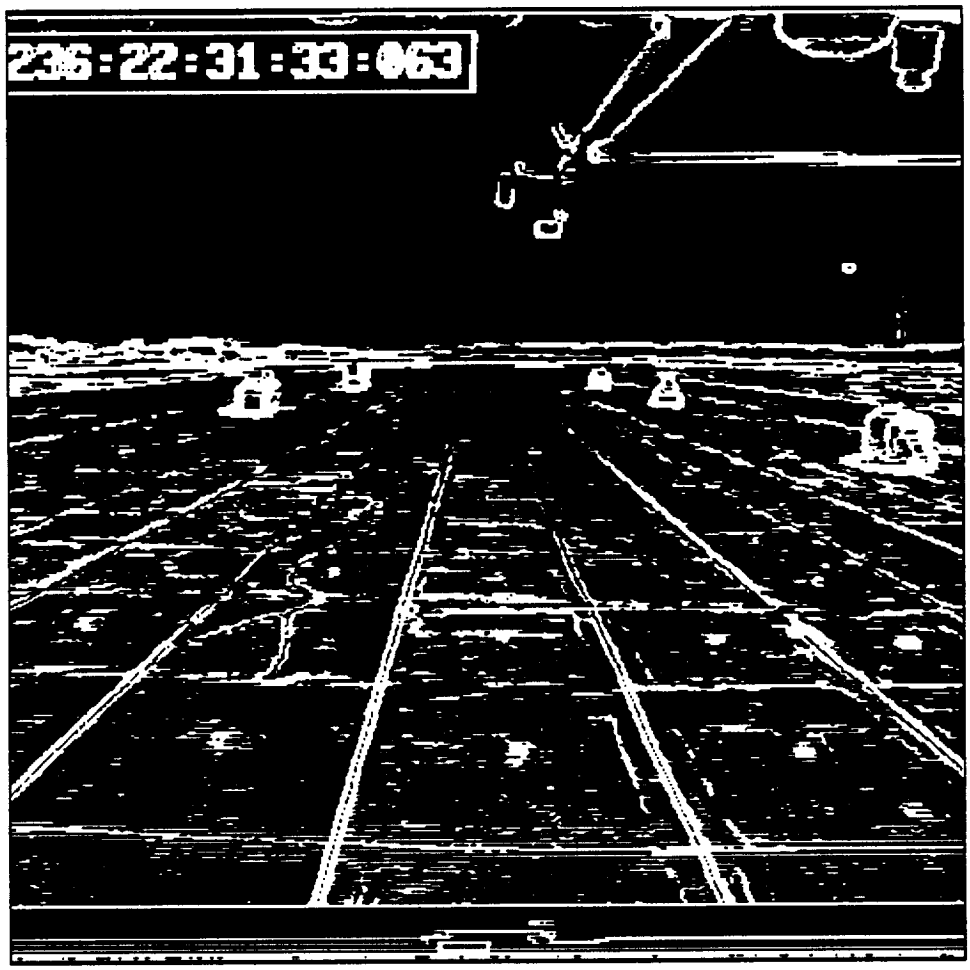

Figure 2.6: Sobel edge magnitude for Figure 2.3.

The Sobel edge operator is defined as:

$$\Delta_u = 2(f(i+1,j) - f(i-1,j)) + (f(i+1,j-1) - f(i-1,j-1))$$

$$+ \quad (f(i+1,j+1) - f(i-1,j+1)) \tag{2.1}$$

$$\Delta_v = 2(f(i,j+1) - f(i,j-1)) + (f(i+1,j+1) - f(i+1,j-1))$$
$$+ \quad (f(i-1,j+1) - f(i-1,j-1)) \tag{2.2}$$

with magnitude:

$$s(i,j) = (\Delta_u^2 + \Delta_v^2)^{\frac{1}{2}} \tag{2.3}$$

and direction:

$$\chi(i,j) = \tan^{-1}(\frac{\Delta_v}{\Delta_u}) \tag{2.4}$$

Here $f(i,j)$ is the gray-level at a pixel location (i,j). The other indices refer to the eight neighboring pixels surrounding this pixel. Figure 2.6 shows the thresholded edge magnitude. The edge direction from Equation (2.4) is illustrated as an image in Figure 2.7. This pseudo image is created by quantizing and scaling the edge direction in the range of zero and 255. Black corresponds to the vertical direction and white corresponds to the horizontal direction.

2.3 Higher–level Vision

Higher–level vision algorithms address the problems related to object representations in a scene. They include boundary detection, segmentation, grouping, geometric modeling, inference techniques and ranging.

2.3.1 Boundary Detection Methods

Boundary detection methods usually fall into one of the following categories: searching near an approximate location, Hough transform, graph searching, dynamic programming and contour following [6]. There is an abundance of literature describing these methods [2, 8, 36, 41, 62, 65, 72].

Searching near an approximate location involves the determination of a likely location of a boundary, which is then used for guiding the refinement of the boundary. One of the methods described in Reference [6] carries out local searches at

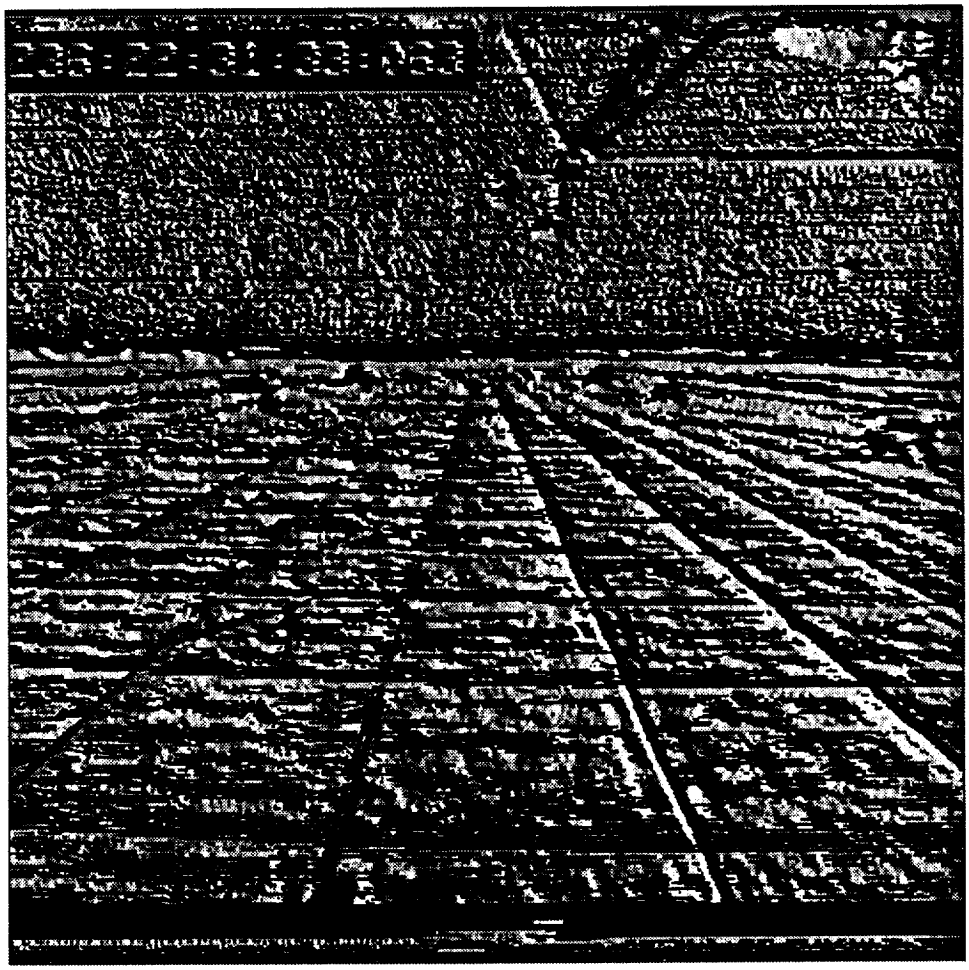

Figure 2.7: Graphical representation of Sobel edge direction for Figure 2.3.

regular intervals along directions normal to the initial boundary. In order to refine the boundary, the edges with the highest magnitude whose orientations are nearly tangential to the initial boundary are approximated using a polynomial. Recursive techniques that construct a boundary by first connecting two edges with a straight line and then searching along the normal at the central point for an edge that exceeds some preselected threshold have also been reported. This technique is then applied to the two segments formed by three edges and so on. Thus, a curved boundary is found.

Hough transform [6, 27] can be used if little information is available about the location of the boundary but its shape can be described as a parametric curve. To illustrate the method, consider a straight line in the parameterized form: $\rho = x\cos\theta + y\sin\theta$ where, θ is the angle of the normal and ρ is the distance from the origin [27]. Given a set of points (x_i, y_i), Hough transform involves setting up a two–dimensional accumulator array $A(\theta, \rho)$ which is incremented each time the particular (θ, ρ) location is visited. θ is quantized and varied between 0 and 2π. Hence for each (x_i, y_i), ρ's are computed using the parameterized form and the accumulator array is incremented by one. If many points lie on a straight line corresponding to a particular θ and ρ, the accumulator value for this θ and ρ is high. Thus by using a threshold, meaningful lines in the image can be determined. As discussed in [6] Hough transform method can also be tailored for other shapes.

Graph searching techniques involve searching through a set of nodes linked via edges to determine minimum cost paths for boundary determination. Minimum spanning tree algorithm described in [75] is one such graph searching algorithm. A spanning tree is defined as a connected set with no loops that contain all the points in the problem. The minimum spanning tree of a set is that whose cost is a minimum. On a historical note, the graph search problem is closely related to the travelling salesman problem in Combinatorics [59]. Several cost functions that can be used for boundary search are described in [6]. Heuristic graph search techniques and methods for managing the data structure are also described in [6].

The dynamic programming procedure can also be used for boundary detection. This procedure builds paths from multiple starting points in a discrete region using a performance index that describes the optimal boundary. A recent application of the dynamic programming procedure for boundary detection is described in [2]. "Energy" is used as the performance index in that work. The energy consists of image intensity, edge magnitude, curvature and smoothness constraint. In this formulation, a penalty is imposed for moving away from the initial contour position.

The central idea behind contour following is to start at an edge and develop a boundary by recursively adding neighboring edge elements based on their edge magnitude and direction. These methods make use of several heuristics. Recent methods that implement this idea are described in [8] and [62].

The boundary detection procedure proposed in [8] attempts to modify parameters of lower level processes such as edge contour tracking using higher level processes such as corner detection. The method encodes each edge element by its relationship to its neighbors using a chain code scheme. A window is then used to determine if the neighbors extend the edge in a straight line. If the neighbors do not extend the edge in a straight line, left and right extensions are examined. This process is continued till either the contour is closed or all the pixels have been examined.

A three step edge detection process is described in [62]. The first step involves computing the gradient magnitude and direction. The direction of the gradient is discretized to one of the eight neighbors surrounding the pixel. A heuristic concept of Likelihood-of-Being-an-Edge (LBE) is introduced as the second step. The third step is a contour following process which attempts to propagate the edge in a direction normal to the gradient direction starting at pixels with the maximum LBE value. The boundaries detected by application of this algorithm to the vehicle image is shown in Figure 2.8.

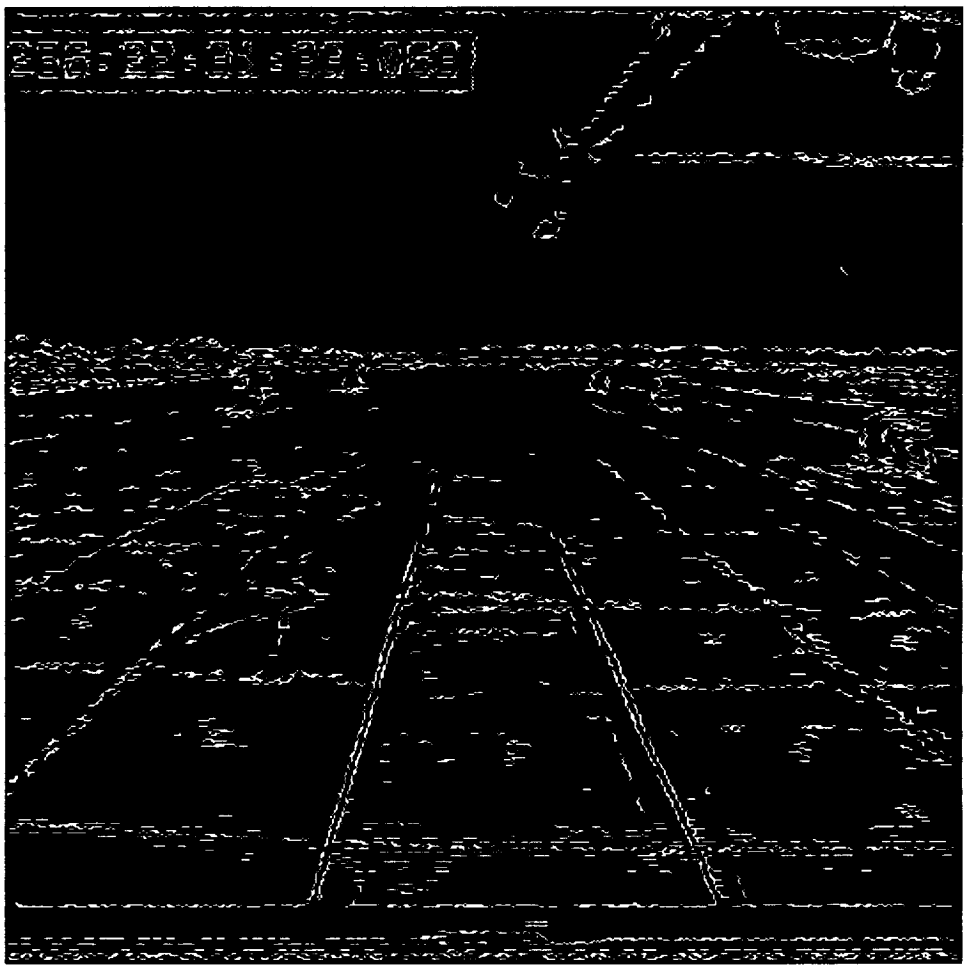

Figure 2.8: Edge boundaries for Figure 2.3 using the three–step process [62].

2.3.2 Gray–Level Segmentation

Segmentation methods serve to consolidate the information available in the image. Segmentation methods either work directly with the gray–level image or with texture properties. The central idea is to determine contiguous regions within the image that satisfy some homogeneity property. Methods reported in the literature can be broadly classified into local techniques, global techniques and splitting and merging techniques [6].

Local techniques attempt to place pixels in a region based on some homogeneity property of a pixel and its neighbors. An example of this technique is blob coloring given in [6]. The technique involves scanning the image from left to right and top to bottom with a special L–shaped template. The idea is to grow the blob by adding a neighboring pixel if its gray–level is approximately equal to the gray–level of the blob.

An example of a global segmentation technique is Thresholding. The idea here is that if an image consists of a background and an object, or equivalently, if the gray–level histogram of the image is bi–modal, a single threshold can be identified for segmenting the image into background and object regions. A more recent algorithm that extends this idea by using multiple–level thresholding is described in [57]. The difficulty with this technique is that many regions are given the same label because groups of pixels in different regions of the image lie within the same portion of the gray–level histogram. This is to be expected because of the global nature of the algorithm. Clearly, this algorithm is suitable if several objects with similar gray–level properties are expected to be seen against a common background in the image. This would be the case in nighttime images of the airport lighting.

Segmentation methods based on merging or bottom up, splitting or top down and a split and merge scheme are discussed in [50]. Merging involves labeling of adjacent regions into a larger region based on similar gray–levels while splitting involves re–labeling a larger region into several smaller regions based on the dissimilarity of the gray–levels in the original larger region. A split and merge

technique uses both splitting and merging operations. Usually merging operations are computationally efficient when compared with splitting operations. On the other hand, it requires larger memory compared to the splitting scheme. The split and merge scheme trades off computational speed for reduced memory requirement when compared to a pure merging scheme. Usually the split and merge scheme such as in Reference [50] is initiated at an intermediate level, close to the final segmentation. In this algorithm, the image is examined at various resolutions. Thus, four neighboring regions are merged if the difference between the maximum and the minimum gray-levels is less than a preset threshold. Similarly, a region in which the difference between the maximum and the minimum gray-levels is greater than a preset threshold is split into four subregions. Since both split and merge operations are done simultaneously, regions that are split are not merged with adjacent regions. To overcome this difficulty, a grouping technique which abandons the tree structure is used. Finally, the remaining small regions are merged with the adjacent large regions. For the vehicle image shown in Figure 2.3, 214 regions found by this algorithm are shown in Figure 2.9. The artifacts of the segmentation boundaries can be seen in this segmented image. Clearly, the power of segmentation is also illustrated by the fact that 262144 pixel regions are compressed into 214 regions.

Although both boundary detection and segmentation are related, it may be noted that the results generated by these methods are quite different as evidenced by Figures 2.8 and 2.9. It is easily seen that not all boundaries are closed but all segmented regions are contiguous.

The foregoing techniques are also applicable to the problem of texture segmentation with the homogeneity criteria based on texture properties. Since texture segmentation forms a large body of work in machine vision literature, it is treated here separately from gray-level segmentation.

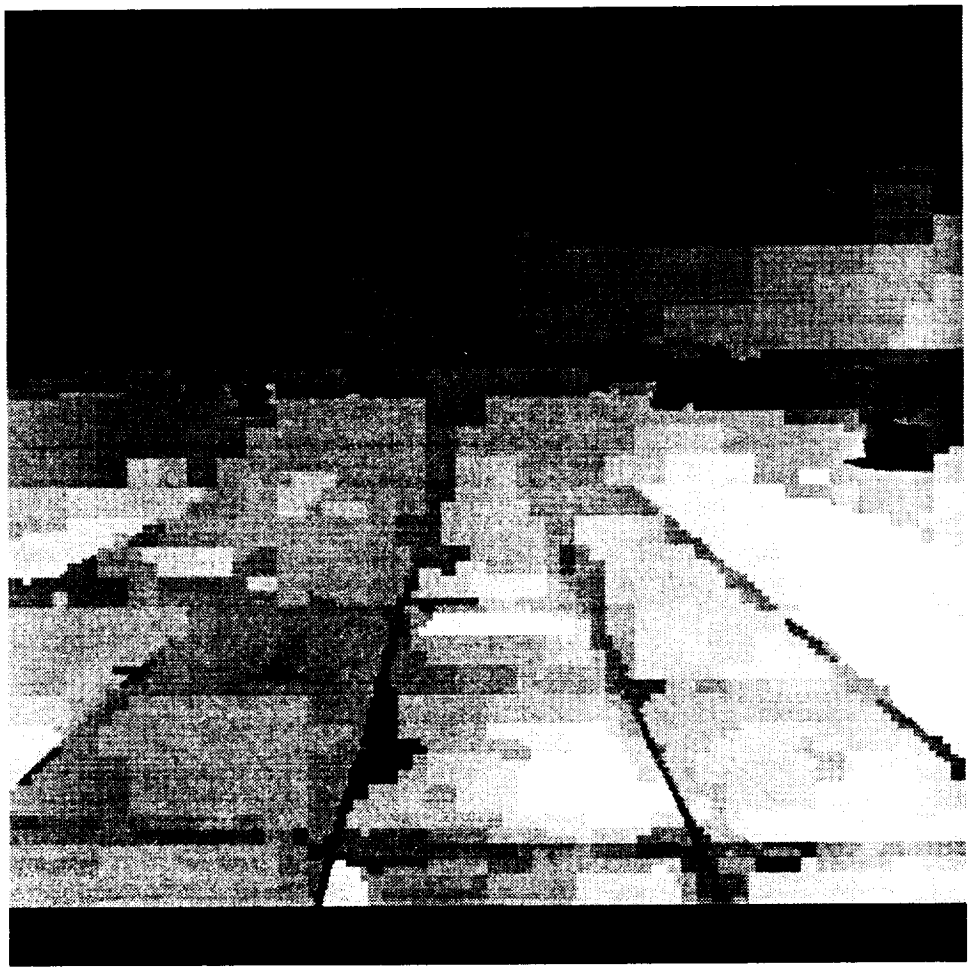

Figure 2.9: Segmented regions for Figure 2.3 using the split and merge scheme [50].

2.3.3 Texture Segmentation

Gray–level images are characterized by pixels of varying intensity. Hence, they can also be described by the stochastic properties of the gray–levels distribution across the image. The properties of this distribution are usually given in terms of the first, second and higher order statistics. First order statistics describes the pixel population in the image without regard to its spatial distribution. The second order statistics take the spatial distribution into account. Two approaches are used to characterize this spatial distribution: (1) a stochastic model–based approach and (2) a data–driven approach. The model–based approach assumes that the image can be modeled in terms of a two–dimensional random field. Several stochastic models are discussed in References [43, 79].

The data–driven or non–parametric approach is based on characterizing the two–dimensional intensity distribution by different types and features of second order statistics. The conditional probability density function $f(i,j|d,\theta)$ represents the probability that two pixels separated by an inter–pixel distance d and orientation θ have intensities i and j. An estimate of the conditional probability density function $c(i,j|d,\theta)$ is sometimes referred to as the gray–level co–occurrance matrix (GLCM) or as the spatial gray–level dependence matrix (SGLDM). SGLDM has been most widely used measure for classification of textures [1, 19, 39, 42, 101]. SGLDM can be obtained by computing the two–dimensional histogram of the frequency of the joint occurrences of two pixels with a fixed displacement and orientation with respect to each other having intensities i and j respectively. A rotationally invariant SGLDM is computed by averaging the individual SGLDM for each angular direction.

Either matrix features or scalar features can be used for texture classification. Many different approaches are available for texture classification using matrix features. Threshold selection based on the SGLDM is described in Reference [1]. In Reference [19], the SGLDMs of four neighbors in the quad–tree are compared with a threshold for merging or splitting operations. Results using this technique are

also presented in [103]. A technique for image segmentation by detecting clusters in the SGLDM, which correspond to the regions and boundaries in the image, is described in [39]. A maximum likelihood texture classifier using matrix and scalar features is examined in [101]. In Reference [77] segmentation is carried out by thresholding. The threshold levels are selected by projecting the off-diagonal elements of the SGLDM onto the diagonal and treating the resulting vector as a histogram. Although these methods are useful for segmentation, their storage requirements are high due to the use of matrix features. For example, 256×256 locations are needed to store a matrix feature for an image containing 256 gray-levels. These methods are also computationally intensive. The storage requirements and computational speed are the motivating factor for considering scalar features for image segmentation. However, it should be noted that many of the scalar features derived from the matrix features may not contain all the important texture information contained in the matrix features [21].

Several scalar features are derivable from the matrix features. For example, 14 scalar texture features based on the SGLDM are presented in [42]. For each of the scalar features, their means and variance computed by using the SGLDMs corresponding to the four directions, may be used for texture classification. Some scalar features derived from SGLDM, Fourier power spectrum, Gray-level difference statistics and Gray-level run length statistics are described in [21, 102]. Scalar texture features derived from the SGLDM may also be computed from sum and difference histograms [99]. Compared to computing the full SGLDM, sum and difference histograms are computationally fast and require significantly reduced storage. Except for the two scalar features energy and entropy, all the other scalar features can be obtained by using the sum and difference histograms. Methods such as [19] and [101] can be used for classification using scalar features. Additional methods such as piecewise linear discriminant function method, min-max decision rule method [42] and Fisher linear discriminant technique [102] can also be used for classification using scalar features.

Some of the scalar features relate to specific characteristics in the image such as homogeneity, contrast and organized structure. Other features characterize the complexity. Even though each scalar feature contains textural information, it is hard to identify which specific textural characteristic is represented by which feature. In Reference [93], the classification characteristics of scalar features derived from SGLDM are examined. It is shown that scalar features used in combination result in superior image segmentation when compared with a single scalar feature. This completes the discussion of the various tools and techniques available for image segmentation.

2.3.4 Clustering Methods

The need for clustering occurs naturally in many systems. For example, vision based range computations [70, 94] often result in a sparse set. A vision based ranging method described in Reference [94] is able to compute ranges at discrete locations shown as white squares in Figure 2.10. Scene understanding, navigation and display functions require these discrete set of ranges to be grouped into sets which correspond to objects in the real world. Clustering techniques can be used for grouping the discrete range points, varying from a few hundreds to several thousands, into a small number of objects in the scene.

Clustering [4] has been used for a long time in disciplines such as biology, geology and psychiatry. In computer vision, clustering methods have been used for classification of multispectral data and image segmentation using attributes like gray-level, color, texture, gradient, and range. The main idea behind clustering or grouping is similar to segmentation in the sense that both the techniques attempt to partition a given set into subsets based on discriminants. In computer vision, clustering has been associated with statistical pattern recognition using discrete samples as in Reference [27] while segmentation has been associated with partitioning the image into homogeneous regions as in Reference [6]. In this section, clustering is described as a problem of partitioning discrete data with the range

Figure 2.10: Range locations in the image from Reference [94].

map given in Figure 2.10 as an example.

Clustering techniques can be broadly classified into supervised (model–based) and unsupervised (data–driven) methods. Supervised methods require labeled training samples. For example, if a mixture is known to be composed of samples from two Gaussian distributions and the problem is to separate the two types, known samples from each Gaussian distribution can be used for estimating the mean and the variance of the Gaussians. Thus, a threshold or decision boundary can be found for classifying an unknown sample into one of the two types. Euclidean distance and Mahalanobis distance [27] measures can also be used for classifying the unknown sample to the type represented by the closest mean [27]. In the case of unsupervised clustering, the structure is directly obtained from the data. However in order to design a reasonable classifier, assumptions are invariably needed. For example, one may assume that the mixture is composed of two Gaussians even though their means and standard deviations are unknown. To estimate the means and the standard deviations, additional assumptions will have to be made before data can be utilized for obtaining the Gaussians. A "K–means" algorithm described in Reference [37] can be used for this purpose.

Of the two broad categories of clustering methods, unsupervised clustering is more useful in practice. This is due to the following factors: (1) for certain problems it is not easy to label the training samples due to their size, (2) the clusters can undergo small changes, and (3) very little is often known about the structure of the data. One of the ways of discovering structures in the data is by constructing a weighted graph. Distance relationships in the graph can then be used to partition the graph into sub–graphs to further improve the distance relationships. The graph–theoretical method described in Reference [104] uses a minimum spanning tree to partition the set of points into perceptually organized clusters. The perceptual organization is defined by the principles of proximity, similarity and continuity.

As an example of unsupervised clustering, Reference [95] describes a hierar-

chical clustering method for grouping discrete ranges for the scene shown in Figure 2.10. The technique described there first represents the range histogram as a sum of Gaussian. Next, the features are grouped based on separation in the horizontal plane. Finally, an algorithm based on the minimum spanning tree (MST) [104] is used for grouping the range points based on the separation in the image plane. The results from unsupervised clustering for a sample scene is shown in Figure 2.11. It can be observed from the figure that the method generated six clusters.

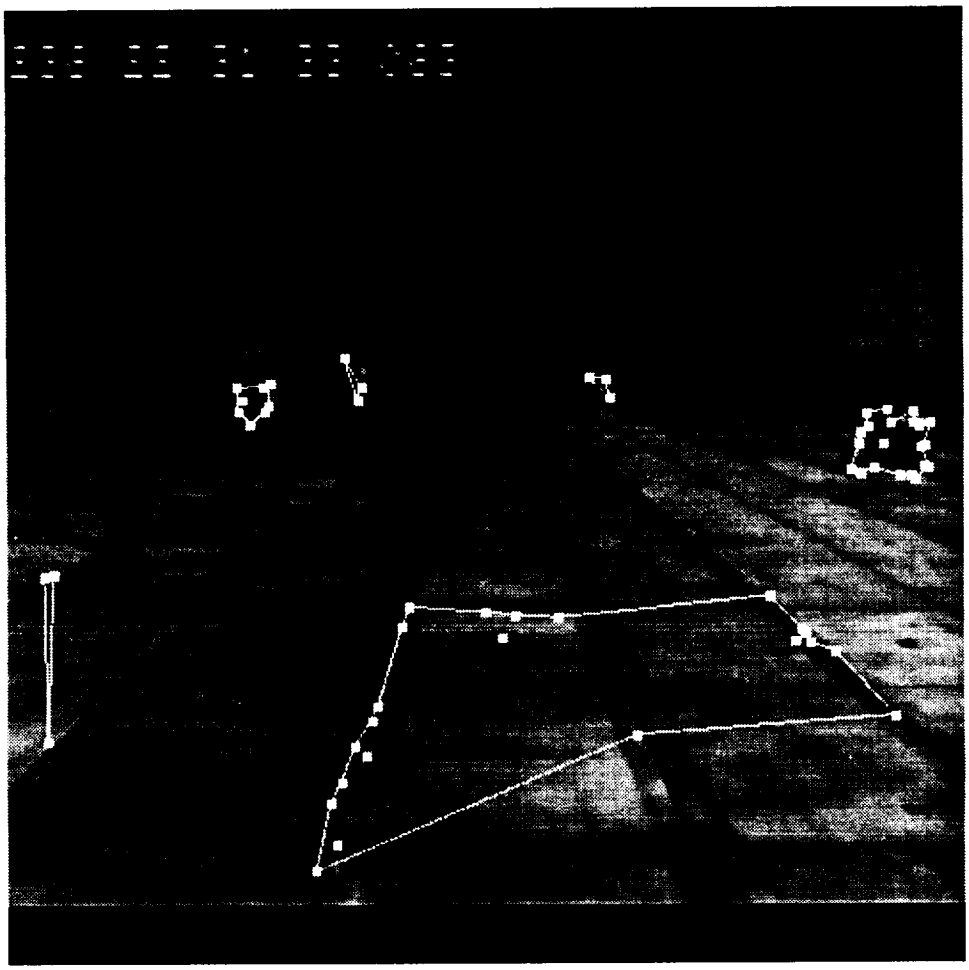

Figure 2.11: Groups in the image using unsupervised clustering.

Another way of addressing the clustering problem is to cast it as a discrete optimization problem which minimizes a certain distance function. Distance functions such as within-cluster and between-cluster distance measures based on the

scatter matrices are discussed in [27]. Since the set of features or points is finite, there can only be a finite number of partitions. Thus, in theory, the clustering problem can always be solved by exhaustive search. However, in practice, such an approach is not feasible because there are approximately $c^n/c!$ ways of partitioning a set of n elements into c groups. Due to this reason, the approach most frequently used is that of iterative optimization.

In Reference [14] application of the Monte–carlo methods for clustering range points into objects is described. It should be noted that these methods guarantee local but not global optimization [27]. Despite these limitations, the fact that computational requirements are reasonable make these approaches desirable. A technique based on Simulated Annealing for refining the initial grouping is described in [51]. The initial grouping in this case is obtained by assigning the range points to image regions obtained by labeling a segmented image.

2.3.5 Geometric Modeling

The clustering of discrete range points enables one to assume the range to be continuous within a group. It is possible to subsequently create a dense range map via interpolation within the groups. Modeling of dense range images has been studied by several authors [10, 58, 61]. The dense range images can be modeled into objects by fitting surfaces using polynomials, splines [74], Delaunay triangles [23] and other mathematical surfaces.

Several different approaches for representing surfaces defined by a set of randomly located points using triangular grids are described in [23]. These representations approximate the surface as a network of connected triangles with vertices at the data points. Many of the surface fitting algorithms use the properties of Delaunay triangles to discretize the domain with triangular elements. These algorithms may be broadly classified as incremental algorithms and divide–and–conquer algorithms. Incremental algorithms start from a boundary or interior point and create triangles by adding the remaining points. Divide–and–conquer algorithms

recursively split the set of data points into equal subsets until elementary sets are obtained, and then merge them pairwise. For example, application of the incre-

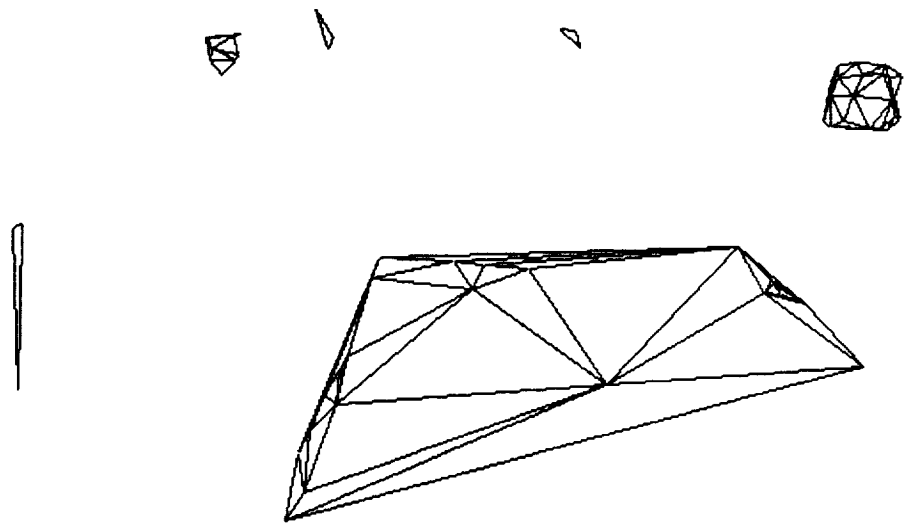

Figure 2.12: Surface representation of the groups in Figure 2.11 by triangular elements.

mental algorithm in Reference [64] to the clusters in Figure 2.11 yields a surface representation shown in Figure 2.12.

Once object modeling is accomplished by surface representation, additional geometric details can be extracted using surface interpolation. Several different element shapes and shape functions are discussed in the Finite Element Method literature [80, 105]. Some of these can be used for efficient interpolation. For the example shown in Figure 2.12, the interpolated range data is encoded as gray-

levels and presented in Figure 2.13. The figure also shows the modeled ground

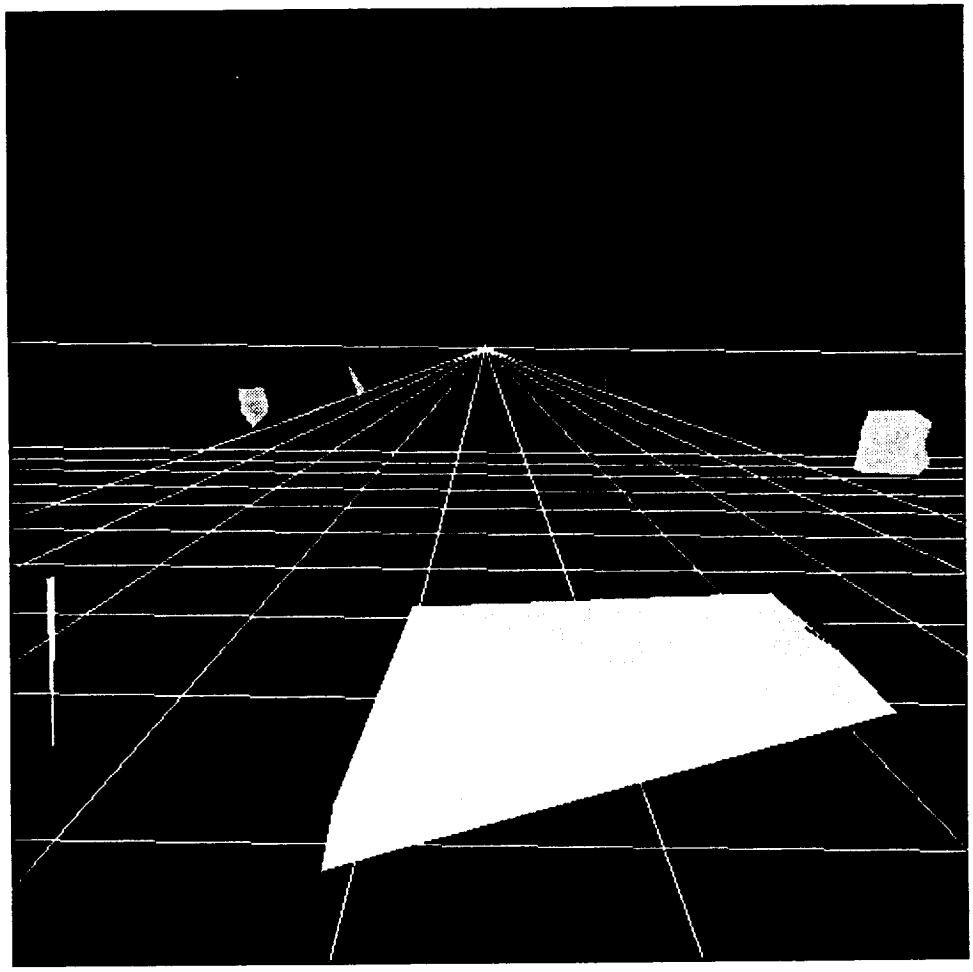

Figure 2.13: Finite element object models for surfaces in Figure 2.12.

plane.

For ground plane modeling, a Least Squares method can be used with the points from every group that are below a certain altitude. An assumption implicit in such modeling is that all the objects observed in the scene lie on a ground plane. A perspective projection of a rectangular grid on the ground plane can then be created to aid visualization. This process requires knowledge of camera altitude, and pitch and roll angles with respect to a local horizontal. Using the relative geometry of the camera with respect to the ground, the locations of horizon and the vanishing point can be obtained. The relationships needed for obtaining the

ground plane representation are described in [27] and [44]. The representation of ground plane using a grid projection is shown in Figure 2.14. The locations of the horizon and the vanishing point are also shown in the figure. The grid size of 12.5

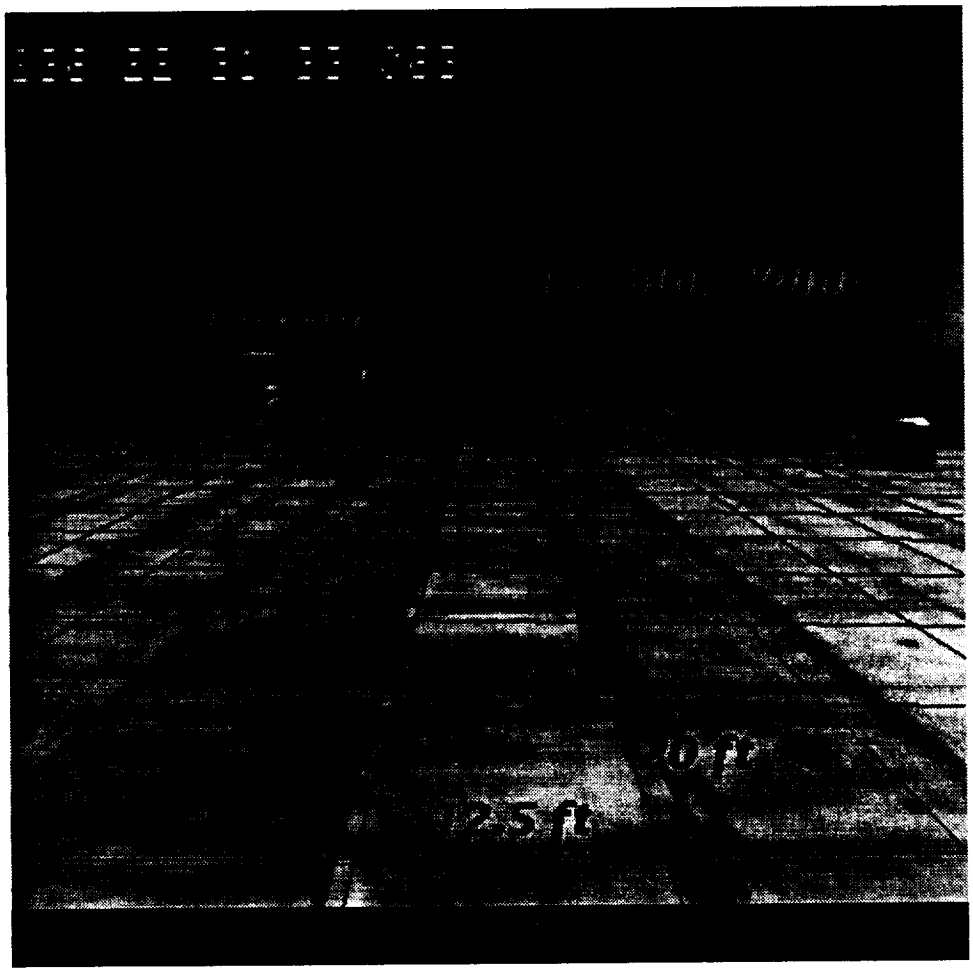

Figure 2.14: Ground plane representation.

feet by 20 feet was used in this case.

For the example presented above, inference is direct once the scene is assumed to be a model of a plane with objects lying on it. Geometric modeling directly yields the orientation of the plane with respect to the local horizontal, and the size, distance and shape of the objects with respect to the camera. Note that general scene understanding is much more complex and requires sophisticated inference techniques.

2.3.6 Inference Techniques

In a vision system, once the primitives or features are derived and classified, rules of inference can be used for object recognition. Many vision systems implicitly assume object models in order to aid the object recognition process and to develop an understanding of the scene. Comparative studies of several model-based object recognition algorithms are discussed in References [12] and [20].

Current model-based object recognition systems have several limitations. One of them is the difficulty in representing and describing objects. Only simple objects can be recognized by matching two-dimensional features with two-dimensional object models. The non-availability of higher dimensional features restrict the recognition capabilities to few object classes viewed in a particular way. A more general system will require the ability to extract three-dimensional features that are view point independent and match them with three-dimensional object models. Another difficulty is the non-availability of descriptors of surface properties of objects. Three issues that a model-based object recognition system has to deal with are: (1) design of features that describe physical properties and their spatial relationships, (2) a meaningful representation of the feature vector for an object class and (3) matching between the feature vector and object models for object recognition in a general scene [20].

The discussion of model-based object recognition with the background of algorithms and processes discussed in the previous sections indicates that it is difficult to design a general purpose vision-based object recognition system, and that a sequence of several low-level and high-level vision techniques are needed. Finally, without a model, the task of object recognition is virtually hopeless. In view of these observations, model-based techniques appear to offer the most direct scene interpretation without using elaborate inference techniques.

2.3.7 Ranging

Recovery of three–dimensional geometry from two–dimensional images is based on the fact that the differences between the locations of objects in two or more images obtained from different vantage points is a measure of their range. The process of finding the same object in multiple images is known as the correspondence problem in the machine vision literature. Since real images have limited field–of–view and resolution, the concept of correspondence is effectively a correspondence hypothesis. The relative object displacement obtained by satisfying this hypothesis in the image plane is called disparity. Due to perspective projection from the three–dimensional scene to the two–dimensional image, the farther the object is from the imaging device, the less disparity it exhibits. Closer objects exhibit larger disparity. Many vision–based methods discussed in the literature compute the disparity, thus recover the range to objects in the scene [7, 11, 46, 66, 81, 90].

In the simplest case of stereo vision where a pair of images are acquired by two cameras separated by a baseline, range can be computed by triangulation. For example, consider the geometry in Figure 2.15. In this figure, a point object appears along the line connecting the camera centers at u_1 in one image and at u_2 in the other image. Let the distance between the camera centers be b and the camera focal length be f. The Azimuth angles with respect to the optic axes of the two cameras then are $\psi_1 = tan^{-1}(u_1/f)$ and $\psi_2 = tan^{-1}(u_2/f)$. Since the two angles and the base of the triangle are known, the lengths of the range with respect to the cameras can be computed. An equivalent calculation can be done using a single moving camera. In this case, motion establishes the baseline required for triangulation. This is also known as Cyclopean vision, inspired by the mythical single eyed monster in Homer's Odyssey.

Driven by the needs of helicopter nap–of–the–earth guidance problem, machine vision techniques for ranging has recently attracted significant research attention. There are two distinct classes of algorithms that determine range by satisfying the correspondence hypothesis. They are known as field–based and

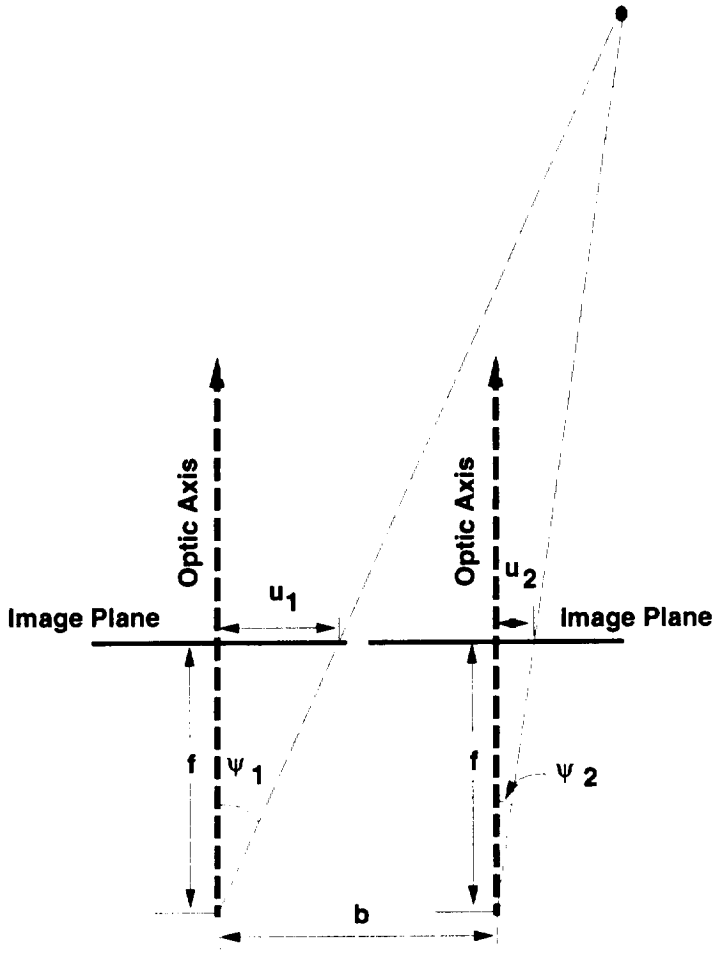

Figure 2.15: Range determination by triangulation.

feature–based methods. Field–based methods, such as [7, 46, 66, 81], assume a continuous variation of image intensity as a function of position or position and time. Feature–based methods, such as [18, 60, 76, 90], identify features in images, such as points, lines and contours in order to establish correspondence. Both, field–based and feature–based methods can only compute range to few locations in the image. Although field–based schemes have the potential of providing denser range maps, experience has shown [67, 68] that range can be reliably computed only at about 10% of the points in the image. This is due to the fact that computations break down in regions of near uniform brightness. Feature–based methods by their very nature can only compute range at discrete locations. An example of the range computations using a feature–based method is given in Figure 2.10.

Reference [66] describes a field–based ranging procedure using motion sequences generated by a single camera fixed to a moving vehicle. The method is based on the Optical Flow Constraint Equation of Reference [46] that relates the temporal partial derivatives with the spatial derivatives of the image function. Due to the use of partial derivatives, a smoothness constraint has to be enforced for the computation of range [46]. As discussed in Reference [66], incremental perspective projection equations can be directly combined with Optical Flow Constraint to yield a single navigation equation. This equation can then be used for obtaining the range. A crucial part of this method is evaluation of partial derivatives of the image function. In [66], the partial derivatives are estimated using a method based on the Calculus of Variations.

In Reference [67], a ranging scheme using image pairs is described. A multi–dimensional Taylor Series approximation of the correspondence hypothesis is used. The advantage of this method is that it does not require temporal partial derivative of the image function. Hence, this formulation does not require the concept of optical flow. In this method also, the incremental perspective projection equations are used with the Taylor series approximations to formulate a navigation equation. Since the navigation equation depends on the spatial partial derivatives,

the derivatives are computed using a finite difference scheme. Camera translation distance between the image pairs is used to perform range computations. No rotational motion is assumed.

To overcome the difficulties associated with derivative computation of noisy image functions using finite difference schemes, causal estimators that attenuate the noise in the process of derivative estimation are proposed in Reference [68]. This method offers the possibility of derivative computation during the image data collection process. Multi-dimensional Taylor series approximation of the correspondence hypothesis is used in this study also. Eliminating the disparities in favour of camera motion parameters and scene depth using the perspective projection equations, an Optical Ranging Polynomial is obtained. This polynomial is then solved to obtain the range. The algorithm has been demonstrated on a stereo image pair of a laboratory scene.

An extension of the previous algorithm that includes both translational and rotational displacements is described in Reference [70]. The central theme, including the technique for estimation of the partial derivatives, is the same as the previous algorithm. The algorithm has been demonstrated on an outdoor image sequence acquired by a camera mounted on the nose of a helicopter. The images are temporally separated and were acquired as the helicopter underwent both translational and rotational motion.

To overcome the difficulties of noise amplifying derivative estimation process, a derivative free ranging method is proposed in Reference [71]. In this algorithm, the correspondence hypothesis is approximated using Pade' approximation and used as a differential constraint in an optimization problem with a quadratic cost. The state variable is the sum of image functions of the two stereo images and the control variable is the range to objects seen in the images. The resulting necessary conditions for optimality are linear permitting the solution using the backward sweep method [13]. The method was demonstrated on a pair of stereo images of a laboratory scene.

Feature-based ranging methods are considered next. Given corresponding features in two images, the study in [90] describes how the estimated object location is influenced by the estimation algorithm and the relative geometry between the camera and the object. Three different Kalman Filter formulations are proposed for ranging in Reference [90]. These are: (1) inertial coordinate formulation (2) sensor coordinate formulation and (3) polynomial model for image point motion. The polynomial filter was found to be unsuitable for general camera motion. The methods described in Reference [90] assume that the motion parameters such as camera position, attitude, and translational and rotational velocity are available from an onboard Inertial Navigation System.

Research reported in Reference [86] develops a normalized correlation function based feature correspondence procedure. This technique forms the first step in the Kalman filtering algorithm. Feature detection is accomplished by using an edge operation and correspondence is achieved by using the gray-levels of the detected features. In addition, a recursive algorithm for range estimation based on translational motion is also described. Since translational motion is assumed, the search for correspondence is restricted to envelopes along radial lines eminating from the focus-of-expansion in the images. Results for a laboratory image sequence are obtained by using the recursive algorithm.

Details of the correspondence procedure when images are acquired from a camera undergoing general motion are described in Reference [88]. Thus, this work extends the procedure given in Reference [86] for more general motion involving translation and rotation. An elliptical search window based on the propagated range estimate is used to minimize the search effort. Results for a laboratory image sequence are described.

Results for an outdoor image sequence obtained with the Kalman Filter formulated in the sensor frame in Reference [90] are reported in Reference [89]. It is shown that good range accuracy is obtained for the objects in the field-of-view of the camera. This result is significant because it is much more difficult to establish

correspondence in the images taken from a rotating and translating platform, such as a helicopter in flight. Details of the image acquisition procedure are described in Reference [82]. The results of application of the motion algorithms on the laboratory and flight image sequences are described in [91]. That report summarizes the procedure and the results of References [88] and [89].

Since the objects in the field–of–view are at various ranges, It may be advantageous to use a different measurement rate in different portions of the image. This idea is explored in Reference [92]. The technique for range estimation involves accepting the measurement for the Kalman Filter only when the tracked feature moves more than a set threshold in the next image. Numerical results presented for the outdoor image sequence show that the multirate filter provides the same estimation accuracy as the standard Kalman Filter, with a significantly lower computational effort. Since different features are updated at different times, the book–keeping task is more involved when compared to the single rate filter implementation.

When range information is obtained using a single camera, it is sensitive to the direction of motion. Hence, the estimates are poor close to the focus–of–expansion. An analysis of motion and stereo methods is provided in Reference [87] to demonstrate that motion methods provide more accurate range information away from the focus–of–expansion and stereo methods provide superior accuracy close to the focus–of–expansion. In order to overcome the limitations of the stereo method, a recursive stereo method is described in Reference [87]. This method is then contrasted with standard stereo method and the earlier recursive motion algorithm [86]. It is suggested that an integrated stereo and motion method based on the recursive motion method and the recursive stereo method has the potential for providing more accurate range estimates when compared to either of the two methods.

A hybrid motion/stereo algorithm is described in Reference [83]. This algorithm is an extension of the recursive motion algorithm given in Reference [90].

The range predictions generated by the Kalman filter are used for constraining the search space for feature correspondence in the stereo and motion pairs. One of the advantages of the method is that the Kalman Filter can be initialized with the range estimates obtained by processing the stereo pair. Results of application to an outdoor image sequence shows that the hybrid estimates are an improvement over the monocular estimates. Both methods generate estimates which appear to converge to the true range over time.

These algorithms have been applied to numerous images of outdoor scenes obtained from helicopter-borne cameras. The results obtained using these algorithms have been verified against range data obtained by a laser range finder. These algorithms can be considered to represent a mature class of vision based ranging algorithms.

The following conclusions can be drawn based on the review of the field-based and feature-based methods: (1) correspondence of regions in one image to regions in another image is the most significant problem, (2) even in an unstructured scene, ranging algorithms can be made to work reasonably if the camera position and angular displacements are available from an independent source like an Inertial Navigation System, (3) inclusion of system dynamics in the design of a recursive state estimator leads to higher estimation accuracy, and (4) a hybrid motion-stereo method provides higher accuracy when compared to pure motion and stereo methods.

Egomotion or self motion problem is the dual of the ranging problem. In this case, the camera position and orientation are the unknowns to be determined. In order to solve the problem, it is often assumed that the objects in the field-of-view are stationary. Along with this assumption, if correspondence can be established between features in successive images, the change in camera position and orientation can be computed. To determine the absolute camera position and orientation, the location of objects in the field-of-view have to be known with respect to an inertial coordinate system. Thus, the solution of egomotion problem

requires an underlying scene model via the definition of an inertial coordinate system together with the locations of the objects with respect to this coordinate system. This suggests two possible approaches: (1) use of the vision based ranging algorithms to recover the absolute camera position and orientation, or alternatively (2) use of a model–based approach to directly recover the camera position and orientation.

Camera calibration problem [6, 27] is a special model–based approach in which the correspondence between the objects in the scene and the image are known. The objective of camera calibration is the determination of camera optical characteristics and the camera position and orientation. Intrinsic camera parameters are: focal length, lens distortion, scale factor, and center of the image plane. In order to calibrate the camera, a planar grid target is placed at a certain orientation and distance away from the camera, and an image of the grid is obtained. The position of every grid point on the plane is known with reference to an inertial frame. In addition, the correspondence between every grid point on the plane and in the image are known. Since every grid location of the plane is related to its image via the intrinsic and extrinsic parameters, these parameters can be computed using an iterative algorithm. The traditional approach is to use a nonlinear parameter optimization technique. An alternative two–stage technique for camera calibration is described in Reference [98]. This technique solves the problem by using the least–squares method. Only few parameters are computed using nonlinear search. Efficiency of the process can be greatly enhanced by generating the initial guess using the least–squares method.

If correspondence is unknown, the calibration technique is not applicable even with known intrinsic parameters. Model–based methods are useful in this case. A model–based method that uses local feature correspondence and a Kalman Filter is described in References [25] and [26]. In References [25] and [26], the initial camera position and orientation estimates are used together with the perspective projection equations for projecting known model features such as corners and curve

segments into the image plane. This results in the creation of the model image. These features are then identified in the actual image acquired by the camera. The position difference of the features in the actual image and the model image are used to drive the Kalman Filter to improve the camera position and orientation estimate. Locating model features in the actual image is the main limitation of the algorithm. The difficulty is caused by the fact that in general, the features in the actual image are significantly different than the features in the model image. Therefore, the search involves matching a considerably simplified model template with regions in the image. In contrast, the ranging methods reviewed earlier use a templete based on a previous real image. In order to work satisfactorily, model based matching requires the features to be invarient to scale and rotation. It is difficult to find such features in real scenes. If the scene is such that one feature cannot be distinguished from another, matching the features may be difficult because a model feature could potentially match with many image features.

2.3.7.1 Uniqueness of Solutions

Before embarking on the development of pilot aids based on machine vision techniques examined in this chapter, it is essential to address the question of uniqueness of the solutions. Machine vision literature [48, 73] poses the uniqueness problem as follows: given displacements and velocities of image points, under what conditions is it possible to recover the shape of the scene and the relative motion between the camera and the objects in the scene?

For differential motion, the research given in Reference [48] shows that ambiguities arise only in the case of certain hyperboloids of one sheet and their degeneracies, such as circular cylinders, elliptic cones, hyperbolic paraboloids, and two intersecting planes that are viewed from a point on their surface. The governing equation of hyperboloids of one sheet is [85]:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \tag{2.5}$$

where (x, y, z) is the coordinate of a point on the surface of the hyperboloid, and a, b and c are constants. For large motion, it is shown in Reference [28] that if only five points are available in the image, then up to ten solutions are possible. Research in Reference [97] shows that the solution is usually unique if the displacements of seven points in two successive images are known. The solution is non–unique only if these points lie on a cone passing through the origin or on two planes, with one plane passing through the origin.

More recently, it has been shown in [73] that only certain hyperboloids of one sheet and their degeneracies when viewed from a point on their surface can give rise to ambiguity. Moreover, Reference [73] shows that in the case of hyperboloids of one sheet and hyperbolic paraboloids, there can be at most three solutions. That work also demonstrates that in the case of intersecting planes and circular cylinders, there can exist at most two solutions. It is also pointed out that cones cannot give rise to ambiguity unless the motion is differential.

The next chapter will apply the machine vision techniques described in this chapter to develop pilot aids for night landing. Machine vision algorithms for aircraft position, velocity and attitudes with respect to the runway will be derived and evaluated in simulations. Data sources for these algorithms will also be identified. Past research examined in this chapter indicates that it may be possible to synthesize machine vision systems that produce unique solutions for pilot aiding during night landing. This conclusion is a consequence of the two facts: (1) the airport lighting layout is viewed from above and (2) the underlying lighting geometry is planar.

2.4 Summary

Since the algorithms for machine vision are subject areas of Computer Vision, a review of the literature relevant for the design for such algorithms was discussed in this chapter. Two broad classes of algorithms, low–level and higher–level vision

algorithms were examined. Several algorithms for low–level vision process of image conditioning and edge detection were outlined. Higher–level vision algorithms for boundary detection, gray–level segmentation, texture segmentation, clustering, geometric modeling, inference and ranging were also discussed. Results of applying several of the low–level and high–level algorithms to an actual image of a runway scene were presented. These results illustrate the types of information that can be derived using image processing algorithms. Since position determination is the central topic of this research, field–based and feature–based algorithms reported in the literature that are closely related to this work were described. Many of these methods required the position and orientation of the camera to be known. As a result, they are not directly useful for the runway relative aircraft position and orientation determination problem. The literature for camera calibration problem was also found to be inadequate because those techniques assume correspondence between the grid points on the calibration plane and the image plane. Since correspondence between the model of the runway scene and the image acquired by the camera is unknown a priori, template–based local feature correspondence methods were also found to be unsuitable. Finally, the question of uniqueness of solutions was addressed. Based on the available literature, it was established that a unique solution of the runway relative position and orientation could be found for the viewing geometry used in this research.

Chapter 3

Machine Vision Based Landing Aids

This chapter develops the basic building blocks for constructing the machine vision algorithms for aircraft runway relative position and orientation estimation. With this goal, the nature of the landing task and the accuracy requirements are examined first. Clearly, the specification of a runway fixed inertial coordinate system, body coordinate system and camera coordinate system are essential components of methods for aircraft position and orientation estimation. Aircraft equations of motion which relate the aerodynamic and propulsive forces and moments to the translational and rotational motion of the aircraft are then developed with respect to the body and inertial coordinate systems. An onboard pinhole camera model that relates the inertial location of an airport light to its location in the image plane is described subsequently. Finally, landing and image simulation procedures are described to tie these building blocks together.

3.1 Aircraft Landing Operation

Aircraft arrival flight to the destination airport can be broken up into two broad segments: en-route descent and final approach to touchdown. A host of

procedures have to be followed by the pilot during both these phases. During descent from cruise, the rate of descent and airspeed have to be controlled to comply with the restrictions imposed by the air traffic control system. The airspeed has to be reduced to 250 Knots Indicated-Airspeed or less, when below 10,000 feet MSL (Mean Sea Level) [24].

A published arrival procedure called Standard Terminal Arrival (STAR) is used to transition from the en-route structure to an outer fix or an instrument approach fix or an arrival waypoint in the terminal area. The aircraft is then flown to the final approach fix to intercept the glide slope. The optimum length of the final approach is five miles; the maximum length is ten miles [24]. Once on the glide slope, aircraft speed, rate of descent and certain altitude distance relationships are maintained until the aircraft is beyond the runway threshold and at a prescribed altitude. At this stage, the aircraft executes the flare maneuver to achieve a gentle touchdown.

During landing, the pilot controls the aircraft lateral displacement from the runway centerline, distance from the touchdown point, altitude, yaw-pitch-roll orientations, rate of descent and rate of closure with the touchdown point on the runway surface. The desired glide path which describes the altitude, time and distance relationships during a typical landing are shown in Figure 3.1. Figure 3.1 shows a commonly employed three degree glide slope approach. Glide slope, ν, is defined as:

$$\nu = \tan^{-1}(h/x_{go}) \tag{3.1}$$

where h is the altitude and x_{go} is the distance-to-go to the touchdown point. For precision approach, the glide slope is between 2.5 and three degrees at most airports [24]. The glide slope in conjunction with the location of the touchdown point specifies the desired aircraft position with respect to the runway threshold as a function of altitude. Optimal threshold crossing height is 50 feet but it may be as high as 60 feet or as low as 32 feet [24]. The touchdown point is specified in terms of the distance from the runway threshold. From Figure 3.1, it may be seen

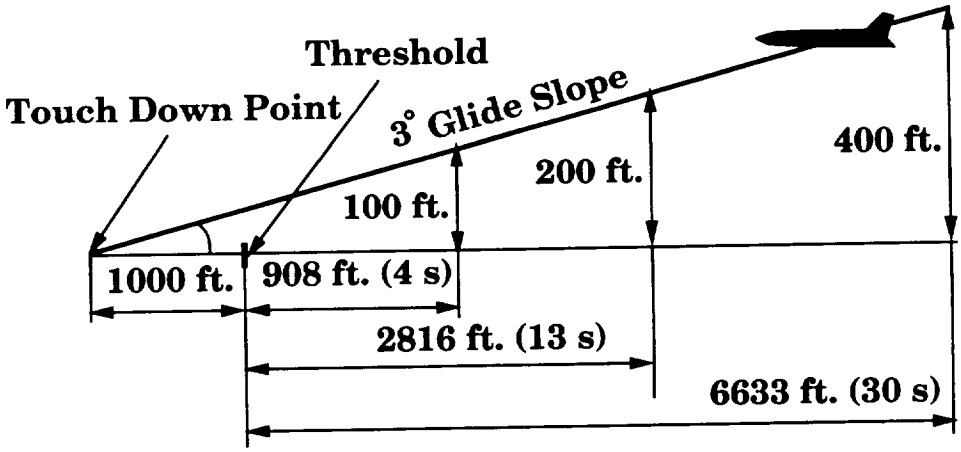

Figure 3.1: Glide slope, altitude, time and distance relationships.

that three degree glide slope requires the aircraft to be at distances of 6633 feet, 2816 feet, and 908 feet corresponding to altitudes of 400 feet, 200 feet and 100 feet respectively. These distances translate to 30 seconds, 13 seconds and four seconds to the runway threshold. Time–to–go calculations are based on a typical approach speed of 220 feet/second (130 knots).

The discussion of landing procedures is incomplete without mention of the abort procedures. Once the aircraft has passed the final approach fix, it is flown to the minimum descent altitude with enough time and distance remaining to identify the runway environment before continuing on the visual approach to the touchdown point. Descent below the minimum descent altitude is not authorized until visual reference with the runway environment is established and the aircraft is in a position to execute safe landing [24]. If it is unable to execute a safe landing, the aircraft is flown at or above the minimum descent altitude to the missed approach point. Subsequently, the aircraft is routed back to the outer fix for another landing attempt. Depending on the ground and airborne equipment, the decision to land can be delayed as the aircraft is flown along the glide slope. There are prescribed landing categories with associated decision heights up to which aircraft can be flown with instruments. Beyond the decision height for

the category of landing, it should be possible to fly the aircraft with just the visual reference. Like the final approach fix data, missed approach points are also published in navigation charts.

3.1.1 Landing Accuracy Requirements

One of the terms used for landing conditions is runway visual range (RVR). RVR is the distance from which the pilot can see the high-intensity runway edge lights. It is determined by transmissometer measurements near the threshold [49]. The transmissometer consists of a light source with a narrow beam projector and a receiver with a narrow beam acceptance angle. In order to make the measurements, these two components are raised to 15 feet above ground and separated by 500 feet. The amount of light received is a measure of atmospheric transmissivity. The measurements are compensated for the intensity setting of the edge lights and the time of day or night. For category II and III operations, two measurements are made. One near the threshold and the other near the midpoint of the runway. While useful, these measurements do not accurately predict the visibility along the approach path, since the measurements are made close to the ground.

Visibility on the runway is classified into I, II and III categories. Category III is further subdivided into a, b, c. The three categories are defined in terms of the RVR and decision height. Decision height (DH) is defined as the minimum height above the runway where a decision must be made by the pilot to continue descent to landing or to abort. The decision is based on the pilot being able to obtain visual guidance cues provided by airport lighting without depending on cockpit instruments.

The various categories and the associated RVR and decision heights are listed in Table 3.1 [31, 32]. Capability for automatic landing all the way to touchdown is required for all category III landings. For category IIIa, the rollout after landing and taxiing is manual. For category IIIb, an automatic rollout capability is additionally required. For category IIIc, an automatic taxiing capability is also

Table 3.1: Visibility Categories

Category	Decision Height (ft)	Visibility (ft)
CAT I	200	1800
CAT II	100	1200
CAT IIIa	50	700
CAT IIIb	$0 \leq DH < 50$	150
CAT IIIc	0	0

required, in addition to the landing and rollout capabilities.

Although Table 3.1 lists the RVR and decision heights for the various categories, it does not list the navigation accuracy requirements. The performance specifications for Federal Aviation Administration (FAA) defined precision approach and landing categories are given in Table 3.2 [100]. These accuracy requirements will be used to evaluate the performance of the machine vision based algorithms developed in this report.

Table 3.2: Aviation Navigation Accuracy Requirements

Category	Lateral (ft)	Vertical (ft)
CAT I	±56.1	±13.45
CAT II	±17.06	±5.58
CAT III	±13.45	±1.97

3.2 Coordinate Systems

Various coordinate systems used in this report are illustrated in Figure 3.2. In this figure, **i** is the origin of the inertial frame attached to the runway threshold. Since the location of all lights are given with respect to the threshold bar, it is a natural choice for the location of the origin of the inertial coordinate system. Furthermore, since the centerline lights form a principal axis of symmetry, following the flight dynamics convention, the x-axis of the inertial frame is aligned with the

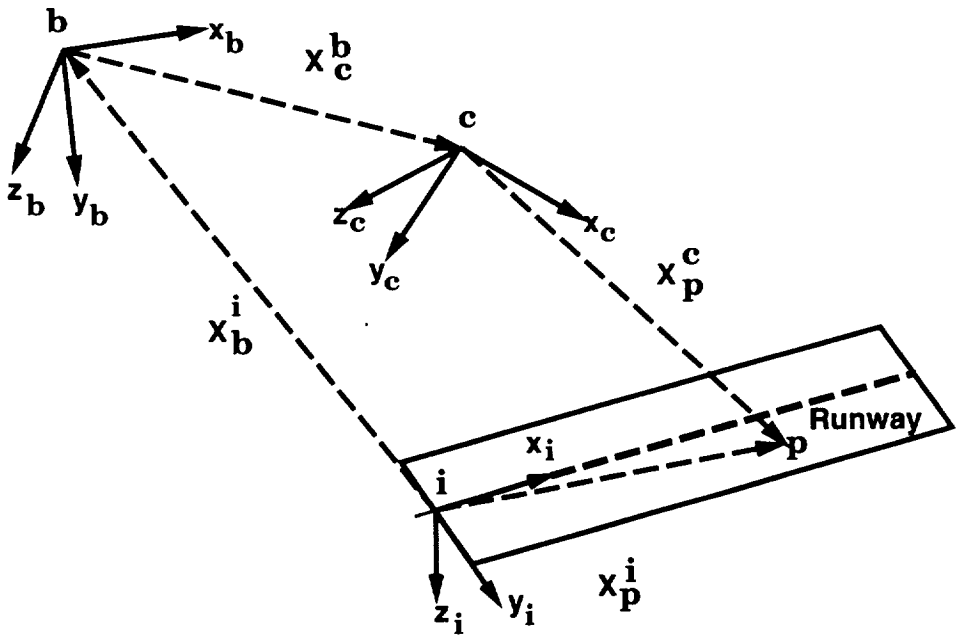

Figure 3.2: Coordinate frames.

runway centerline in the approach direction and the z-axis points down. The y-axis completes the right-handed triad. The origin of the aircraft body axes is located at the point **b**. Its position with respect to the inertial frame is given by the vector $\mathbf{X_b^i}$ with components x_b, y_b and z_b. The camera frame is located at **c**. The camera position with respect to the body frame is given by the vector $\mathbf{X_c^b}$ with components l_x, l_y and l_z. Since the camera is rigidly attached to the aircraft structure, the vector $\mathbf{X_c^b}$ is assumed to be constant in the present research. Let **p** be a light on the runway and let its position with respect to the inertial frame be given by the vector $\mathbf{X_p^i}$ with components x_p, y_p and z_p. Also, let the position of point **p** with respect to the camera frame be given by the vector $\mathbf{X_p^c}$ with components x_{cp}, y_{cp} and z_{cp}.

The position of the point **p** with respect to the aircraft in the inertial frame is given by the vector

$$\mathbf{X_p^i} - \mathbf{X_b^i} = [(x_p - x_b), (y_p - y_b), (z_p - z_b)]^T \tag{3.2}$$

The transformation matrix from the inertial frame to the body frame $\mathbf{T_{b/i}}$ can be

obtained in terms of the yaw attitude ψ, pitch attitude θ, and roll attitude ϕ as [96]:

$$\mathbf{T_{b/i}} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.3}$$

The position of point **p** with respect to the body frame can be obtained as:

$$\mathbf{X_p^b} = \mathbf{T_{b/i}}(\mathbf{X_p^i} - \mathbf{X_b^i}) \tag{3.4}$$

Similarly, the position of point **p** with respect to the camera frame is given by:

$$\mathbf{X_p^c} = \mathbf{T_{c/b}}(\mathbf{X_p^b} - \mathbf{X_c^b}) \tag{3.5}$$

Here, $\mathbf{T_{c/b}}$ is the constant transformation matrix from the body frame to the camera frame. Combining Equations (3.4) and (3.5):

$$\mathbf{X_p^c} = \mathbf{T_{c/b}}\mathbf{T_{b/i}}(\mathbf{X_p^i} - \mathbf{X_b^i}) - \mathbf{T_{c/b}}\mathbf{X_c^b} \tag{3.6}$$

Since the camera is assumed to be fixed with respect to the body, the product: $-\mathbf{T_{c/b}}\mathbf{X_c^b}$ is a known constant vector **k** with components k_x, k_y and k_z. Furthermore, if r_1 through r_9 are defined as the elements of the transformation matrix from the inertial frame to the camera frame, $\mathbf{T_{c/i}} = \mathbf{T_{c/b}}\mathbf{T_{b/i}}$, the components of the position vector $\mathbf{X_p^c}$ can be obtained as:

$$x_{cp} = r_1(x_p - x_b) + r_2(y_p - y_b) + r_3(z_p - z_b) + k_x \tag{3.7}$$

$$y_{cp} = r_4(x_p - x_b) + r_5(y_p - y_b) + r_6(z_p - z_b) + k_y \tag{3.8}$$

$$z_{cp} = r_7(x_p - x_b) + r_8(y_p - y_b) + r_9(z_p - z_b) + k_z \tag{3.9}$$

Equations (3.7) through (3.9) show the relationships between the location of the airport lights in the camera frame and the aircraft position and orientation. The aircraft position and orientation evolve due the to forces and moments acting on it.

3.3 Aircraft Dynamic Model

Aircraft is subjected to aerodynamic, gravitational and propulsive forces and moments. These forces and moments result in translational and rotational motion of the aircraft. Equations of motion relate the airplane motion to the forces and moments. Three coordinate systems are used to express the forces and moments in a convenient way. These are described first. Subsequently, equations of motion are presented with forces and moments expressed in these coordinate systems.

3.3.1 Coordinate Systems

Aerodynamic forces and moments on the aircraft depend on the orientation of the airframe with respect to the airflow. Since rotation around the free–stream velocity vector in a uniform airflow does not cause changes in the aerodynamic forces and moments, they depend only on two orientation angles with respect to the relative wind. These are the angle of attack, α and the angle of sideslip, β illustrated in Figure 3.3.

Figure 3.3 shows the body axes system with the x–axis aligned with the fuselage reference line, the z–axis in the aircraft plane of symmetry and orthogonal to the x–axis, and the y–axis normal to the plane of symmetry. The angle of attack and sideslip are defined by performing plane rotation about the body y–axis by α, followed by another plane rotation about the new z–axis by β such that the x–axis is aligned with the relative wind. The variables α and β are the angle of attack and angle of side slip respectively. The axis system resulting from the first rotation about the y–axis is often called the stability axis system.

With the angles of attack and sideslip defined by the axes systems, the transformation from the body to stability axes $\mathbf{T_{s/b}}$ is given by:

$$\mathbf{T_{s/b}} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.10}$$

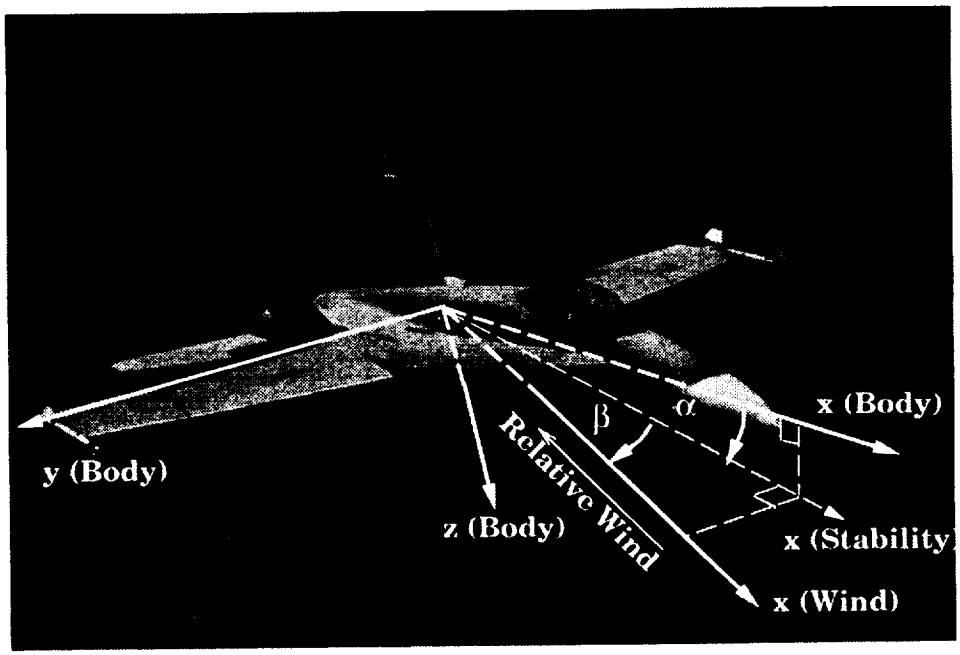

Figure 3.3: Aircraft axes and angles.

The transformation from the stability to the wind axes $\mathbf{T_{w/s}}$ is given by:

$$\mathbf{T_{w/s}} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Concatenating the two transformations, the transformation from the body axes to the wind axes $\mathbf{T_{w/b}}$ can be obtained as:

$$\mathbf{T_{w/b}} = \mathbf{T_{w/s}}\mathbf{T_{s/b}} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.12}$$

Forces and moments can be expressed in body or wind axis systems by using these transformation matrices.

3.3.2 Forces and Moments

The forces and moments on the aircraft arise due to aerodynamics, gravitational acceleration and engine. The aerodynamic forces are specified in the wind

axes. The components along the negative x–axis, positive y–axis and the negative z–axis are called drag, D, sideforce, Y and lift, L. In order to avoid dealing with state dependent moments of inertia, the aerodynamic moments are defined in the body axes. These are: the rolling moment L_r about the x–axes, the pitching moment M about the y–axis, and the yawing moment N about the z–axis. The forces and moments are specified in terms of dimensionless coefficients as follows.

$$D = \bar{q}SC_D \tag{3.13}$$

$$L = \bar{q}SC_L \tag{3.14}$$

$$Y = \bar{q}SC_Y \tag{3.15}$$

$$L_r = \bar{q}Sb_sC_{L_r} \tag{3.16}$$

$$M = \bar{q}Sb_cC_M \tag{3.17}$$

$$N = \bar{q}Sb_sC_N \tag{3.18}$$

where, $\bar{q}$ is the free-stream dynamic pressure, S is the wing reference area, b_s is the wing span, b_c is the wing mean aerodynamic chord, C_D is the drag coefficient, C_L is the lift coefficient, C_Y is the sideforce coefficient, C_{L_r} is the rolling moment coefficient, C_M is the pitching moment coefficient, and C_N is the yawing moment coefficient. The aerodynamic coefficients primarily depend on the aerodynamic angles, α and β, the Mach number and control surface deflections. They also depend upon the body rates. A detailed discussion of these coefficients is available in Reference [96]. The forces and moments due to the engine arise from the thrust, its location with respect to the aircraft center of gravity and misalignment angles. Thrust–related forces and moments are denoted by subscript T in the following.

Forces and moments due to aerodynamics and engine thrust in the body axes are:

$$F_B = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \mathbf{T}_{\mathbf{w/b}}^{-1} \begin{bmatrix} -D \\ Y \\ L \end{bmatrix} + \begin{bmatrix} F_{X_T} \\ F_{Y_T} \\ F_{Z_T} \end{bmatrix} \tag{3.19}$$

and

$$\tau_B = \begin{bmatrix} L_b \\ M_b \\ N_b \end{bmatrix} = \begin{bmatrix} L_r \\ M \\ N \end{bmatrix} + \begin{bmatrix} L_{r_T} \\ M_T \\ N_T \end{bmatrix} \tag{3.20}$$

Here, F_B is the force vector in the body axes, F_{X_T}, F_{Y_T} and F_{Z_T} are the components of the engine thrust vector in the body axes. $\mathbf{T_{w/b}}$ is the transformation matrix from the body to the wind axes, described earlier in Equation (3.12), and τ_B is the moment vector in the body axes. The variables L_{r_T}, M_T and N_T are the rolling moment, pitching moment and yawing moment due to the engine thrust.

3.3.3 Equations of Motion

Six–degree–of–freedom aircraft model is given by twelve first–order nonlinear differential equations involving the position and attitude dynamics. The twelve state variables consist of: (1) the inertial position of the aircraft represented by its topocentric coordinates x, y, z; (2) aircraft velocity components measured in the body axes U, V, W; (3) the body Euler angles denoted by ϕ, θ, ψ in roll, pitch and yaw, respectively; and (4) the angular rate of the body p, q, r in body axes. The aircraft equations of motion are given in the following.

The force equations are [96]:

$$\dot{U} = rV - qW - g\sin\theta + \frac{F_x}{m} \tag{3.21}$$

$$\dot{V} = -rU + pW + g\sin\phi\cos\theta + \frac{F_y}{m} \tag{3.22}$$

$$\dot{W} = qU - pV + g\cos\phi\cos\theta + \frac{F_z}{m} \tag{3.23}$$

Here, g is acceleration due to gravity and m is the mass of the aircraft.

The rotational kinematic equations are [69]:

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta \tag{3.24}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{3.25}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{3.26}$$

The angular acceleration equations are [69]:

$$\begin{aligned}\dot{p} &= [I_1 L_b + I_2 M_b + I_3 N_b - p^2(I_{xz} I_2 - I_{xy} I_3) \\ &+ pq(I_{xz} I_1 - I_{yz} I_2 - I_9 I_3) - pr(I_{xy} I_1 + I_8 I_2 - I_{yz} I_3) \\ &+ q^2(I_{yz} I_1 - I_{xy} I_3) - qr(I_7 I_1 - I_{xy} I_2 + I_{xz} I_3) \\ &- r^2(I_{yz} I_1 - I_{xz} I_2)]/det(I) \end{aligned} \tag{3.27}$$

$$\begin{aligned}\dot{q} &= [I_2 L_b + I_4 M_b + I_5 N_b - p^2(I_{xz} I_4 - I_{xy} I_5) \\ &+ pq(I_{xz} I_2 - I_{yz} I_4 - I_9 I_5) - pr(I_{xy} I_2 + I_8 I_4 - I_{yz} I_5) \\ &+ q^2(I_{yz} I_2 - I_{xy} I_5) - qr(I_7 I_2 - I_{xy} I_4 + I_{xz} I_5) \\ &- r^2(I_{yz} I_2 - I_{xz} I_4)]/det(I) \end{aligned} \tag{3.28}$$

$$\begin{aligned}\dot{r} &= [I_3 L_b + I_5 M_b + I_6 N_b - p^2(I_{xz} I_5 - I_{xy} I_6) \\ &+ pq(I_{xz} I_3 - I_{yz} I_5 - I_9 I_6) - pr(I_{xy} I_3 + I_8 I_5 - I_{yz} I_6) \\ &+ q^2(I_{yz} I_3 - I_{xy} I_6) - qr(I_7 I_3 - I_{xy} I_5 + I_{xz} I_6) \\ &- r^2(I_{yz} I_3 - I_{xz} I_5)]/det(I) \end{aligned} \tag{3.29}$$

where the determinant of the inertia matrix is given by:

$$det(I) = I_x I_y I_z - 2 I_{xy} I_{xz} I_{yz} - I_x {I_{yz}}^2 - I_y {I_{xz}}^2 - I_z {I_{xy}}^2, \tag{3.30}$$

and

$$I_1 = I_y I_z - {I_{yz}}^2 \tag{3.31}$$

$$I_2 = I_{xy} I_z + I_{yz} I_{xz} \tag{3.32}$$

$$I_3 = I_{xy} I_{yz} + I_y I_{xz} \tag{3.33}$$

$$I_4 = I_x I_z - {I_{xz}}^2 \tag{3.34}$$

$$I_5 = I_x I_{yz} + I_{xy} I_{xz} \tag{3.35}$$

$$I_6 = I_x I_y - {I_{xy}}^2 \tag{3.36}$$

$$I_7 = I_z - I_y \tag{3.37}$$

$$I_8 = I_x - I_z \tag{3.38}$$

$$I_9 = I_y - I_x \tag{3.39}$$

In these equations, I_x, I_y and I_z are the moments of inertia about the body x–axis, y–axis and z–axis, respectively. Since the inertia matrix, I is symmetric, only three cross moments of inertia, I_{xy}, I_{xz} and I_{yz} are required in addition to I_x, I_y and I_z for the complete specification of the inertia matrix. However if the aircraft is symmetric about the x–z plane as is mostly the case, two components of the cross moments of inertia I_{xy} and I_{yz} can be assumed to be zero. This assumption will result in considerable simplification of the moment equations.

Finally, the navigation equations are:

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \mathbf{T}_{\mathbf{b/i}}^{-1} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.40}$$

The transformation matrix from the inertial to the body frame $\mathbf{T_{b/i}}$ is given by Equation (3.3).

The six–degree–of–freedom model driven by the forces and moments is summarized in the block diagram given in Figure 3.4. It may be observed that the model requires twelve initial conditions. The only external inputs are the surface deflections and the throttle commands.

The location of any light on the runway in the camera coordinates can be determined if the position and orientation of the aircraft and the coordinates of the runway lights with respect to the inertial frame are known. A mathematical model of the camera is needed to establish the relationship between the position of the runway lights in the camera frame, and their position in the image plane. Such a model will be described in the following section.

3.4 Camera Model

In order to avoid the complexities of having to deal with optical aberrations caused by lenses, it is customary [27] to represent the camera model by a pinhole lens together with an image plane located at the focus. The distance between the

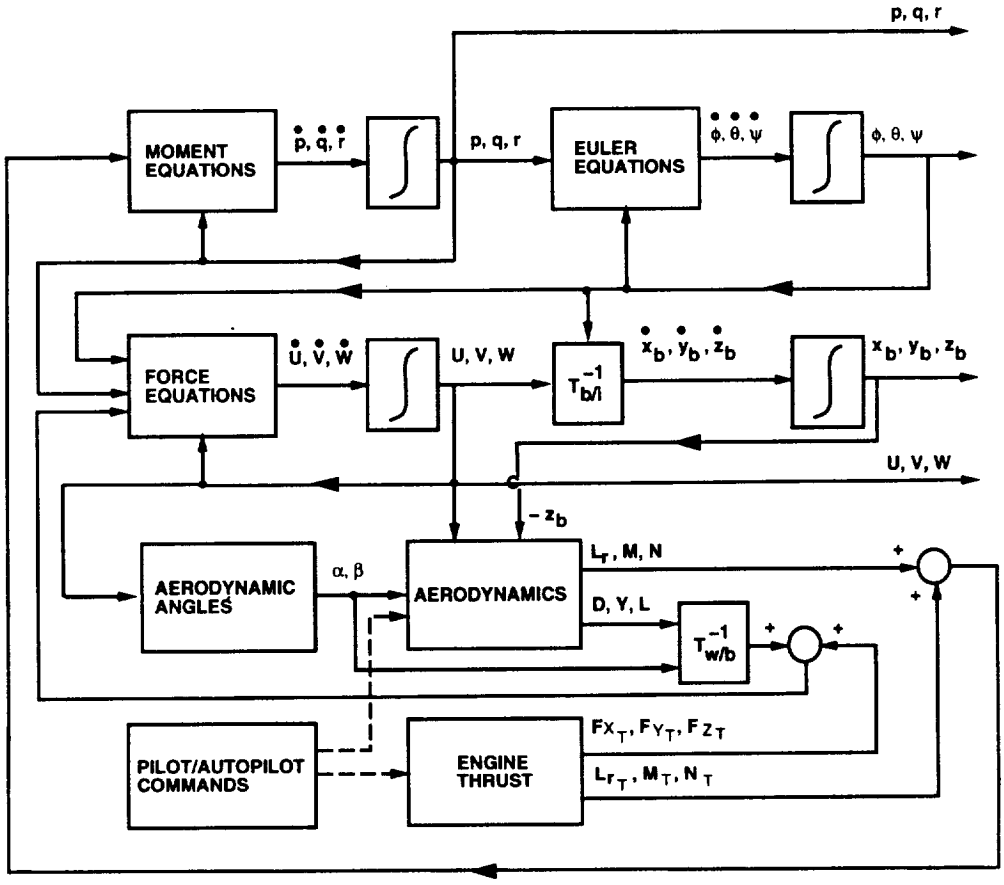

Figure 3.4: Aircraft equations of motion.

lens and the focus is the focal length, f. The image of a point $\mathbf{p}$ in the scene is determined by a ray projected from the point through the lens center. The location where this ray intersects the image plane is where the image of the point is registered. Such a model is shown in Figure 3.5. The model in the figure results

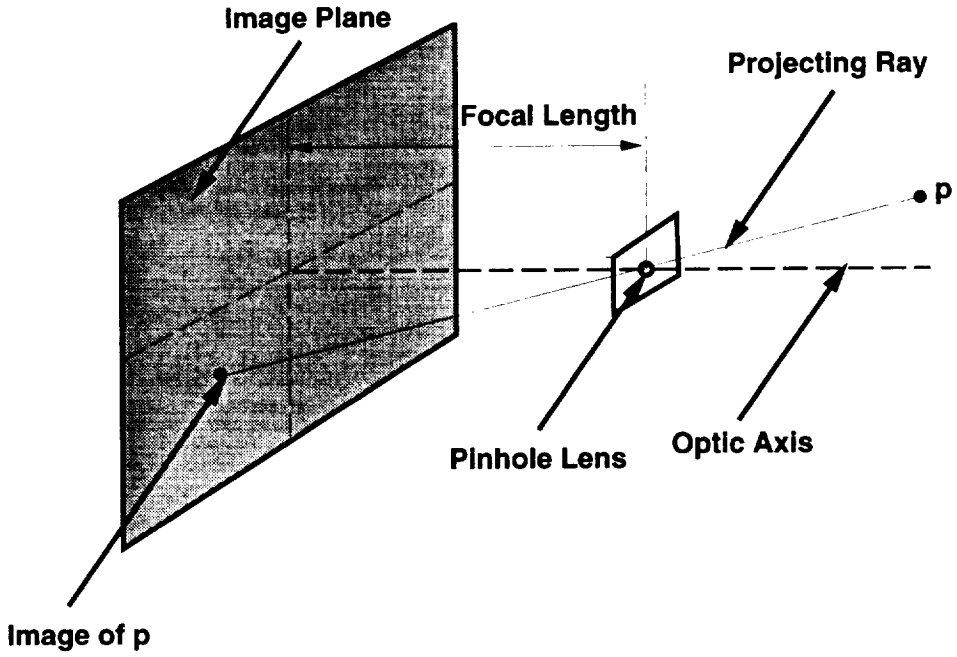

Figure 3.5: Pinhole camera model.

in images that are inverted left to right and top to bottom. This is in contradiction with how the human observer views the scene and how a television camera outputs the image. To avoid the inversion, a mathematically equivalent projection called the central projection [27] can be used.

Central projection involves projecting a ray from a point to a frontal image plane such that the ray passes through the lens center. The geometry is shown in Figure 3.6. The camera axis system has its origin at $\mathbf{c}$ as previously shown in Figure 3.2. The x-axis of the camera coordinate system is aligned with the optic axis. The z-axis points down and the y-axis completes the right-handed coordinate system. It can be observed that the central projection is a many-to-one mapping since all the object points along a projection ray are mapped to a single location in the

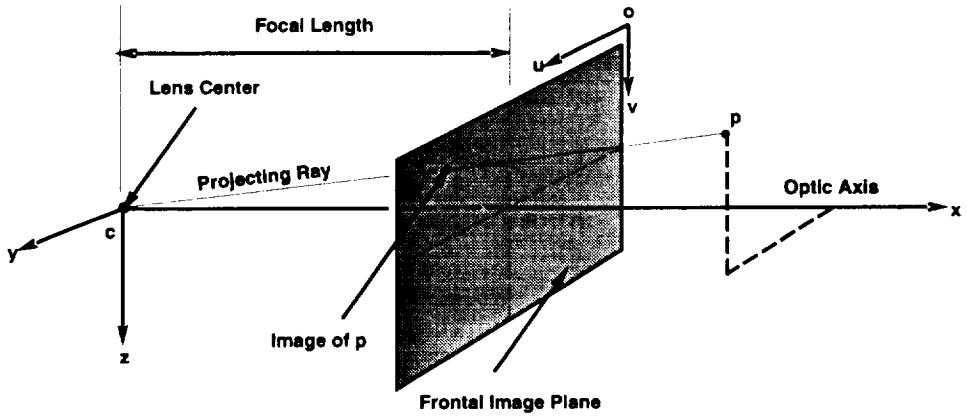

Figure 3.6: Pinhole camera with frontal image plane.

image plane. Thus if the location of an object point is known in the image plane, all that can be said about its three dimensional location is that it is located somewhere along the line passing through the image point and the lens center. On the other hand, if the location of the object point is known in the camera frame, its location in the image plane can be determined uniquely. This process is termed as direct perspective projection. These facts imply that there isn't enough information in one image to recover three dimensional geometry. Two or more images obtained from different vantage points may be required to reconstruct the three dimensional scene. The process of recovering the three dimensional coordinates from one or more images is called inverse perspective projection.

Real cameras capture the scene at discrete pixel locations indexed by rows and columns. Thus, every pixel is referenced by two coordinates: u and v with respect to a coordinate frame called the image frame with its origin, **o**, located at the top left hand corner of the image plane as shown in Figure 3.6. The u-axis is directed from left to right and the v-axis is directed from top to bottom of the image plane. Let, u_c and v_c be the coordinates of the camera center with respect to the image frame origin **o**. The image coordinates u_p and v_p for an object **p** can be obtained by constructing two sets of similar triangles from the geometric

relationships shown in Figure 3.6. Thus,

$$\frac{u_p - u_c}{f} = \frac{y_{cp}}{x_{cp}} \tag{3.41}$$

$$\frac{v_p - v_c}{f} = \frac{z_{cp}}{x_{cp}} \tag{3.42}$$

These two equations describe the direct perspective projection process. The position components used here are defined in Equations (3.7) through (3.9). Note that the parameter, f is the focal length of the camera.

Direct perspective projection equations for a pinhole camera model that take radial lens distortion, uncertainty scale factors and sampling into account are available in the literature [98]. Such models are too complex for the purposes of this report. However, they may be useful for error analysis of cameras.

3.5 Data Sources

Pilots use airport lighting for obtaining alignment guidance and glide slope information during night approach and landing. The geometric information and color coding in the airport lighting layout is utilized by the human perceptual system for estimating position and orientation with respect to the runway. Since the human visual system sees a perspective image of the airport lighting, position and orientation estimation requires the pilot to correlate the scene with the aircraft position. In practice, this is accomplished by repeated landings at particular runways. Based on the parallels drawn between the human perceptual system and a conceptual machine vision system described in Section 1.6, it should be possible to derive the runway relative position and orientation information by comparing the airport images with the geometric model of the airport lighting layout.

Before venturing into developing a system capable of generating the kind of information the pilot requires, an understanding of the standard airport lighting geometry is required.

3.5.1 Airport Lighting

The purpose of airport lighting is to provide information about airport/runway identification, approach direction, alignment and attitude information for safe night landing. Standard airport lighting is composed of the approach and runway lights [49]. The approach lights consist of centerline bars, sequenced flashers, threshold lights, cross bar lights, wing bar lights and the terminating bar lights. The runway lights include edge lights, centerline lights and touchdown zone lights. The approach as well as the runway lights are color coded and are located at fixed distances with respect to the runway threshold.

In airports with several runways, the type of lighting used on the runways assists the pilot in determining if the aircraft is headed towards the desired airport and the correct runway. Sequenced flashers and color coding of the threshold bar indicate the approach direction to the pilot. Similarly the centerline and edge lights provide lateral and vertical alignment guidance. Additional information available in the airport lighting structure useful for a machine vision system is described in the following.

3.5.1.1 Standard Approach Lighting System

Common configurations for approach lighting are the Calvert system and the standard configuration–A system [49]. The Calvert system is widely used in Europe and elsewhere around the world. The standard configuration–A approach lighting system is the national standard for civil and military use in the United States. Both systems are 3000 feet long. Figure 3.7 illustrates the standard configuration–A approach lighting system.

In the standard configuration–A approach lighting system, the centerline bars are composed of five white lights separated by 40.5 inches. There is a sequenced flasher in front of each centerline bar. The distance between the centerline bars is 100 feet. The cross bar, located 1000 feet from the runway threshold, consists of eight white lights on each side of the centerline bar. These lights are separated

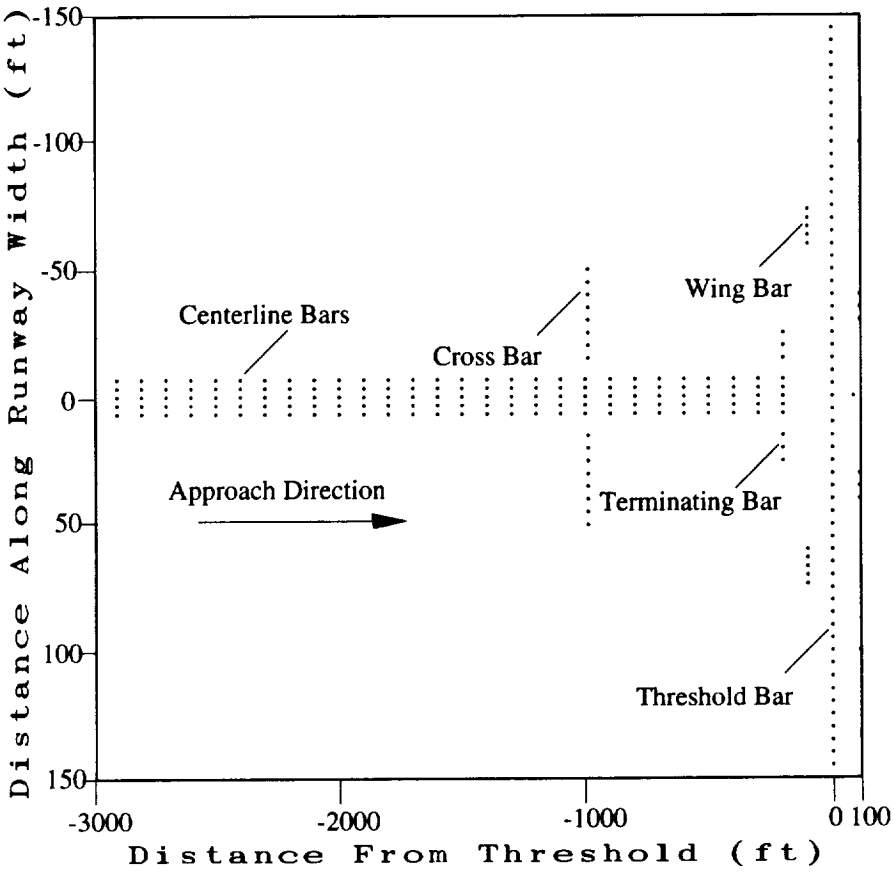

Figure 3.7: Standard configuration-A approach lighting system.

from each other by five feet. The threshold bar is located ten feet from the runway threshold and consists of green lights arranged five feet apart. The threshold bar runs along the entire width of the runway and extends 45 feet beyond the runway on each side. The wing bar is located at a distance of 100 feet from the threshold bar. The wing bar consists of five red lights placed symmetrically about the centerline. The inter-light separation is 40.5 inches. The terminating bar is located at a distance of 200 feet from the threshold bar, and consists of five red lights, 40.5 inches apart at the centerline and two sets of three red lights, five feet apart, placed symmetrically about the centerline.

Approach lights are usually placed on pedestals of different heights. The specifications for mounting approach lights in the United States are available in Reference [29].

For operations in reduced visibility such as Category II or lower, the International Civil Aviation Organization (ICAO) specifications are used for the lighting within 1000 feet of the runway threshold. The remaining 2000 feet of the lighting system is left as is. For the standard configuration-A system, this means that nine rows consisting of three red lights each are placed on either side of the centerline between the threshold bar and the the cross bar. Additionally, two rows with four white lights each are placed at 500 feet from the threshold, symmetrically about the centerline. Detailed layout of the Category II approach lighting system is described in Reference [49].

A medium approach lighting system (MALS) is often used at smaller airports for non-precision approaches. This system is 1400 feet long as opposed to 3000 feet long standard configuration-A approach lighting system. Also, the threshold bar is not continuous. Only four lights are placed on each side of the threshold to indicate the indicate the approach direction. A MALS layout is also given in Reference [49].

3.5.1.2 The Runway Lighting System

The runway lighting system consists of edge lights, centerline lights and touchdown zone lights. A typical runway lighting layout is shown in Figure 3.8. Standards

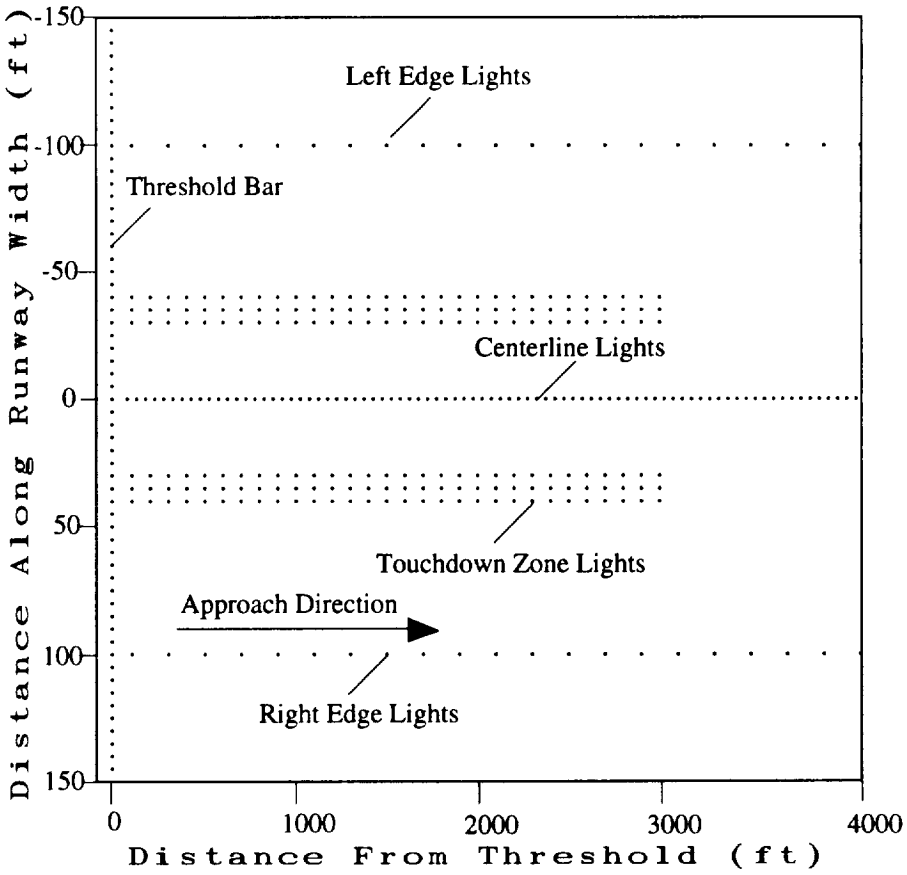

Figure 3.8: Runway lighting system.

for design and installation of runway lighting systems are given in [33, 34].

The runway edge lights are high intensity white lights, except for the last 2000 feet. The edge lights in the last 2000 feet are colored yellow to indicate a caution zone. The edge lights are located ten feet away from the pavement and the distance between the lights along-track is 200 feet.

The centerline lights are 50 feet apart and run all the way to the end of the

runway. The first centerline light is 75 feet from the runway threshold. These lights are white, except for the last 3000 feet in the approach direction where they are color coded. The lights for the first 2000 feet are alternate red and white, while the last 1000 feet are all red.

The touchdown zone lights start at 100 feet from the runway threshold and extend to 3000 feet in the direction of approach. The zone lights consist of three white lights, which are five feet apart and located at a distance of 30 feet about the centerline. The rows of zone lights are 100 feet apart from each other.

3.5.1.3 Model of Airport Lighting

In the two previous subsections, the layout geometry of the approach and runway lighting was described with respect to the threshold bar. Thus, by placing the origin of an inertial coordinate frame on the threshold bar with one axis aligned along the threshold bar and the other axis aligned with the runway centerline, the location of every light can be specified relative to the inertial coordinate system. These position coordinates form the airport lighting model.

In order to construct the geometric model of the runway lighting, information available from a standard airport design text [49] was discussed in this chapter. The deviations from the standard layout for any airport in the United States are documented in Jeppesen Charts [54] and Instrument Flight Rules (IFR) Supplement [22]. Detailed models of most airports can also be built using the information and survey maps available from city and county airport commissions.

The airport lighting model provides one source of information for the machine vision systems. The images from an onboard camera forms the other source of information.

3.6 Landing Flight Trajectory and Image Sequence Simulation

The equations described in the earlier sections can be used for simulating the landing flight trajectories and the associated camera images of the airport lights. The purpose of such a simulation is to serve as a test bed for position and orientation determination algorithms developed in later chapters.

The aircraft landing operation was described earlier in Section 3.1. A portion of this discussion was devoted to the desired glide path. The altitude, time and distance relationships for a approach speed of 220 feet/second along a three degree glide path were illustrated in Figure 3.1.

These relationships can be used to obtain the conditions for simulation. These are: (1) the aircraft is initially at 400 feet altitude and 6633 feet downrange from the threshold, (2) the touchdown point is 1000 feet from the threshold, (3) the aircraft approach speed is 220 feet/second, (4) the aircraft sink rate is 11.5 feet/second and (5) the aircraft is perfectly aligned with the runway centerline. Thus, the descent path is given:

$$x_{b_c} = -6633 + 220t \tag{3.43}$$

$$y_{b_c} = 0 \tag{3.44}$$

$$z_{b_c} = -400 + 11.5t \tag{3.45}$$

where, t is the time from the initial position; x_{b_c}, y_{b_c} and z_{b_c} are the desired or commanded aircraft position components along the inertial x–axis, y–axis and the z–axis.

Aircraft flight along the prescribed path can be simulated by using the aircraft aero–propulsive models along with the equations of motion discussed in a previous section. However, this requires aircraft specific aero-propulsive models and a suitable flight control system. These difficulties can be avoided by assuming that a suitable control system can be designed to closely track the trajectory. In

this case, the aircraft trajectory can be approximately simulated by using just the kinematic equations. Since the aircraft actual trajectory is expected to be close to the desired trajectory with minor deviations, the actual trajectory can be simulated by driving the linear and angular acceleration components by white noise. Thus, the kinematic equations required for simulation are:

$$\dot{x}_b = v_{b_x} \tag{3.46}$$

$$\dot{y}_b = v_{b_y} \tag{3.47}$$

$$\dot{z}_b = v_{b_z} \tag{3.48}$$

$$\dot{v}_{b_x} = \eta_{\ddot{x}_b} \tag{3.49}$$

$$\dot{v}_{b_y} = \eta_{\ddot{y}_b} \tag{3.50}$$

$$\dot{v}_{b_z} = \eta_{\ddot{z}_b} \tag{3.51}$$

$$\dot{\psi} = q \sin\phi \sec\theta + r \cos\phi \sec\theta \tag{3.52}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{3.53}$$

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{3.54}$$

$$\dot{r} = \eta_{\dot{r}} \tag{3.55}$$

$$\dot{q} = \eta_{\dot{q}} \tag{3.56}$$

$$\dot{p} = \eta_{\dot{p}} \tag{3.57}$$

$$\tag{3.58}$$

Here, v_{b_x}, v_{b_y} and v_{b_z} are the components of the inertial velocity, $\mathbf{V_b^i}$; $\eta_{\ddot{x}_b}$, $\eta_{\ddot{y}_b}$ and $\eta_{\ddot{z}_b}$ are the white noise components driving the linear acceleration components; and $\eta_{\dot{r}}$, $\eta_{\dot{q}}$ and $\eta_{\dot{p}}$ are the white noise components driving the angular acceleration components. The nomenclature for other terms remain unchanged.

The landing trajectory simulation is accomplished by integrating the system of Equations (3.46) through (3.58) with the initial conditions: $x_b = -6633$ feet, $y_b = 0$ feet, $z_b = -400$ feet, $v_{b_x} = 220$ feet/second, $v_{b_y} = 0$ feet/second, $v_{b_z} = 11.5$ feet/second, $\psi = 0$ degrees, $\theta = -3$ degrees, $\phi = 0$ degrees, $p = 0$ degrees/second, $q = 0$ degrees/second, and $r = 0$ degrees/second.

The image generation process corresponding to the landing trajectory simulation is summarized in the block diagram in Figure 3.9. The camera position and

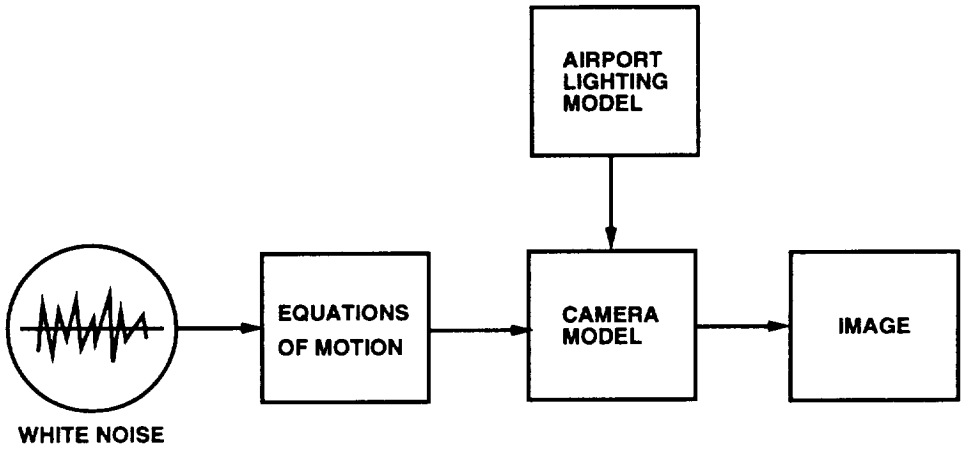

Figure 3.9: Image generation process.

orientation specified by the aircraft equations of motion, the camera model and a model of the airport lighting are used to generate an image of the airport lighting layout.

Nighttime images of the airport are simulated using the lighting layout illustrated in Figures 3.7 and 3.8. For image synthesis, the camera is assumed to be fixed to the aircraft, looking down along the glide slope. Since the camera axis is assumed to be colocated with the body axis, the look down angle is same as the pitch angle θ. The image is assumed to be digitized on a 512×512 pixel array, with the camera focal length being 600 pixels. This translates into a field-of-view of about 46 degrees. Image synthesis is achieved in two steps. First, the airport lights within the camera field-of-view are determined by using the known camera position, orientation and the field-of-view. Second, the lights within the field-of-view are projected onto the image plane by using the perspective projection equations described earlier. A simulated image constructed using this process is shown in Figure 3.10. This image corresponds to the camera located at an altitude of 95 feet and 812 feet downrange from runway threshold.

The steps in the image generation process are summarized in Table 3.3.

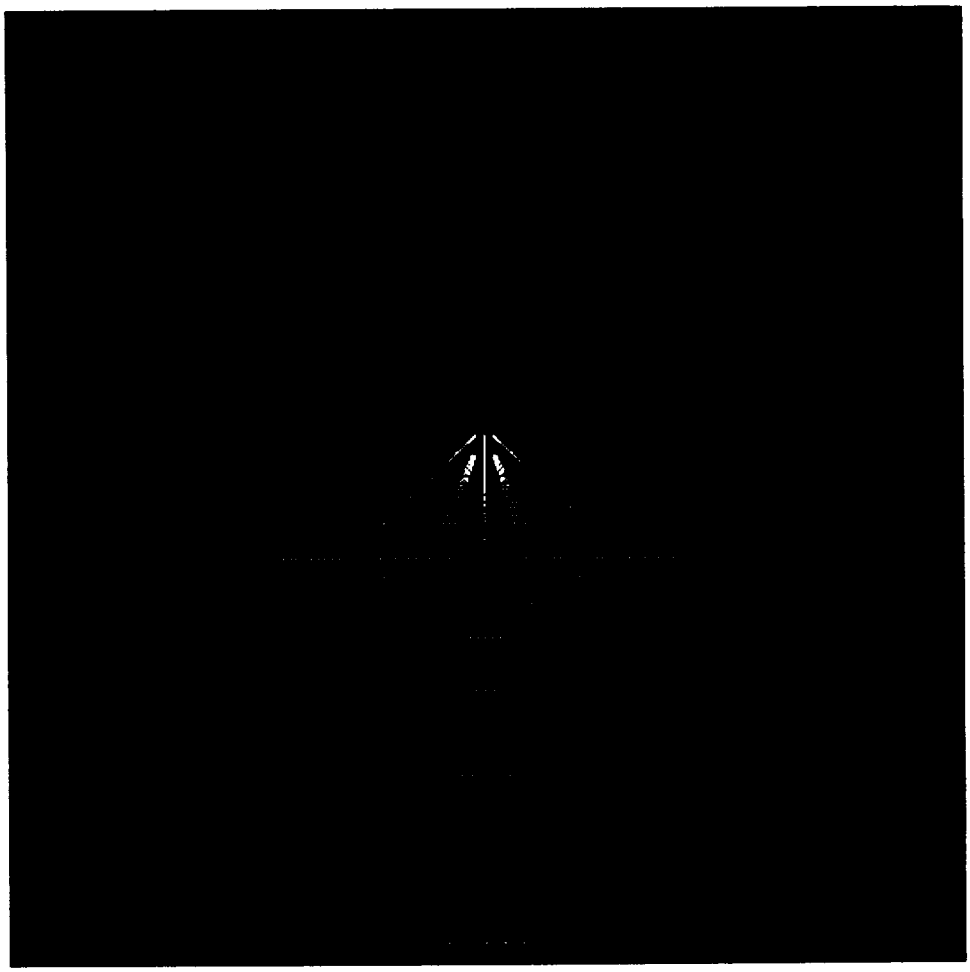

Figure 3.10: Simulated airport image.

Table 3.3: Summary of the Image Generation Process

1. Initialize the image matrix $f(i,j) = 0$; $i = 1,2,\ldots,512$; $j = 1,2,\ldots,512$.

2. Use yaw, pitch and roll attitudes ψ, θ and ϕ to compute the elements of the transformation matrix r_1 through r_9 using Equation (3.3).

3. Compute the position of the airport lights x_{cp}, y_{cp} and z_{cp} using Equations (3.7), (3.8) and (3.9) with the actual aircraft position components, x_b, y_b and z_b, such that $p = 1,2,\ldots,M$ where, M is the number of lights within the field–of–view of the camera.

4. Compute the location of each light in the image plane u_p and v_p using the perspective projection Equations (3.41) and (3.42).

5. Quantize every u_p and v_p using $\lfloor u_p + 1/2 \rfloor$ and $\lfloor v_p + 1/2 \rfloor$ where, $\lfloor \quad \rfloor$ is the floor function. Following the definition of the floor function in Reference [38], $\lfloor u_p + 1/2 \rfloor$ is the greatest integer smaller than or equal to $u_p + 1/2$.

6. Set the image matrix $f(\lfloor u_p + 1/2 \rfloor, \lfloor v_p + 1/2 \rfloor) = 256$.

It should be noted that when the aircraft is at 400 feet altitude along three degree glide slope, both the approach lights and the runway lights are within the field–of–view. As the aircraft proceeds along the descent path, the approach lights begin moving out of field–of–view below 230 feet altitude. Any position and orientation determination algorithm must be able to adapt to this fact.

3.7 Algorithm Development Considerations

Algorithm development is dependent on the available data sources and mathematical models. Two data sources were identified in previous sections. These are: (1) image of the airport lighting acquired by the camera and (2) the known airport lighting geometry. Earlier in this chapter, the airport light locations in the model were mathematically related to their respective locations in the image plane using a pinhole camera model and camera motion parameters. The image formation process using the mathematical relations was further discussed in Section 3.6. Earlier in Section 2.3.7, the difficulties of correlating the image features with the model features were examined. Clearly, these difficulties can be eliminated if the image features and model features are transferred into a common framework. There are two natural choices for performing such comparisons. The comparisons can be carried out in the inertial plane or in the image plane. Each of these choices result in different families of algorithms.

Figure 3.11 illustrates the inertial frame-based family of methods for runway relative position and orientation. The image of the runway lighting acquired by the camera is transformed to the inertial plane using inverse perspective projection. This requires an initial estimate of aircraft position and attitude. Equations (3.41) and (3.42) are used with Equations (3.7) through (3.9) to recover the inertial locations of the lights x_p and y_p by assuming a camera position, orientation and that all the airport lights are located on the $z_p = 0$ plane. The difference between the features extracted from this layout and the features extracted from the known

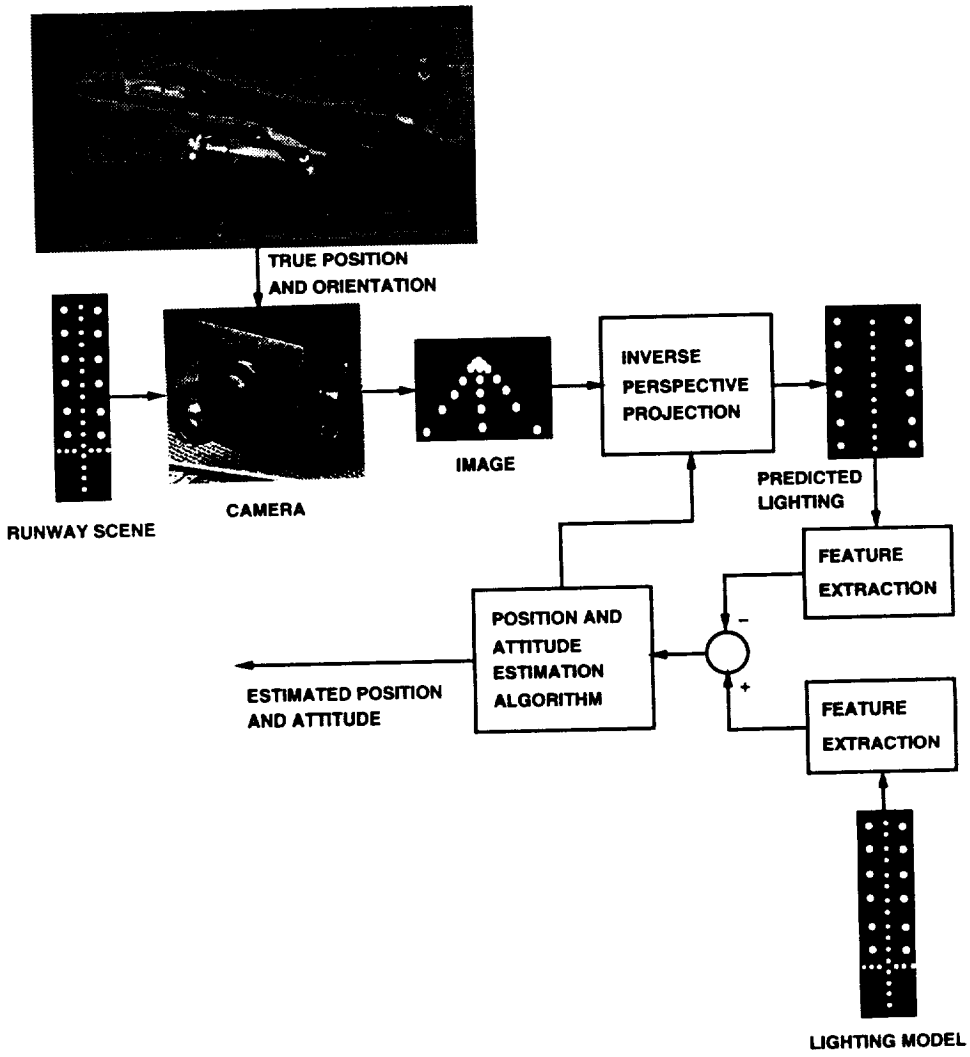

Figure 3.11: Solution family I.

lighting layout form the inputs to the position and orientation algorithm. The algorithm iteratively improves the position and orientation estimates in order to achieve a better matching. The improved estimates are then used for inverse perspective projection. This procedure recovers the camera position and orientation by driving the feature errors to zero.

The procedure given in Figure 3.12 describes the second family of methods for runway relative position and orientation estimation by carrying out feature matching computations in the image plane. In this case, the camera model is

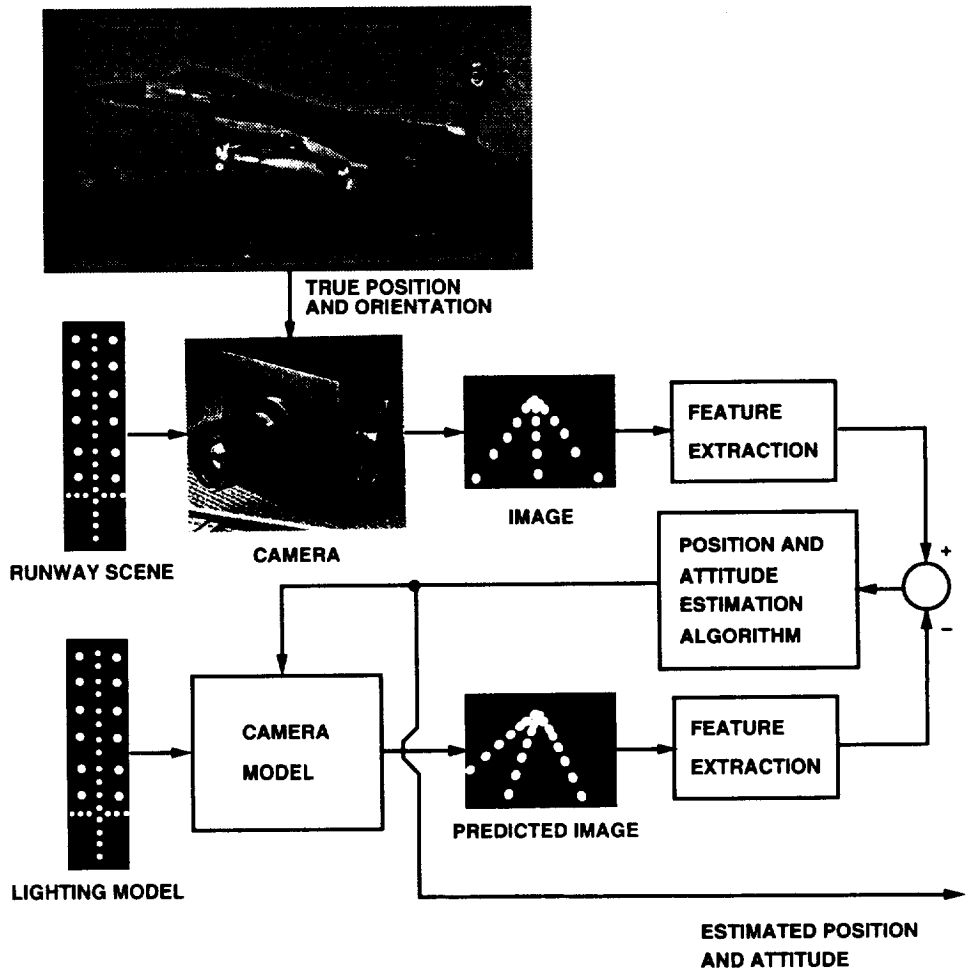

Figure 3.12: Solution family II.

used together with the airport lighting model to predict the image of the lighting arrangement. This prediction is based on an assumed camera position and orienta-

tion. Next, the difference between the features extracted from the predicted image and those extracted from the actual image of the airport lighting as seen by the camera are fed into a position and orientation algorithm to refine the position and orientation estimates. These estimates are then used for updating the predicted image. Thus as in the first family of methods, the position and orientation states are recovered by driving the feature errors to zero.

The two family of solution methods have their advantages and disadvantages. Since inverse perspective projection is used in the first family of methods, the assumption that all airport lights lie on the $z_p = 0$ plane is necessary. Moreover, due to perspective projection, lights farther down the runway are bunched together in the image plane. Location of these lights with respect to the inertial frame cannot be accurately recovered using the inverse perspective projection. Additionally, this family of methods require active model adaptation to remove the lights outside the field-of-view of the model as the aircraft moves because portions of the airport along the descent path. The main advantage is that since the structure of the predicted and model lighting is well defined in the inertial frame, the comparisons are straight forward.

The second family of methods use direct perspective projection to synthesize the predicted image. As a result, the assumption that all airport lights lie on the ground plane $z_p = 0$ plane is not needed. The model adaptation process is automatic because only the lights that are within the field-of-view of the pinhole camera model are used for synthesis of the predicted image. The main disadvantage of this family of methods is that the structure of lighting in the predicted and the camera images is difficult to identify due to perspective distortion.

Note that in both the procedures outlined in the foregoing, a single image of the airport lighting is used as a part of an iterative scheme to recover the camera position and orientation coordinates. Algorithms based on each of these families of solution methods will be discussed in the ensuing chapters.

3.8 Summary

The foundations for the development of vision–based algorithms for position and orientation determination were laid in this chapter. First, the kinematics of flight along the glide slope were described. Federal Aviation Administration defined landing categories together with the associated decision heights, and the performance specification for precision approach and landing for these categories were then discussed. Since the glide slope is defined with respect to the runway centerline and the touchdown point on the runway, and the airport lighting geometry was also defined with respect to the runway threshold. The origin of the inertial coordinate frame was assumed to be located at the threshold with the x–axis pointing along the centerline in the approach direction, the y–axis along the threshold and the z–axis pointing down. With this choice of the inertial coordinate system, and definitions of the body and camera coordinate systems, the locations of lights in the lighting model were related to their location with respect to the camera coordinate system. This relation was established in terms of the aircraft location and orientation with respect to the inertial coordinate system, and the camera location with respect to the aircraft fixed body coordinate system. Since, both the aircraft position and the orientation are a consequence of the translational and rotational motion of the aircraft subjected to propulsive, gravitational and aerodynamic forces, the equations of motion describing the dynamics and the kinematics of the aircraft were discussed. A pinhole camera model was then described for relating the camera relative coordinates of the model lights to their image coordinates. Using the models and equations, procedures for simulating the landing flight and images along the landing path were described. Finally, the fact that direct and inverse transformations between the location of the lights in the inertial frame and the image plane can be computed using the equations described in the chapter resulted in two possible solution approaches for determination of runway relative aircraft position and orientation.

Chapter 4

Parameter Optimization Based Position Determination Methods

The runway position determination techniques discussed in this chapter are based on the first solution family illustrated in Figure 3.11. The runway relative orientations, ψ, θ and ϕ, are assumed to be known in all the algorithms presented in this chapter.

Let the image coordinates of a light p in the image be given by u_p and v_p. The relation between these coordinates and the camera relative position components is given by the direct perspective projection equations, described earlier in Equations (3.41) and (3.42). For notational simplification let,

$$U_p = \frac{u_p - u_c}{f} \tag{4.1}$$

$$V_p = \frac{v_p - v_c}{f} \tag{4.2}$$

where, u_c and v_c are the coordinates of the image center with respect to the image frame and f is the focal length of the camera. Substituting for u_p and v_p in terms of light and aircraft position vector components results in the following relations:

$$U_p = \frac{r_4(x_p - x_b) + r_5(y_p - y_b) + r_6(z_p - z_b) + k_y}{r_1(x_p - x_b) + r_2(y_p - y_b) + r_3(z_p - z_b) + k_x} \tag{4.3}$$

$$V_p = \frac{r_7(x_p - x_b) + r_8(y_p - y_b) + r_9(z_p - z_b) + k_z}{r_1(x_p - x_b) + r_2(y_p - y_b) + r_3(z_p - z_b) + k_x} \tag{4.4}$$

Since the camera location with respect to aircraft center of gravity k_x, k_y and k_z are known quantities, they can be set to zero without loss of generality. This simplification implies that the camera frame is colocated with the body frame. The inverse perspective projection equations can be obtained from Equations (4.3) and (4.4) as:

$$x_p = x_b + \frac{s_{2p}s_{6p} - s_{3p}s_{5p}}{s_{1p}s_{5p} - s_{2p}s_{4p}}(z_p - z_b) \tag{4.5}$$

$$y_p = y_b + \frac{s_{1p}s_{6p} - s_{3p}s_{4p}}{s_{2p}s_{4p} - s_{1p}s_{5p}}(z_p - z_b) \tag{4.6}$$

The quantities s_{1p} through s_{6p} depend on the image coordinates U_p and V_p, and the known elements of the transformation matrix r_1 through r_9. They are defined by the equations:

$$s_{1p} = U_p r_1 - r_4 \tag{4.7}$$

$$s_{2p} = U_p r_2 - r_5 \tag{4.8}$$

$$s_{3p} = U_p r_3 - r_6 \tag{4.9}$$

$$s_{4p} = V_p r_1 - r_7 \tag{4.10}$$

$$s_{5p} = V_p r_2 - r_8 \tag{4.11}$$

$$s_{6p} = V_p r_3 - r_9 \tag{4.12}$$

Examination of the inverse perspective projection equations given by Equations (4.5) and (4.6), reveals that the x_p and y_p position components of all airport lights are shifted by the aircraft position components x_b and y_b, and scaled by the aircraft altitude, $-z_b$. Given the aircraft position components and the vertical coordinate z_p of each light, Equations (4.5) and (4.6) uniquely determine the individual light horizontal position components x_p and y_p. In order to make the problem solvable, z_p can be set to zero or a constant for all airport lights. This assumption is reasonable specially when the aircraft is at higher altitudes. Since the position of the aircraft relative to the plane containing the airport lights is of interest, define the aircraft altitude above the plane of the runway by h. Thus,

with $z_p = 0$ assumption:

$$x_p = x_b + \frac{s_{2p}s_{6p} - s_{3p}s_{5p}}{s_{1p}s_{5p} - s_{2p}s_{4p}}h \tag{4.13}$$

$$y_p = y_b + \frac{s_{1p}s_{6p} - s_{3p}s_{4p}}{s_{2p}s_{4p} - s_{1p}s_{5p}}h \tag{4.14}$$

These two equations show that the projected positions of the lights on the plane of the runway are dependent on three parameters, x_b, y_b and h.

Since the position of each airport light is known from the airport lighting layout, the position components x_b, y_b and h that results in a match between the known position and the projected position of each light is the desired solution. This problem can be conveniently posed as parameter optimization problem for determining aircraft position components that minimize the matching error. Early versions of two algorithms based on this idea are described in Reference [15]. Revised version of these algorithms are described in the following sections. The first algorithm assumes that the aircraft altitude is available from an onboard altimeter. The second algorithm does not make this assumption.

4.1 Algorithm I

This algorithm assumes that in addition to the aircraft attitudes, ψ, θ and ϕ, the aircraft altitude is available from an onboard altimeter. Thus, for every light detected in the image s_{1p} through s_{6p} can be computed using Equations (4.7) through (4.12). These can then be used in Equations (4.13) and (4.14) for computing the relative position components of each light, detected in the image, as follows:

$$x_p - x_b = \frac{s_{2p}s_{6p} - s_{3p}s_{5p}}{s_{1p}s_{5p} - s_{2p}s_{4p}}h \tag{4.15}$$

$$y_p - y_b = \frac{s_{1p}s_{6p} - s_{3p}s_{4p}}{s_{2p}s_{4p} - s_{1p}s_{5p}}h \tag{4.16}$$

In the ideal case, the projected airport lighting layout would be bounded by a rectangle of the same dimensions as the rectangle bounding the model lighting

layout as shown in Figure 4.1.

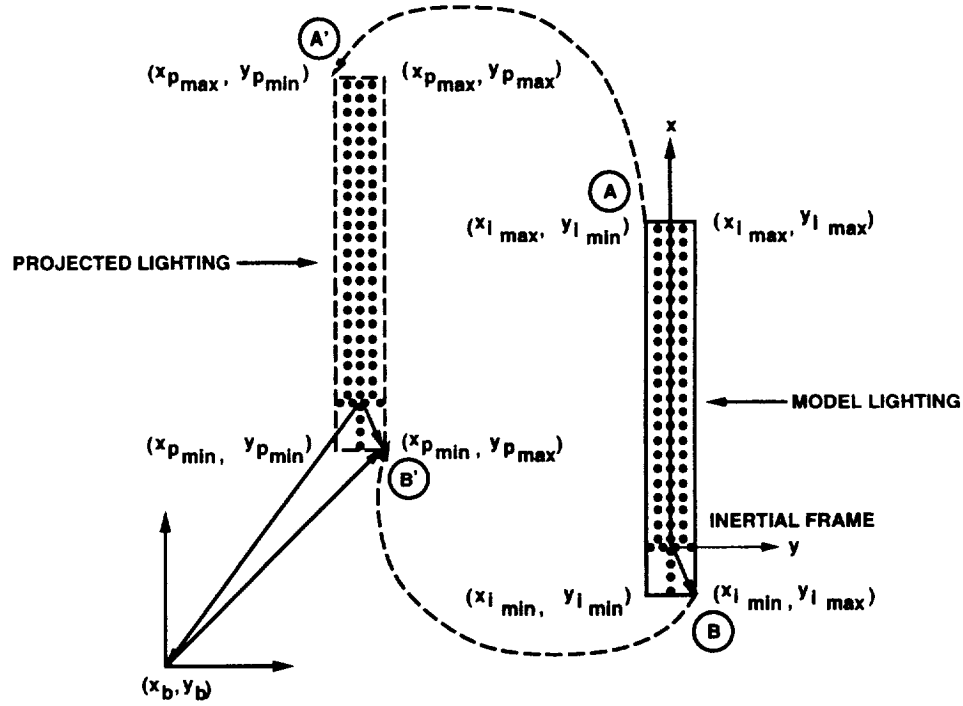

Figure 4.1: Envelope matching in the inertial frame.

Using Equations (4.15) and (4.16), the minimum and the maximum relative coordinates of the projected lighting can be found to be $x_{p_{min}}$, $x_{p_{max}}$, $y_{p_{min}}$ and $y_{p_{max}}$. The maxima and minima define the rectangle on the horizontal plane that encloses the projected lighting layout as shown in Figure 4.1. Since the coordinates of every airport light are known with respect to the inertial coordinate system, via the airport lighting model, the minimum and maximum coordinates of the lighting model can be found to be x_{imin}, x_{imax}, y_{imin} and y_{imax}. These coordinates define the rectangle that encloses the model lighting shown in Figure 4.1.

Consider the coordinates of the upper left corners, **A'** and **A** of the enclosing rectangles in Figure 4.1. These are, $(x_{p_{max}}, y_{p_{min}})$ with respect to the aircraft and (x_{imax}, y_{imin}) with respect to the inertial coordinate system. Using the relative geometry shown in Figure 4.1, it may be observed that:

$$x_b = x_{imax} - x_{p_{max}} \tag{4.17}$$

$$y_b = y_{i_{min}} - y_{p_{min}} \tag{4.18}$$

Similarly, using the coordinates of the lower right corners, **B'** and **B**,

$$x_b = x_{i_{min}} - x_{p_{min}} \tag{4.19}$$

$$y_b = y_{i_{max}} - y_{p_{max}} \tag{4.20}$$

The average of these two calculations can be taken as the aircraft position estimate. Combining Equations (4.17) with (4.19) and (4.18) with (4.20) yields:

$$x_b = \frac{(x_{i_{min}} - x_{p_{min}}) + (x_{i_{max}} - x_{p_{max}})}{2} \tag{4.21}$$

$$y_b = \frac{(y_{i_{min}} - y_{p_{min}}) + (y_{i_{max}} - y_{p_{max}})}{2} \tag{4.22}$$

So far only the ideal case has been examined. In the real situation, perspective projection causes the lights at the far end of the runway to be bunched together in the image plane making it difficult to determine their position. Since the lights closer to the camera are well separated in the image plane, it is more reasonable to use a weighted average of the x_b position estimates. Thus an improved position estimate for x_b is:

$$x_b = \frac{w_1(x_{i_{min}} - x_{p_{min}}) + w_2(x_{i_{max}} - x_{p_{max}})}{(w_1 + w_2)} \tag{4.23}$$

w_1 and w_2 are the weighting factors. No such rationale can be applied along the y–axis. Hence, Equation (4.22) can be used directly.

The position estimation algorithm is summarized in Table 4.1.

The attractive features of this algorithm are: (1) envelope matching does not require any explicit identification of individual lights, (2) no iterative computations are required, and (3) only a single image is required. However, the two significant difficulties with this algorithm are that accurate knowledge of altitude is required and the elevation of the airport lights, $-z_p$, is not included in the computations.

Algorithm I does not take advantage of the fact that numerous images are available along the descent path. Since these images are related to each other by

Table 4.1: Summary of Algorithm I

1. For every light in the scene compute U_p and V_p using Equations (4.1) and (4.2) with $p = 1, 2, \ldots, M$ where, M is the number of lights detected in the image.

2. Use ψ, θ and ϕ to compute r_1 through r_9 using Equation (3.3).

3. Compute s_{1_p} through s_{6_p} using Equations (4.7) through (4.12).

4. Compute $x_p - x_b$ and $y_p - y_b$, $p = 1, 2, \ldots, M$, using Equations (4.15) and (4.16) along with the known altitude.

5. Compute $max\{x_p\}$, $min\{x_p\}$, $max\{y_p\}$ and $min\{y_p\}$.

6. Compute $max\{x_i\}$, $min\{x_i\}$, $max\{y_i\}$ and $min\{y_i\}$, $i = 1, 2, \ldots, N$, using the coordinates of N airport lights within the model.

7. Compute the inertial position components, x_b and y_b using Equations (4.23) and (4.22).

aircraft motion, equations of motion can be used with this algorithm for obtaining improved aircraft position estimates. Kalman filter [3] is a natural choice for generating improved position estimates by weighting the position propagated using the equations of motion and the position estimates provided by the algorithm in Table 4.1 in a statistically optimal manner. For this purpose, a six-state Kalman filter with three position components and three velocity components is used together with Algorithm I for obtaining aircraft position and velocity estimates along the descent path. The details of this Kalman filter are described in Appendix A. The matrices required for implementing the Kalman filter are described in Appendix B. The two position components generated by Algorithm I are used as measurements for the Kalman filter. The known altitude is used as the third measurement. With these measurements, the Kalman filter provides the improved position and velocity estimates.

Most general aviation aircraft use barometric altimeter which has a limited accuracy because the measurement depends on ambient temperature. Although the measurement accuracy is sufficient for maintaining the required vertical separation for Air Traffic Control, it is inadequate for operations close to the ground. If altitude could be computed reliably, it could be used for augmenting the barometric altimeter reading.

An algorithm that does not require altitude measurements is described in Section 4.2. However, the lack of altitude information results in an iterative algorithm because three position components are estimated from two inverse perspective projection equations.

4.1.1 Results Using Algorithm I

Results of two cases obtained using Algorithm I are described in this section. The first case is obtained using $w_1 = 1$ and $w_2 = 0$. The second case is obtained using the weighting factors $w_1 = 1$ and $w_2 = 1$ in Equation (4.23). In both cases, the landing scenario discussed in Section 3.6 is used. Aircraft landing flight trajec-

tory and the images along the glide slope are also simulated using the procedures described in Section 3.6. For initializing the Kalman filter, errors of 1000 feet in the along–track position x_b and 100 feet in the cross–track position y_b are assumed. The inertial velocity components were all initialized to zero.

4.1.1.1 Algorithm I with $w_1 = 1$ and $w_2 = 0$

The error residuals of the runway relative position and velocity components are described in this section. The error residual is defined as the difference between the value estimated by the Kalman filter and the true value.

The along–track position error residual is shown in Figure 4.2. It may be seen

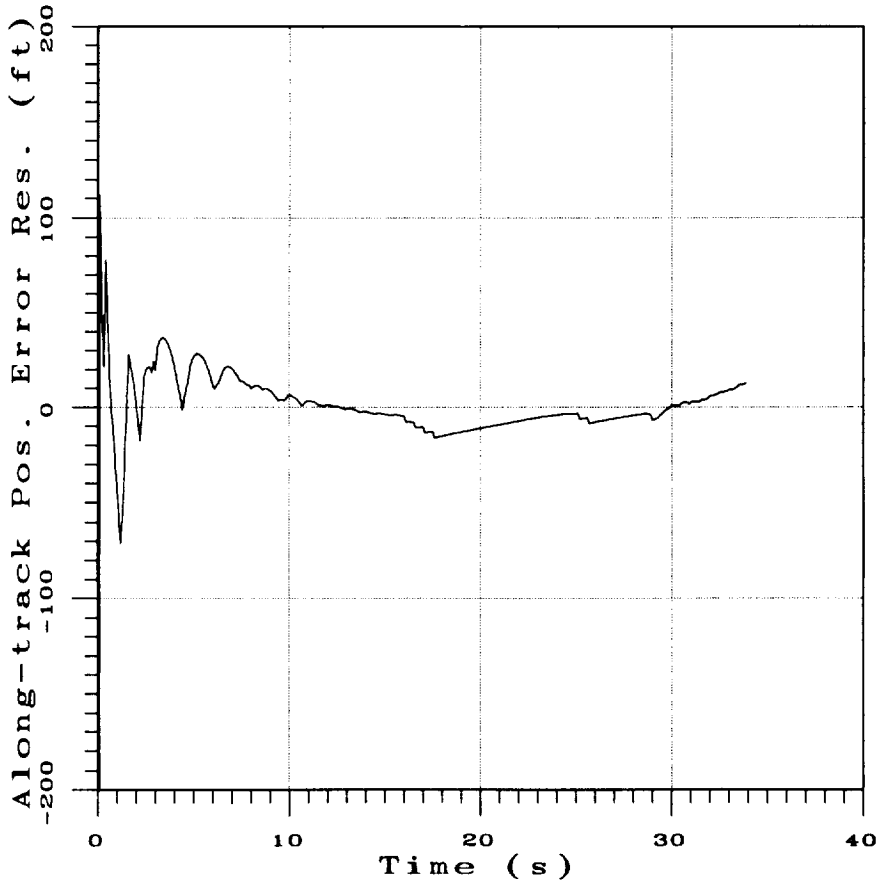

Figure 4.2: Along–track position error using Algorithm I with $w_1 = 1$ and $w_2 = 0$.

from the figure that the along-track position estimate converges to within ±100 feet in less than one second.

The cross-track position error residual presented in Figure 4.3 shows that the

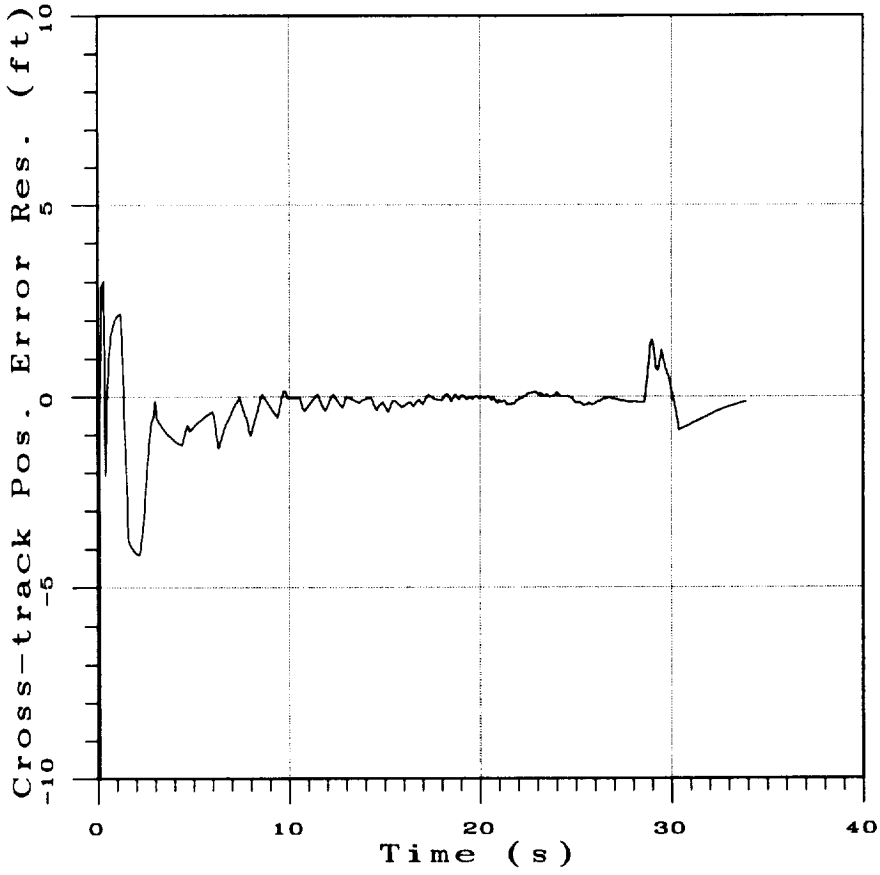

Figure 4.3: Cross-track position error using Algorithm I with $w_1 = 1$ and $w_2 = 0$.

cross-track position estimate converges to within ±5 feet in less than one second.

Figures 4.4 and 4.5 show the along-track velocity (v_{b_x}) and cross-track velocity (v_{b_y}) error residuals. It may be observed from the figures that the along-track velocity estimate settles to within ±10 feet/second in five seconds and the cross-track velocity settles to within ±5 feet/second in less than one second.

The position error residuals for a range of altitudes corresponding to the FAA landing categories are summarized in Table 4.2. The cross-track position (y_b)

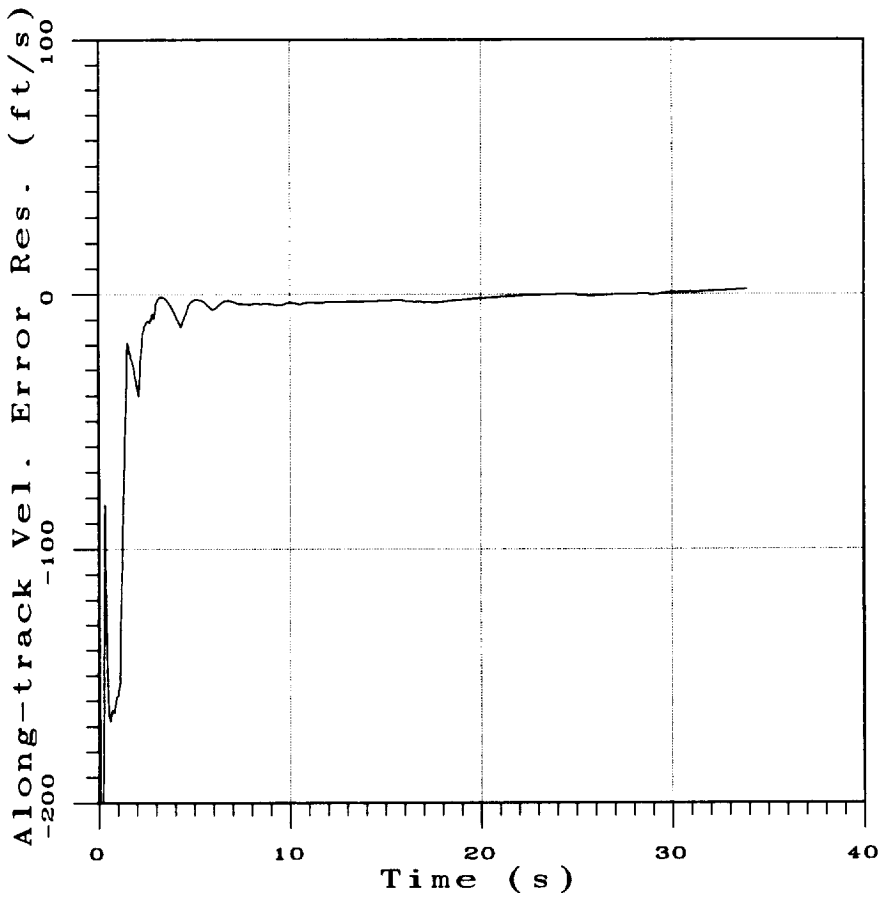

Figure 4.4: Along–track velocity error using Algorithm I with $w_1 = 1$ and $w_2 = 0$.

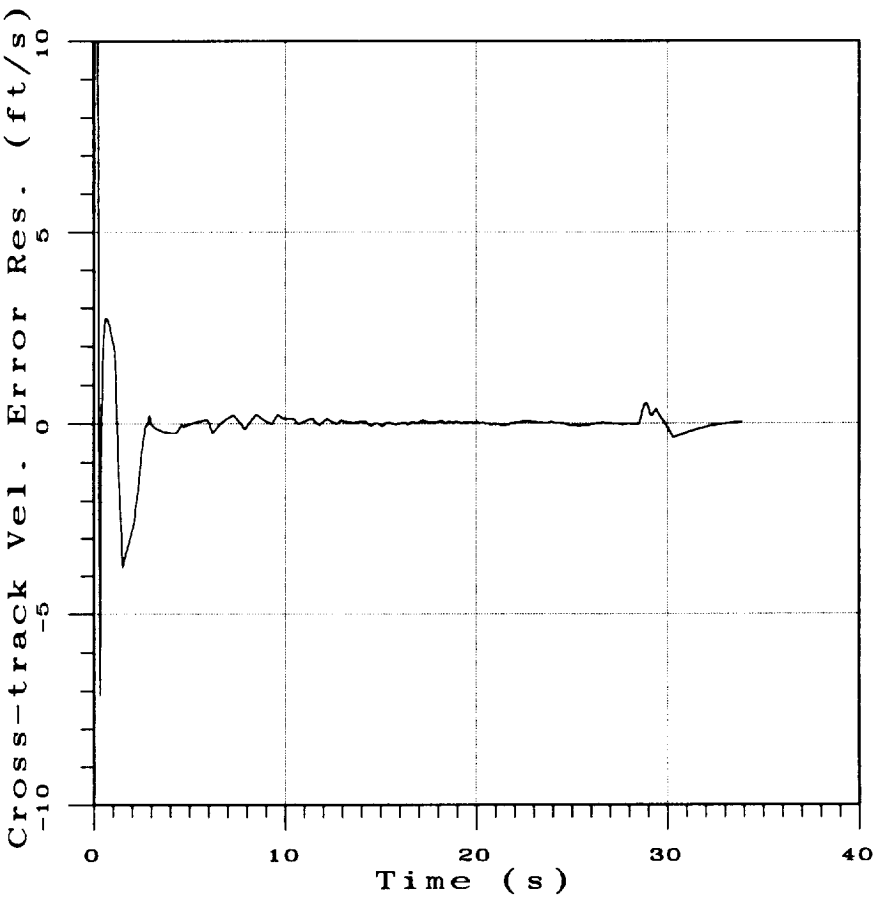

Figure 4.5: Cross-track velocity error using Algorithm I with $w_1 = 1$ and $w_2 = 0$.

error residual is in feet. By comparing Table 4.2 to Table 3.2, it may be seen that

Table 4.2: Algorithm I with $w_1 = 1$ and $w_2 = 0$ Results

Category	y_b
CAT I	± 0.54
CAT II	± 0.22
CAT IIIa	± 1.49
CAT IIIb & c	± 0.85

Algorithm I along with the Kalman filter generates position estimates which meet the navigation accuracy requirements for all the three categories.

4.1.1.2 Algorithm I with $w_1 = 1$ and $w_2 = 1$

In order to assess the benefit of matching both the near and the far ends of the envelopes shown in Figure 4.1, the error residuals of the runway relative position and velocity components obtained using Algorithm I with $w_1 = 1$ and $w_2 = 1$ are examined in this section.

Figure 4.6 shows the along-track position error residual. It may be seen from the figure that the along-track position estimation error continues to grow as a function of time. This is due to the inability of correctly recovering the position components of distant lights using the inverse perspective projection equations. The dimension of the predicted envelope along the viewing direction is therefore shorter than the model envelope. In an attempt to match the shorter envelope to the longer model envelope, the aircraft position estimate is erroneously estimated. These erroneous position estimates when provided as measurements to the Kalman filter results in incorrect position estimates. From Figure 3.11 it may be seen that since the position estimates are used for inverse perspective projection, grossly incorrect estimates of the aircraft position components would lead to a gross mismatch between the predicted lighting and model layouts to an extent that subsequent correct recovery of the aircraft position components may not be

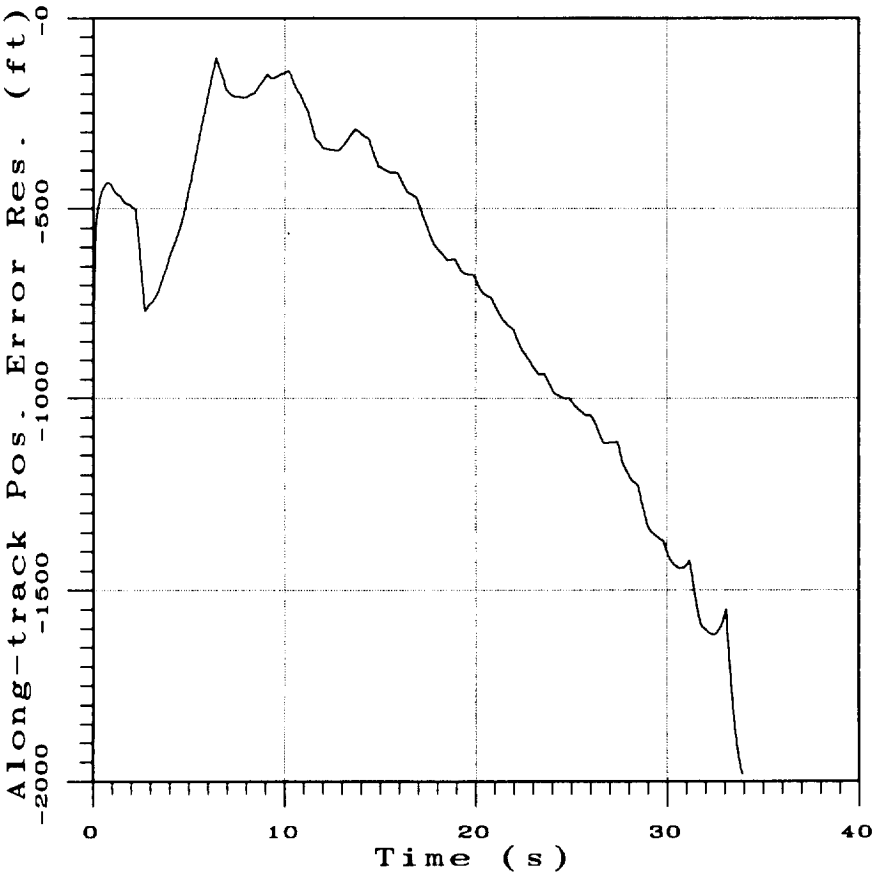

Figure 4.6: Along-track position error using Algorithm I with $w_1 = 1$ and $w_2 = 1$.

possible.

The along-track position result described in Figure 4.6 under-predicts the true position which could cause the pilot to overshoot the touchdown point. A far more dangerous situation would arise if the along-track position estimates over-predicted the along-track position because that could cause the pilot to land short of the runway.

The cross-track position error residual is shown in Figure 4.7. This figure shows that the cross-track position estimate converges to within ± 5 feet in less than one second.

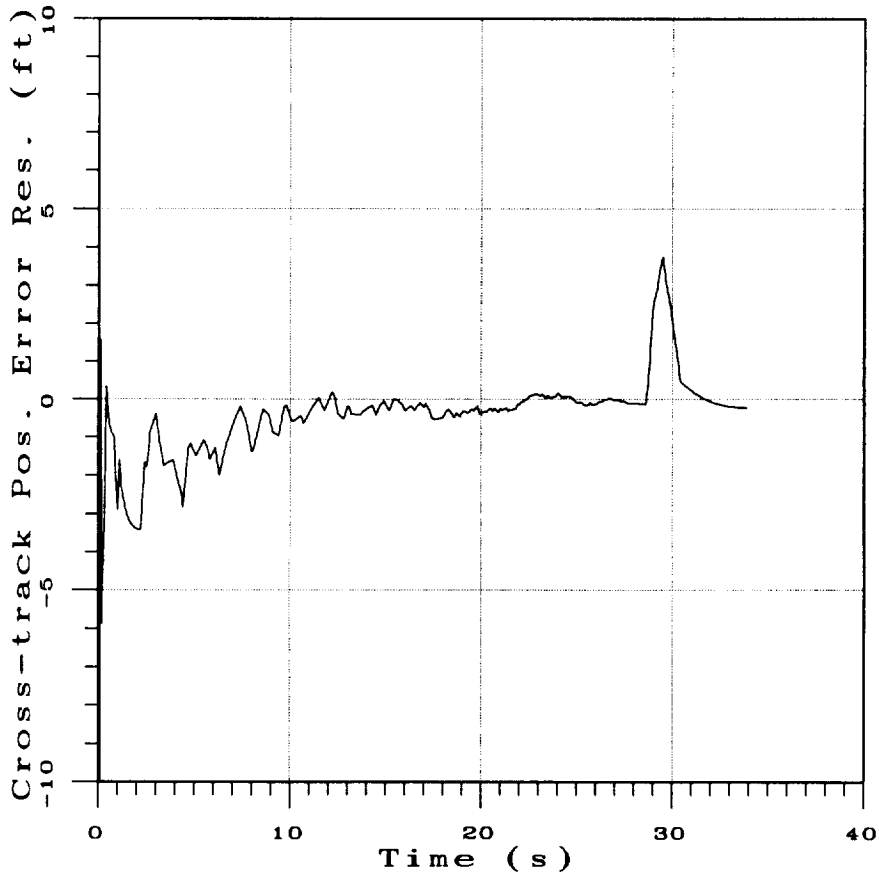

Figure 4.7: Cross-track position error using Algorithm I with $w_1 = 1$ and $w_2 = 1$.

The along-track velocity and cross-track velocity error residuals are illustrated

in Figures 4.8 and 4.9. Figure 4.8 shows that the along–track velocity estimation error never settles down. The cross–track velocity estimates on the other hand settle to within ±5 feet/second in less than one second.

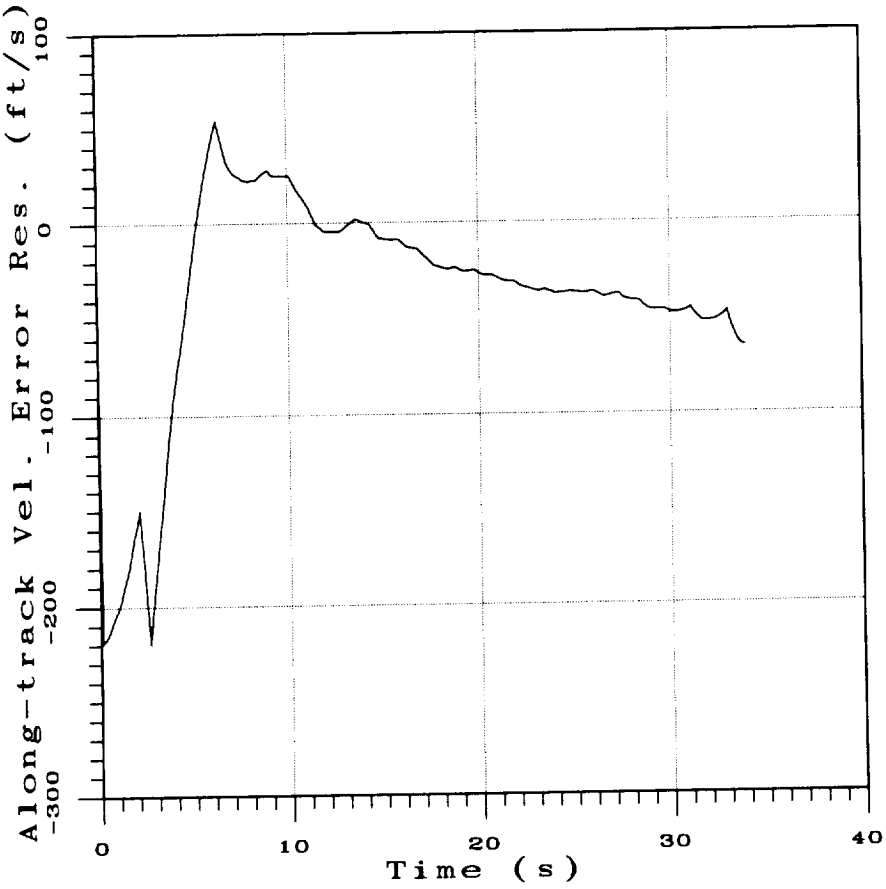

Figure 4.8: Along–track velocity error using Algorithm I with $w_1 = 1$ and $w_2 = 1$.

The cross-track position error residuals for FAA landing categories are summarized in Table 4.3. It can be observed that the navigation accuracy requirements for all the three categories are met. Note navigation accuracies for the landing categories are not specified for the along–track position.

Comparing the results of Subsections 4.1.1.1 and 4.1.1.2 suggests the weights $w_1 = 1$ and $w_2 = 0$ yields better position estimates.

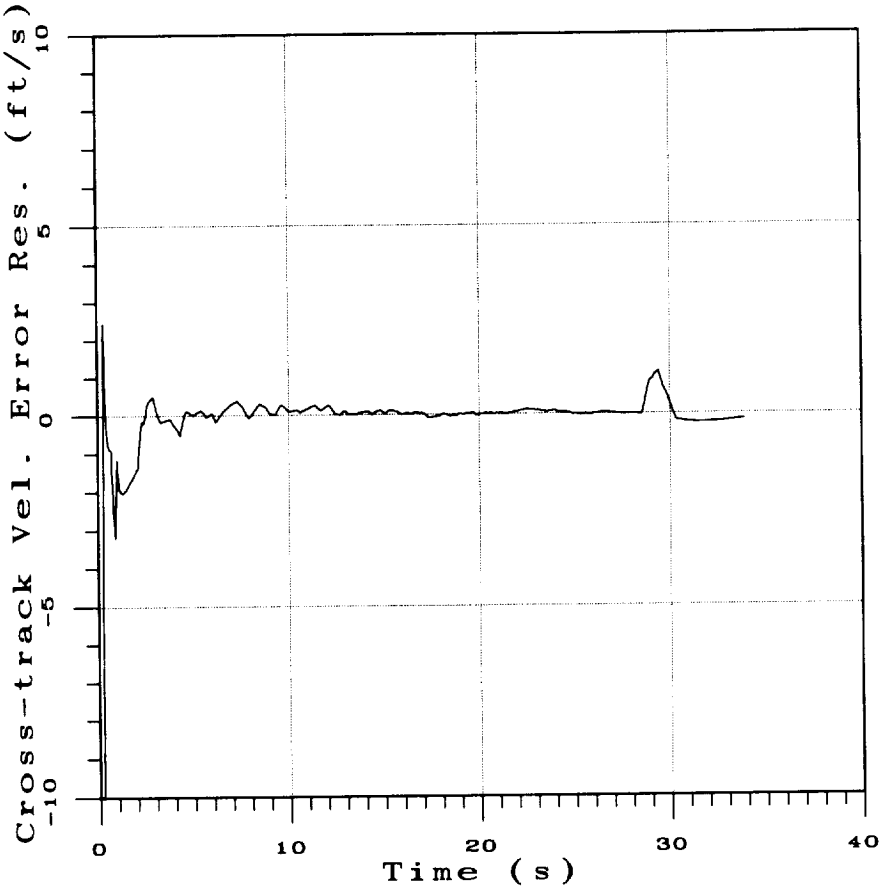

Figure 4.9: Cross-track velocity error using Algorithm I with $w_1 = 1$ and $w_2 = 1$.

Table 4.3: Algorithm I with $w_1 = 1$ and $w_2 = 1$ Results

Category	Lateral (y_b)
CAT I	±0.97
CAT II	±0.53
CAT IIIa	±3.73
CAT IIIb & c	±0.44

4.2 Algorithm II

The main difference between this algorithm and the previous one is that the present algorithm estimates all the three position components of the aircraft.

As in the previous algorithm, the minimum and maximum values x_{imin}, x_{imax}, y_{imin} and y_{imax} can be determined using the known x_i and y_i coordinates of the lights in the model. However, the lack of knowledge about the aircraft altitude makes the process of projecting the lights detected in the image onto the horizontal plane somewhat more complicated. The projection requires the assumed values of aircraft position components, x_b, y_b and h. The quantities s_{1_p} through s_{6_p} in these equations can be evaluated in exactly the same manner as in Algorithm I. The position coordinates x_p and y_p of every light can be found using the guessed initial position. These coordinates can then be used to find the values $x_{p_{min}}$, $x_{p_{max}}$, $y_{p_{min}}$ and $y_{p_{max}}$. Since the rectangle formed by the maximum and minimum values are required to enclose the projected and model lighting layouts, the position determination problem can be viewed as a parameter optimization problem for matching the projected rectangle with the rectangle enclosing the airport light database.

A quadratic cost function for measuring the matching error can be constructed in terms of the coordinates of the upper left and lower right corners of the enclosing rectangles, $(x_{p_{max}}, y_{p_{min}})$, (x_{imax}, y_{imin}), $(x_{p_{min}}, y_{p_{max}})$ and (x_{imin}, y_{imax}), as follows:

$$\begin{aligned} J &= \frac{w_1(x_{imin} - x_{p_{min}})^2 + w_2(x_{imax} - x_{p_{max}})^2}{\sqrt{(w_1^2 + w_2^2)}} \\ &+ \frac{(y_{imin} - y_{p_{min}})^2 + (y_{imax} - y_{p_{max}})^2}{\sqrt{2}} \end{aligned} \tag{4.24}$$

As in Algorithm I, the weights $0 \leq w_1 \leq 1$ and $0 \leq w_2 \leq 1$ can be used to establish the error contribution of lights near the camera and those that are far away. The objective of the optimization algorithm is to determine aircraft position components x_b, y_b and h that minimize the performance index J. Any one of the

several unconstrained optimization methods can be used for this purpose. The Newton-Raphson method is chosen for the present research.

The Newton–Raphson method [74] is an iterative method for finding the zeros of a function. In one dimension, the Newton–Raphson technique extrapolates the derivative at the current location in order to find an improved estimate of the zero. The method has its basis in the Taylor Series expansion of a function in the neighborhood of a point. It is known to converge quadratically when the function is smooth and convex. The Newton–Raphson technique can be readily extended to multiple dimensions. The technique is suitable for solution of nonlinear systems of equations. The technique can be adapted for finding the extremum of a function by driving the gradient vector of the function to zero. The difference between searching for a zero of a function and the zero of a derivative is that, in the first case the Jacobian matrix is used while in the second case the Hessian matrix is used.

Thus, the Newton–Raphson formula for position determination is:

$$\begin{bmatrix} x_b \\ y_b \\ h \end{bmatrix}_{n+1} = \begin{bmatrix} x_b \\ y_b \\ h \end{bmatrix}_n - H^{-1} \begin{bmatrix} \frac{\partial J}{\partial x_b} \\ \frac{\partial J}{\partial y_b} \\ \frac{\partial J}{\partial h} \end{bmatrix} \tag{4.25}$$

where, the Hessian matrix H is given by:

$$H = \begin{bmatrix} \frac{\partial^2 J}{\partial x_b^2} & \frac{\partial^2 J}{\partial y_b \partial x_b} & \frac{\partial^2 J}{\partial h \partial x_b} \\ \frac{\partial^2 J}{\partial x_b \partial y_b} & \frac{\partial^2 J}{\partial y_b^2} & \frac{\partial^2 J}{\partial h \partial y_b} \\ \frac{\partial^2 J}{\partial x_b \partial h} & \frac{\partial^2 J}{\partial y_b \partial h} & \frac{\partial^2 J}{\partial h^2} \end{bmatrix} \tag{4.26}$$

The subscript $n+1$ denotes the improved estimate of the aircraft inertial position. Starting from an initial guess, the position vector can be iteratively computed using Equation (4.25) until the change in the cost function is smaller than a preset tolerance.

Equation (4.25) requires the computation of the first and second partial derivatives of the cost function. Since the cost function depends on $(x_{p_{max}}, y_{p_{min}})$ and

$(x_{p_{min}}, y_{p_{max}})$ coordinates, their partial derivatives are required for computing the partial derivatives of the cost function with respect to aircraft position coordinates. These can be derived from Equations (4.13) and (4.14) as follows:

$$\frac{\partial x_{p_{max}}}{\partial x_b} = 1 \tag{4.27}$$

$$\frac{\partial x_{p_{max}}}{\partial y_b} = 0 \tag{4.28}$$

$$\frac{\partial x_{p_{max}}}{\partial h} = \frac{s_{2p}s_{6p} - s_{3p}s_{5p}}{s_{1p}s_{5p} - s_{2p}s_{4p}} \tag{4.29}$$

$$\frac{\partial x_{p_{min}}}{\partial x_b} = 1 \tag{4.30}$$

$$\frac{\partial x_{p_{min}}}{\partial y_b} = 0 \tag{4.31}$$

$$\frac{\partial x_{p_{min}}}{\partial h} = \frac{s_{2p}s_{6p} - s_{3p}s_{5p}}{s_{1p}s_{5p} - s_{2p}s_{4p}} \tag{4.32}$$

$$\frac{\partial y_{p_{max}}}{\partial x_b} = 0 \tag{4.33}$$

$$\frac{\partial y_{p_{max}}}{\partial y_b} = 1 \tag{4.34}$$

$$\frac{\partial y_{p_{max}}}{\partial h} = \frac{s_{1p}s_{6p} - s_{3p}s_{4p}}{s_{2p}s_{4p} - s_{1p}s_{5p}} \tag{4.35}$$

$$\frac{\partial y_{p_{min}}}{\partial x_b} = 0 \tag{4.36}$$

$$\frac{\partial y_{p_{min}}}{\partial y_b} = 1 \tag{4.37}$$

$$\frac{\partial y_{p_{min}}}{\partial h} = \frac{s_{1p}s_{6p} - s_{3p}s_{4p}}{s_{2p}s_{4p} - s_{1p}s_{5p}} \tag{4.38}$$

Note that these partial derivatives are evaluated at the instantaneous values of $x_{p_{max}}$, $x_{p_{min}}$, $y_{p_{max}}$ and $y_{p_{min}}$. Partial derivatives of the cost function with respect to aircraft position vector components can be computed analytically by using Equation (4.24) and Equations (4.27) through (4.38) as:

$$\frac{\partial J}{\partial x_b} = -\frac{2}{\sqrt{(w_1^2 + w_2^2)}}\left\{w_1(x_{i_{min}} - x_{p_{min}}) + w_2(x_{i_{max}} - x_{p_{max}})\right\} \tag{4.39}$$

$$\frac{\partial J}{\partial y_b} = -\sqrt{2}\left\{(y_{i_{min}} - y_{p_{min}}) + (y_{i_{max}} - y_{p_{max}})\right\} \tag{4.40}$$

$$\frac{\partial J}{\partial h} = -2\left\{\frac{w_1(x_{imin}-x_{p_{min}})\frac{\partial x_{p_{min}}}{\partial h}+w_2(x_{imax}-x_{p_{max}})\frac{\partial x_{p_{max}}}{\partial h}}{\sqrt{(w_1^2+w_2^2)}}\right.$$
$$\left.+\ \frac{(y_{imin}-y_{p_{min}})\frac{\partial y_{p_{min}}}{\partial h}+(y_{imax}-y_{p_{max}})\frac{\partial y_{p_{max}}}{\partial h}}{\sqrt{2}}\right\} \tag{4.41}$$

The elements of the Hessian matrix are given by:

$$\frac{\partial^2 J}{\partial x_b^2} = \frac{2(w_1+w_2)}{\sqrt{w_1^2+w_2^2}} \tag{4.42}$$

$$\frac{\partial^2 J}{\partial y_b \partial x_b} = 0 \tag{4.43}$$

$$\frac{\partial^2 J}{\partial h \partial x_b} = \frac{2}{\sqrt{w_1^2+w_2^2}}\left\{w_1\frac{\partial x_{p_{min}}}{\partial h}+w_2\frac{\partial x_{p_{max}}}{\partial h}\right\} \tag{4.44}$$

$$\frac{\partial^2 J}{\partial x_b \partial y_b} = \frac{\partial^2 J}{\partial y_b \partial x_b} \tag{4.45}$$

$$\frac{\partial^2 J}{\partial y_b^2} = 2\sqrt{2} \tag{4.46}$$

$$\frac{\partial^2 J}{\partial h \partial y_b} = \sqrt{2}\left\{\frac{\partial y_{p_{min}}}{\partial h}+\frac{\partial y_{p_{max}}}{\partial h}\right\} \tag{4.47}$$

$$\frac{\partial^2 J}{\partial x_b \partial h} = \frac{\partial^2 J}{\partial h \partial x_b} \tag{4.48}$$

$$\frac{\partial^2 J}{\partial y_b \partial h} = \frac{\partial^2 J}{\partial h \partial y_b} \tag{4.49}$$

$$\frac{\partial^2 J}{\partial h^2} = 2\left\{\frac{w_1\left(\frac{\partial x_{p_{min}}}{\partial h}\right)^2+w_2\left(\frac{\partial x_{p_{max}}}{\partial h}\right)^2}{\sqrt{(w_1^2+w_2^2)}}\right.$$
$$\left.+\ \frac{\left(\frac{\partial y_{p_{min}}}{\partial h}\right)^2+\left(\frac{\partial y_{p_{max}}}{\partial h}\right)^2}{\sqrt{2}}\right\} \tag{4.50}$$

Since the gradient vector and Hessian computations are analytic, the Newton–Raphson iterations can be carried out at a high computational rate. The essential steps involved in Algorithm II are summarized in Table 4.4.

The three aircraft position components estimated using Algorithm II are used as the measurements of the six–state position/velocity Kalman filter. As in Algorithm I, the Kalman filter integrates the information derived from multiple images

Table 4.4: Summary of Algorithm II

1. Compute $max\{x_i\}$, $min\{x_i\}$, $max\{y_i\}$ and $min\{y_i\}$ using the coordinates of N airport lights within the model. Here, $1 \leq i \leq N$.
2. Compute U_p and V_p $\forall$ p, such that $1 \leq p \leq M$ where, M is the number of lights detected in the image, using Equations (4.1) and (4.2).
3. Use ψ, θ and ϕ to compute r_1 through r_9 using Equation (3.3).
4. Compute s_{1_p} through s_{6_p} using Equations (4.7) through (4.12).
5. Assume x_b, y_b and h.
6. Using Equations (4.13) and (4.14), compute x_p and y_p $\forall$ p, such that $1 \leq p \leq M$.
7. Compute $max\{x_p\}$, $min\{x_p\}$, $max\{y_p\}$ and $min\{y_p\}$.
8. Compute cost J_2 using Equation (4.24).
9. If this is the first time, skip to step 11; else, continue.
10. Is $\mid J_2 - J_1 \mid \leq \delta$? If yes, stop; else, continue. The parameter δ is the stopping tolerance.
11. Set $J_1 = J_2$.
12. Compute the partial derivatives using Equations (4.39) through (4.50).
13. Using the Newton–Raphson formula, Equation (4.25), compute the inertial position components, x_b and y_b and h.
14. Return to step 6.

processed using Algorithm II. The six–state Kalman filter used here is described in Appendix A. The matrices required for this Kalman filter are available in Appendix B.

4.2.1 Results Using Algorithm II

Results of the two cases obtained using Algorithm II are described in this section. The first case is obtained using $w_1 = 1$ and $w_2 = 0$ while the second case is obtained using $w_1 = 1$ and $w_2 = 1$ in Equation (4.24). As in Algorithm I, the landing scenario discussed in Section 3.6 is used for both the cases. Errors of 1000 feet in the along–track position x_b, 100 feet in the cross–track position y_b and 100 feet in the altitude h are assumed for initializing the Kalman filter. The inertial velocity components are all initialized to zero.

4.2.1.1 Algorithm II with $w_1 = 1$ and $w_2 = 0$

The along–track position error residual is shown in Figure 4.10. It may be seen from the figure that the along–track position estimate converges to within ± 100 feet in less than one second.

The cross–track position error residual presented in Figure 4.11 shows that the cross–track position estimates converge to within ± 5 feet in less than three seconds.

The altitude error residual is shown in Figure 4.12. It may be observed from the figure that the altitude error converges to ± 5 feet within two seconds and stays within these bounds up to 30 seconds. Beyond that point the altitude error increases. The error increase can be attributed to the reduction in the number of lights within the field–of–view. The altitude of the aircraft at 30 seconds is 55 feet.

Figures 4.13, 4.14 and 4.15 show the along–track velocity (v_{b_x}), the cross–track velocity (v_{b_y}) and the sink rate ($-v_{b_z}$) error residuals. It may be observed from the figures that the along–track velocity estimate settles to within ± 10 feet/second in less than five seconds, the cross–track velocity settles to within ± 5 feet/second

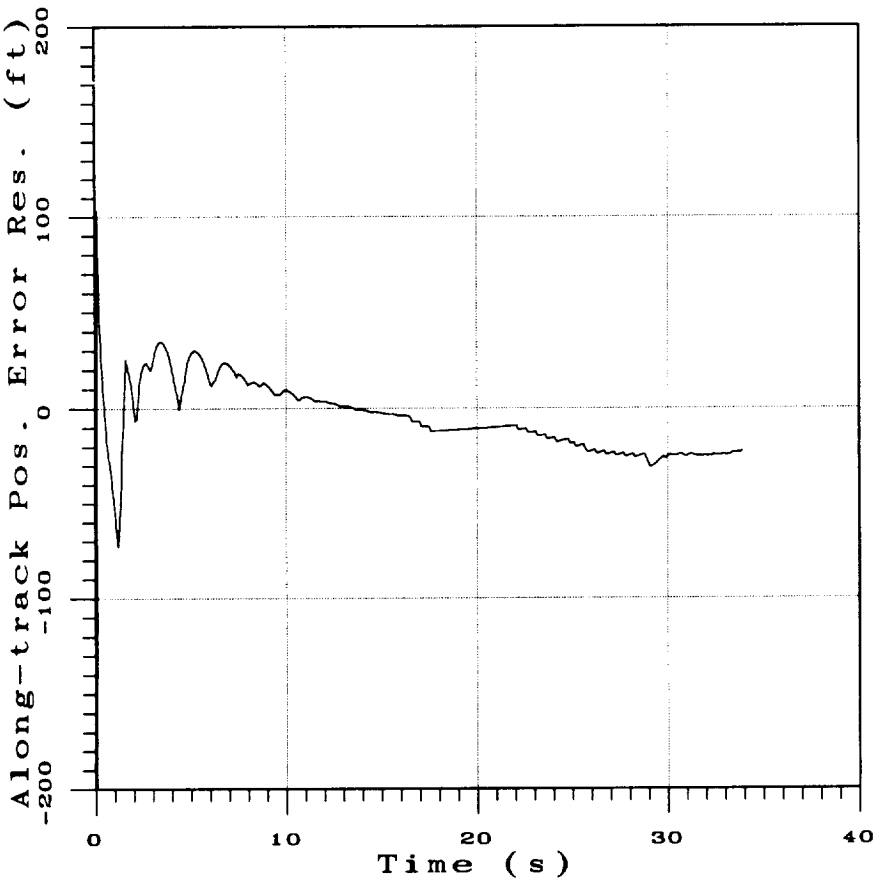

Figure 4.10: Along-track position error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

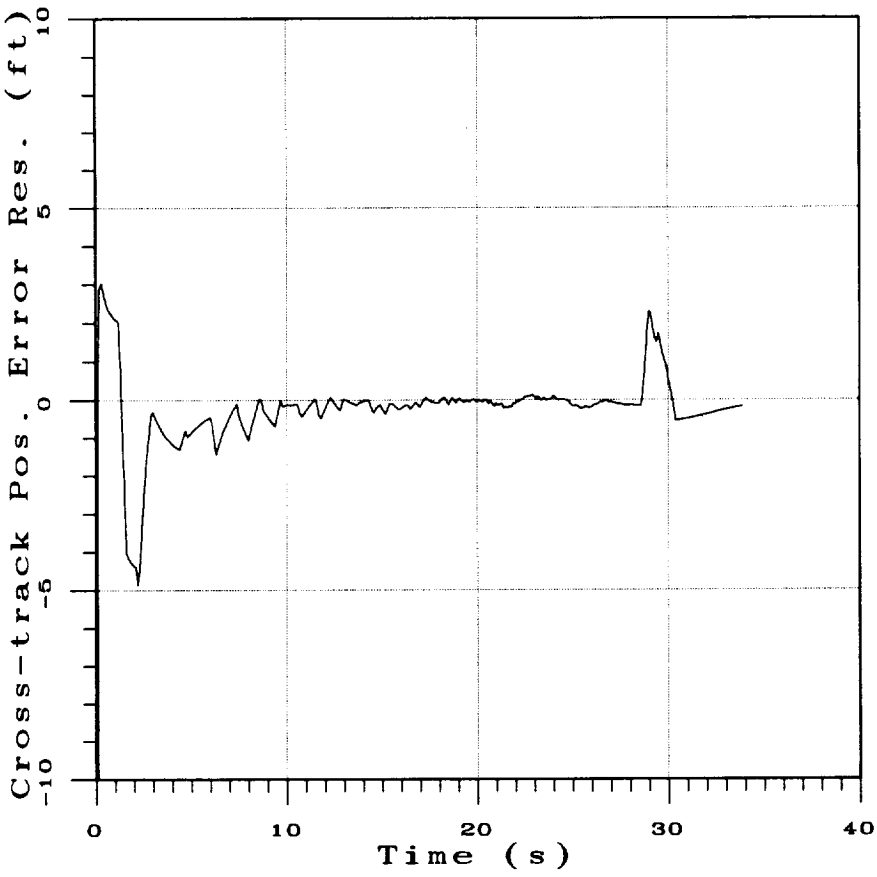

Figure 4.11: Cross-track position error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

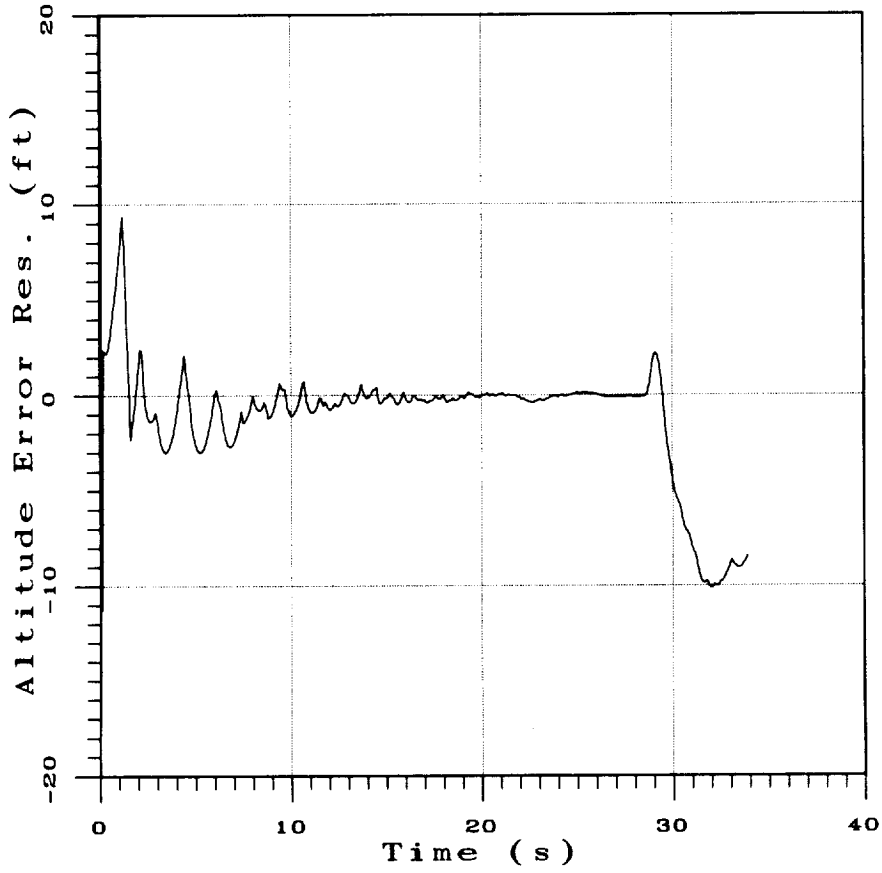

Figure 4.12: Altitude error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

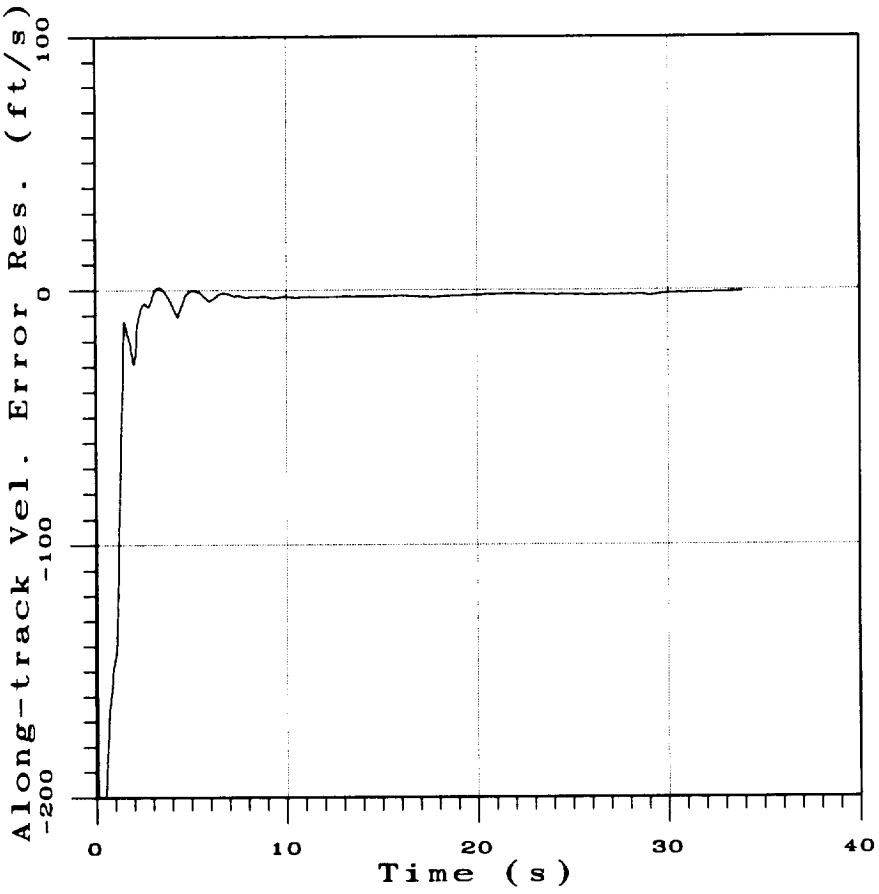

Figure 4.13: Along-track velocity error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

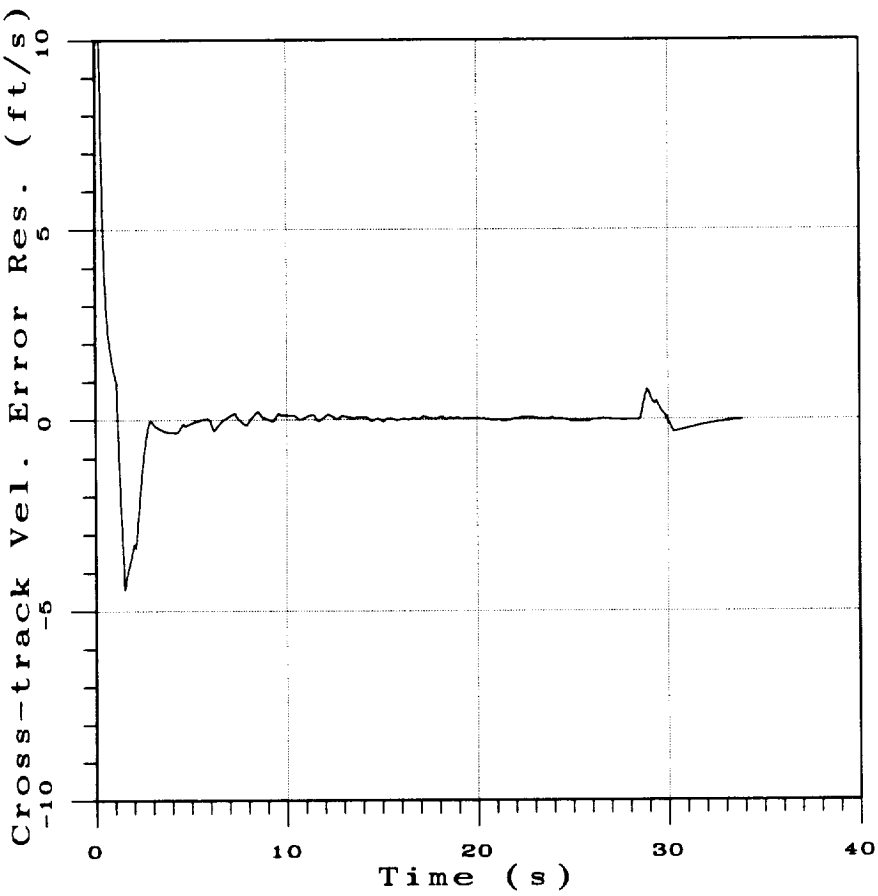

Figure 4.14: Cross–track velocity error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

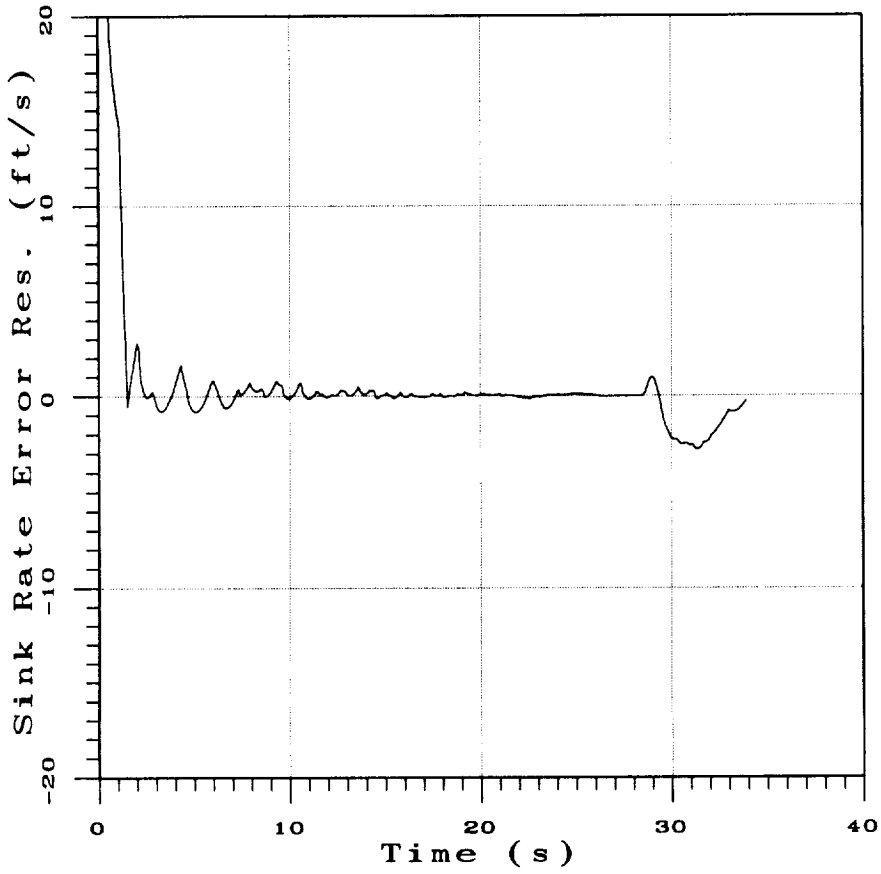

Figure 4.15: Sink rate error using Algorithm II with $w_1 = 1$ and $w_2 = 0$.

in less than one second and the sink rate settles to within ±5 feet/second in less than two seconds. These results are comparable to those obtained using Algorithm I for the same set of weights.

Table 4.5 lists the position error residuals. In Table 4.5, the position components are in feet. Comparing Table 4.5 with Table 3.2, it may be seen that

Table 4.5: Algorithm II with $w_1 = 1$ and $w_2 = 0$ Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±0.70	±1.21
CAT II	±0.22	±0.40
CAT IIIa	±2.30	±5.82
CAT IIIb & c	±0.55	±10.11

Algorithm II along with the Kalman filter generates position estimates which meet the navigation accuracy requirements for Categories I and II.

4.2.1.2 Algorithm II with $w_1 = 1$ and $w_2 = 1$

The error residuals of the runway relative position and velocity components obtained using Algorithm II with $w_1 = 1$ and $w_2 = 1$ are described in this section.

Figure 4.16 shows that the along–track position estimates converge to within ±100 feet in about six seconds.

The cross–track position error residual shown in Figure 4.17 shows that the cross–track position estimates converge to within ±5 feet in less than one second.

The altitude error residual given in Figure 4.18 shows that the altitude estimates converge to within ±20 feet in less than six seconds.

By comparing Figure 4.16 with Figure 4.6 it may be noted that the along–track position is significantly improved by Algorithm II at the cost of increased altitude estimation error shown in Figure 4.18.

The along–track velocity, cross–track velocity and sink rate error residuals are illustrated in Figures 4.19, 4.20 and 4.21. Figure 4.19 shows that the along–

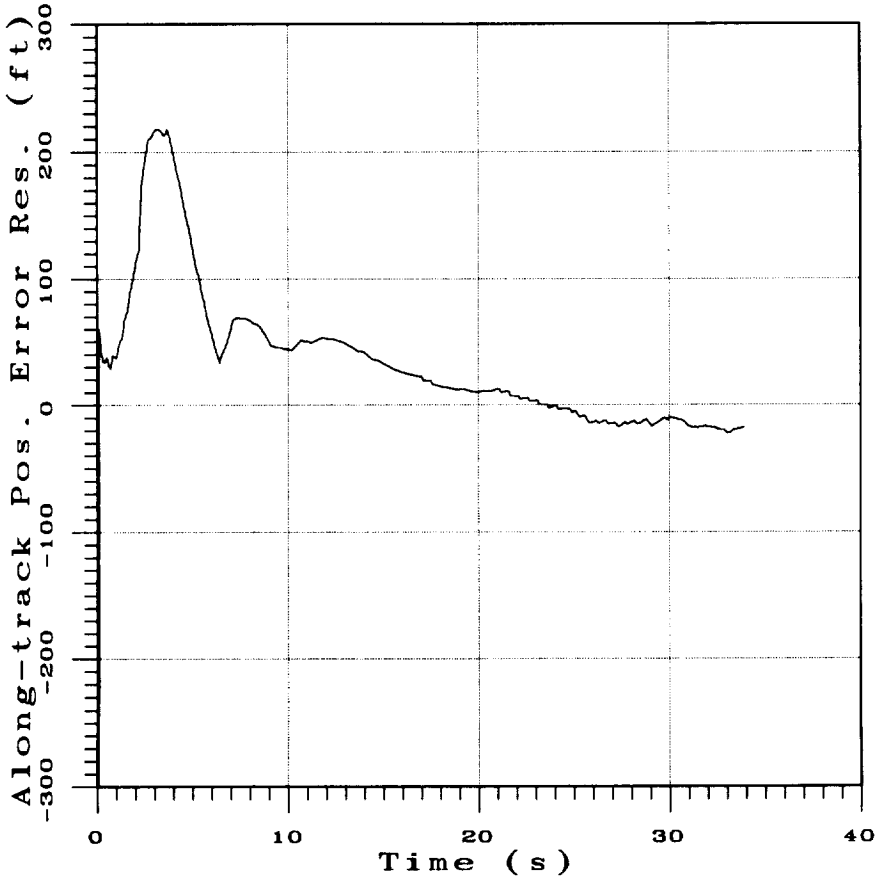

Figure 4.16: Along-track position error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

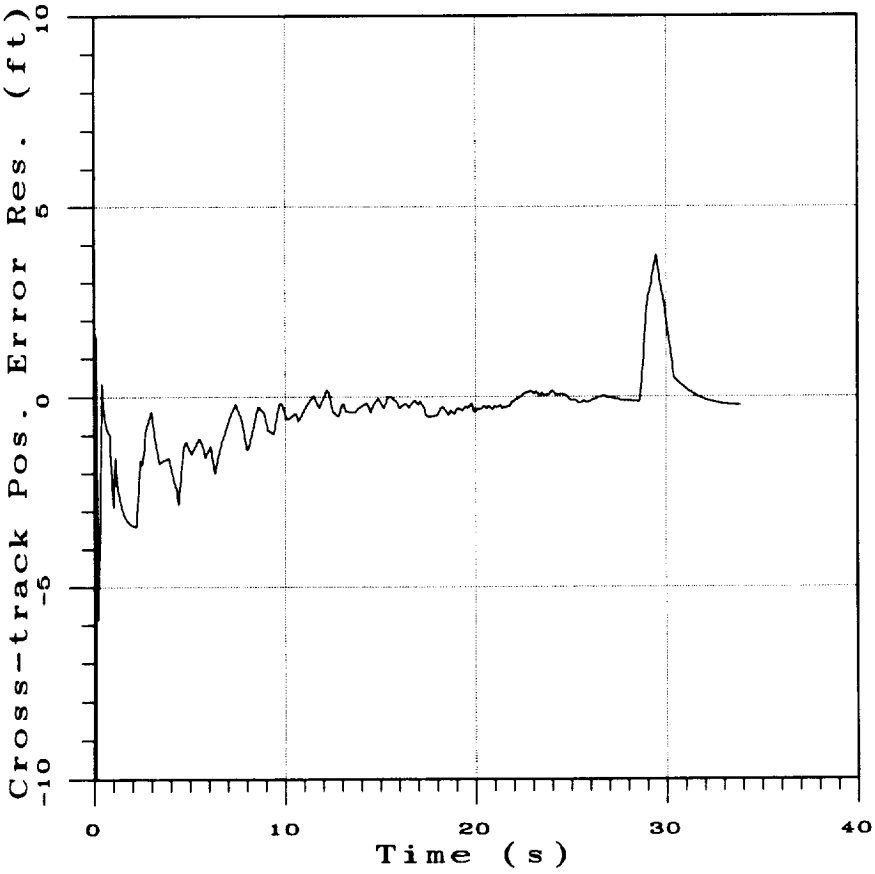

Figure 4.17: Cross–track position error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

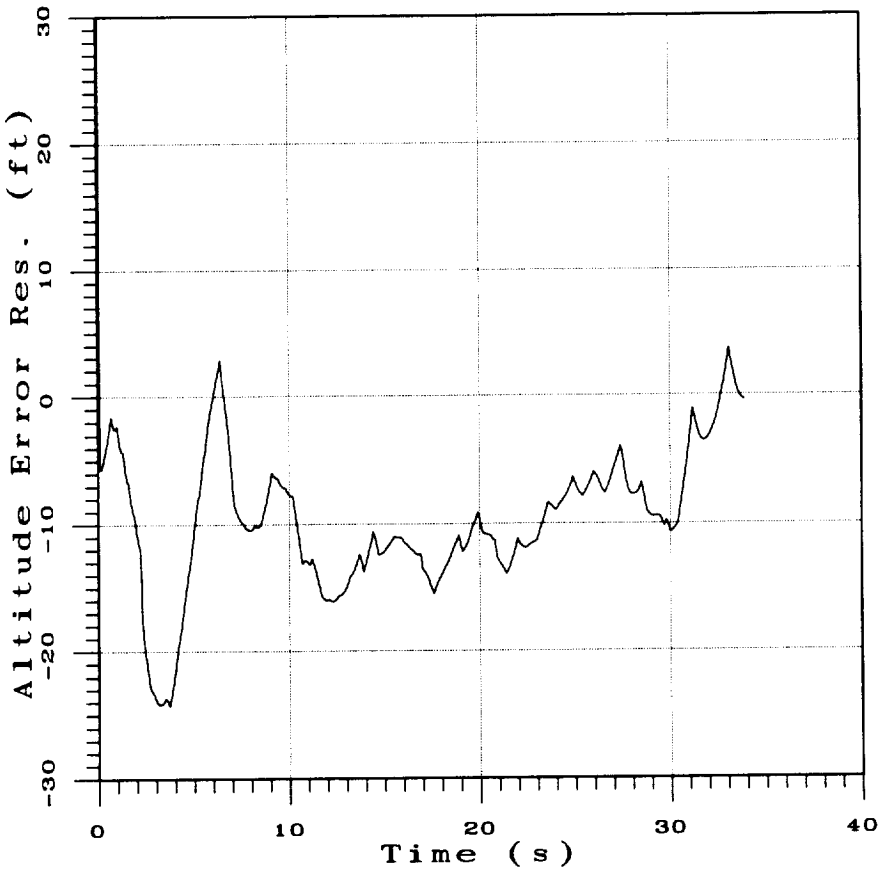

Figure 4.18: Altitude error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

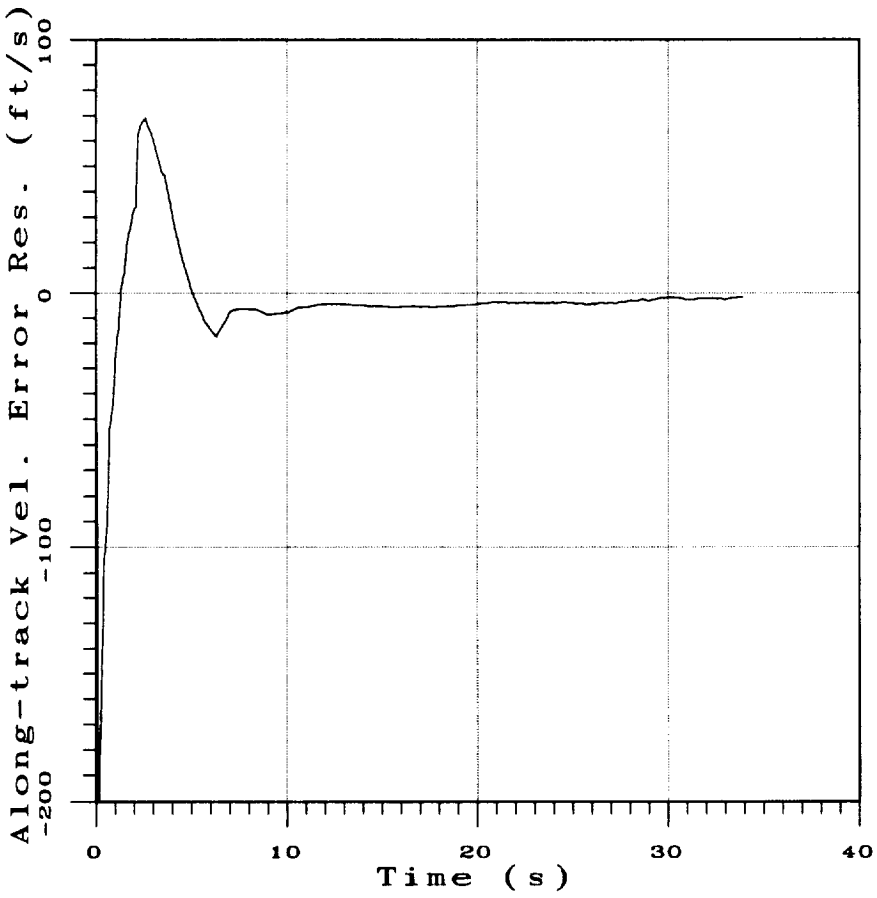

Figure 4.19: Along–track velocity error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

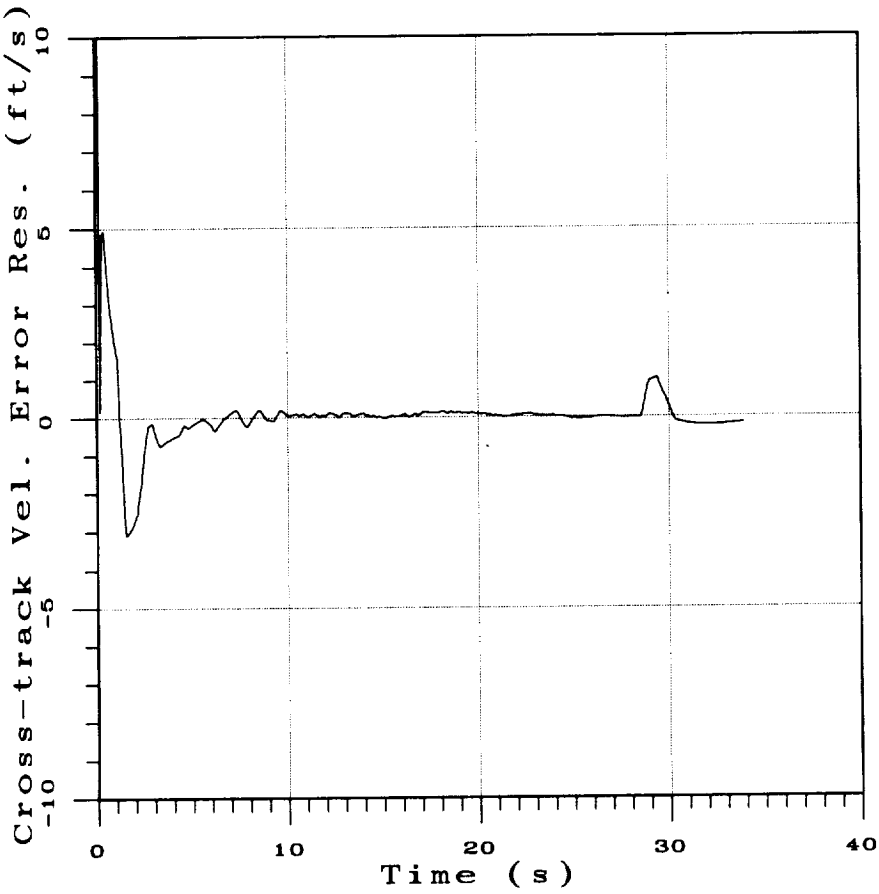

Figure 4.20: Cross-track velocity error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

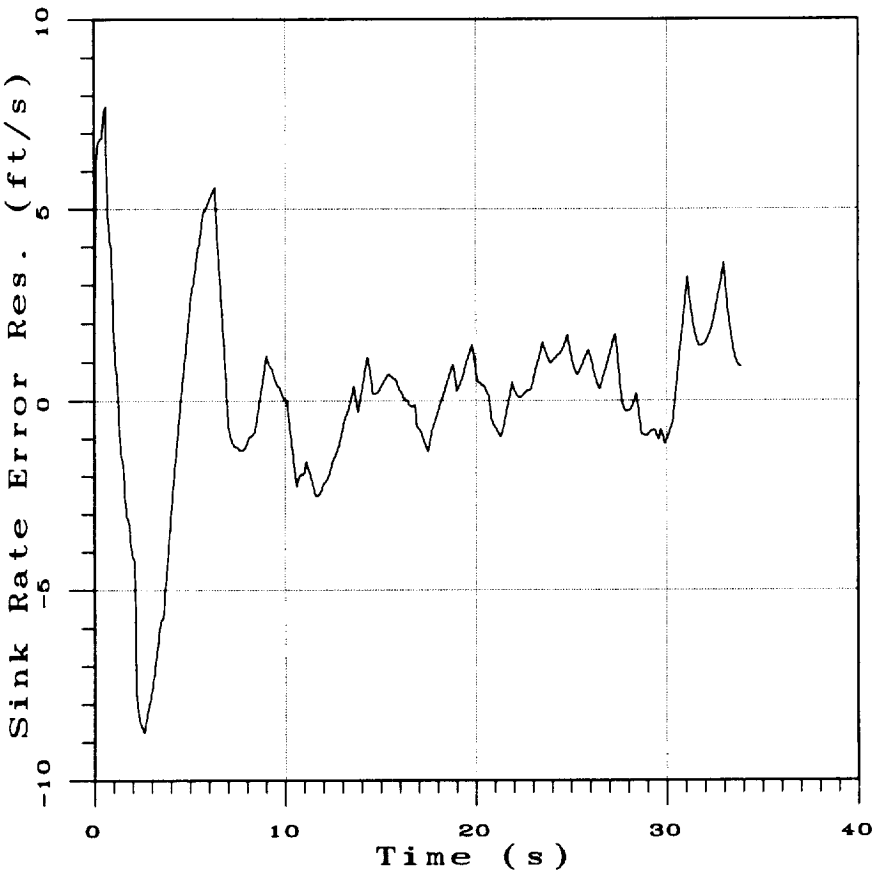

Figure 4.21: Sink rate error using Algorithm II with $w_1 = 1$ and $w_2 = 1$.

track velocity settles down to within ±10 feet/second in less than five seconds. The cross–track velocity estimates settle to within ±5 feet/second in less than one second. The sink rate estimates settle to within ±5 feet/second in less than five seconds.

The position error residuals for ranges of altitudes are summarized in Table 4.6. The nomenclature for this table is same as that used in Table 4.5. Table 4.6

Table 4.6: Algorithm II with $w_1 = 1$ and $w_2 = 1$ Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±1.21	±16.11
CAT II	±0.19	±15.50
CAT IIIa	±3.52	±10.72
CAT IIIb & c	±0.62	±9.01

shows that in this case, the navigation accuracy requirements listed in Table 3.2 are not met for any of the categories because of the reduced altitude estimation accuracy.

As in Algorithm I, the results of these two cases indicate that Algorithm II should be used with $w_1 = 1$ and $w_2 = 0$ weight combination.

4.3 Summary

Two methods for aircraft position estimation were described in this chapter. Both are based on envelope matching in the inertial frame. The first method required the altitude in addition to the yaw, pitch and roll orientation angles. The second method did not require knowledge of aircraft altitude. Both the algorithms were based on the inverse perspective projection equations which relate the image coordinates of the airport lights to their inertial coordinates via the aircraft inertial position components. The problem of position determination was posed as a problem of parameter optimization with the aircraft inertial position components

as the parameters. In Algorithm I, the two position components were determined by matching the envelope of the predicted lighting geometry with the envelope of the stored lighting geometry model. In Algorithm II, a quadratic cost function was used for minimizing the envelope matching error using a Newton–Raphson method. The position components computed by these algorithms were used as measurements for a six–state Kalman filter for estimating the position and velocity components along the glide slope. Results obtained for the four cases were discussed. It was shown that both Algorithm I and Algorithm II generate more accurate estimates with $w_1 = 1$ and $w_2 = 0$ weight combination. This implies that these image based algorithms should place a higher weight on nearer lights, than on lights farther away along–track.

Chapter 5

Feature Correspondence Based Aircraft Position Estimation Methods

Two position determination methods were presented in the previous chapter. The central idea there was to match the observed lighting layout to the model of the airport lighting layout. One of the difficulties with the algorithms described in Chapter 4 is that the cost function is based only on the coordinates of two corners of the envelope of the airport lights. Thus, they do not exploit all the information available in the image. The objective of the methods presented in this chapter is to formulate methods that use most of the information available in the image.

Reason for using information from many lights as opposed to a few is obvious. An algorithm using information from multiple lights can be expected to be less sensitive to camera induced errors. Additionally, the estimation accuracy will remain unaffected even if some of the lights have failed. Algorithms presented in this chapter do not require any iterative computations. Significant benefits of such direct schemes are that they are computationally efficient and robust.

The main idea in this chapter is to synthesize measures that capture the structure of the airport lighting layout in terms of scalar functions. Such measures

can be used as features for establishing the correspondence between the observed lighting layout and the model lighting layout. The important structures in the standard airport lighting layout were presented in Figures 3.7 and 3.8. Of these, the prominent ones are centerline, left and right edge lights and the threshold bar. It is possible to model these structures as straight lines using the Hough transform method discussed in Subsection 2.3.1. However, the Hough transform requires careful threshold selection to prevent phantom lines from being detected. Another approach for discovering the structure is to simply arrange the coordinates of the airport lights in a non–decreasing order. This arrangement causes the lights to be re–indexed so that their inertial coordinates are in a non–decreasing order. Any efficient method can be used for arranging. The Quicksort method [56] has been used in this research. For example, Figure 5.1 shows the y–coordinates of the airport lights arranged in a non–decreasing order.

The basis for arranging the y–coordinates in a non-decreasing order lies in the fact that the left edge lights have a y–coordinate value of -100 feet, the centerline lights have a y–coordinate of zero feet and the right edge lights have a y–coordinate of 100 feet as may be seen in Figures 3.7 and 3.8. The graph shown in Figure 5.1 is obtained by plotting the inertial y–coordinates of the re–arranged lights against their scaled indices. The scaled index for a light is obtained by normalizing its re–ordered index by the total number of lights in the model and multiplying the result by 100. Note that the scaled index is no longer an integer, but a rational number. The multiplication factor of 100 is chosen for convenience. This choice of multiplication factor results in the left edge lights having scaled indices with values less than 15 as may be seen in Figure 5.1. The range of scaled indices for the centerline lights and the right edges lights are also marked in Figure 5.1. The benefit of scaling is that the observed lighting layout is made comparable to the model lighting layout. Note that usually fewer lights are observed in the image compared to those available in the airport lighting model.

The inertial along–track x–coordinates of the model lights arranged in a non-

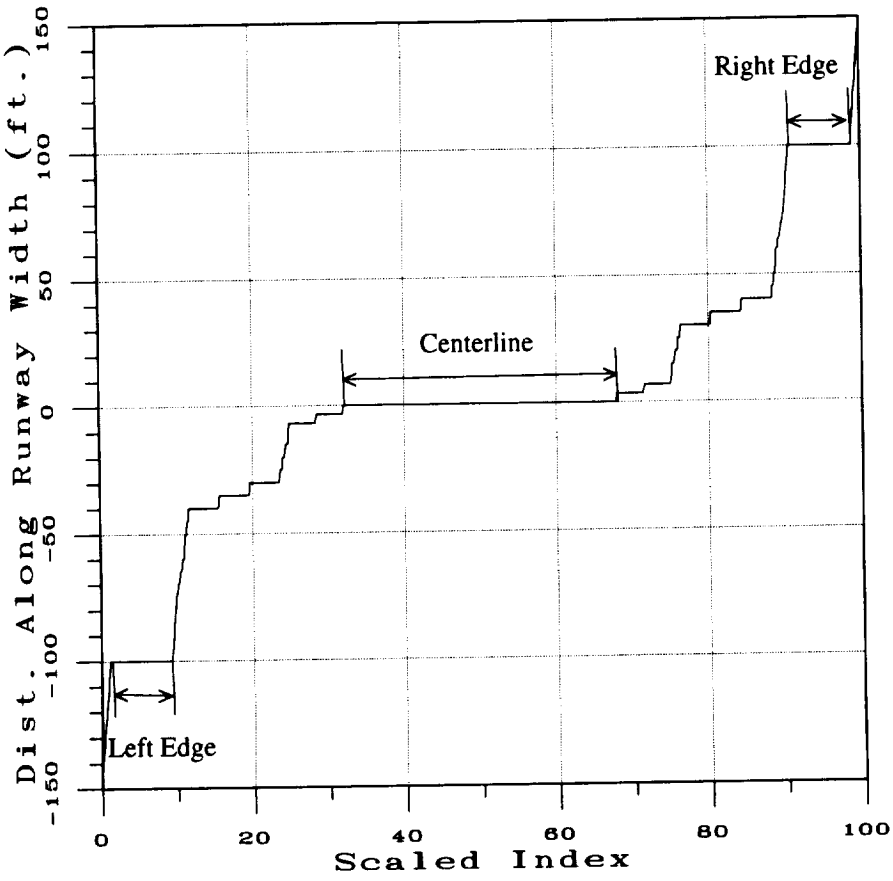

Figure 5.1: The cross–track y-coordinates of the runway lights in a non-decreasing order.

decreasing order are shown in Figure 5.2. From the standard approach and runway

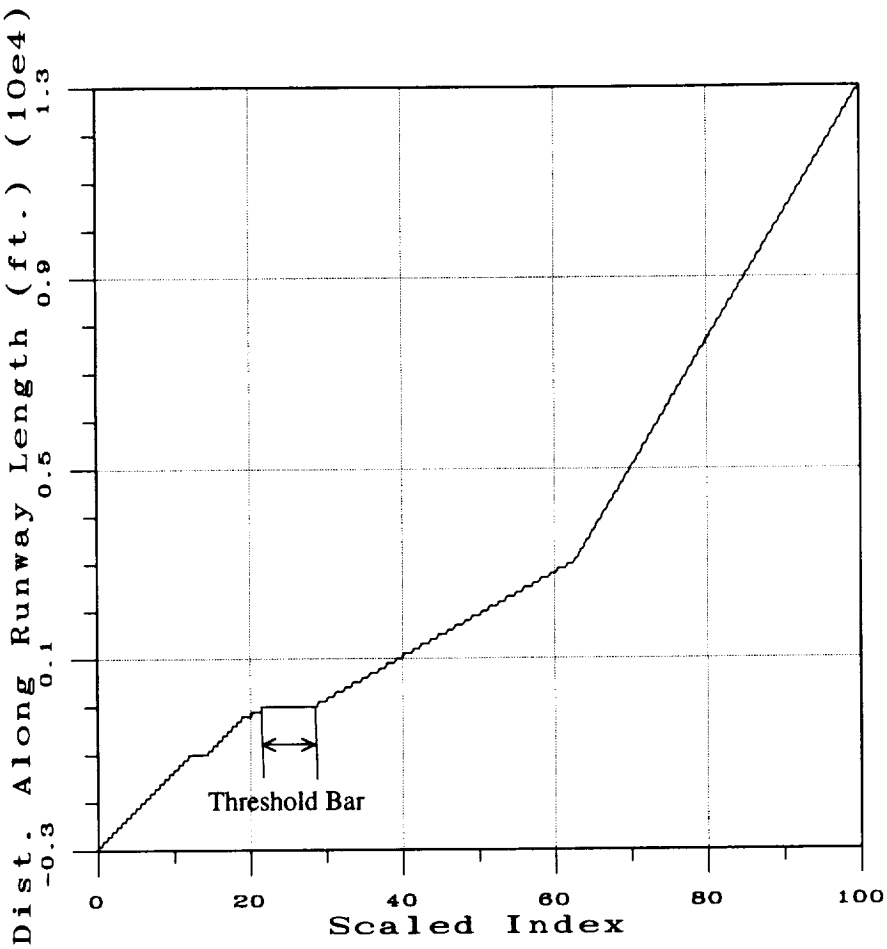

Figure 5.2: Along-track x-coordinates of the airport lights in a non-decreasing order.

lighting layouts shown in Figures 3.7 and 3.8, it may be observed that the x-coordinates of the threshold bar lights have a common value of zero. Figure 5.2 shows that threshold lights have scaled indices between 20 and 30. Other structures are difficult to identify in this graph because the x-coordinates of structures like centerline lights and edge lights are mixed together along the length of the runway. These structures appear prominently when their y-coordinates are arranged as shown in Figure 5.1.

Since the observed and model lighting have the same range of scaled indices,

the corresponding structures are related by the range of indices. For example, the left edge lights in both the observed and model lighting layouts are expected to have scaled indices with values between 0 and 15. Thus, this range of scaled indices can be used for establishing the correspondence between features based on the left edge lights identified in both the layouts.

Two position determination methods that exploit the ability to identify structures in the graphs such as those in Figures 5.1 and 5.2 are described in this chapter. Both the algorithms belong to the first family of solution methods described in Figure 3.11. The first algorithm assumes that the altitude is known while the second algorithm does not make this assumption. Both algorithms assume that the aircraft attitudes, ψ, θ and ϕ, are known. Starting points for both the algorithms are the relationship between the aircraft position coordinates and the position of the lights given by the Equations (4.13) and (4.14). Thus, like the earlier algorithms, they are not designed to account for the varying elevation of the airport lights. The details of these algorithms are given in the following sections.

5.1 Algorithm III

Algorithm III assumes that in addition to the aircraft attitudes, ψ, θ and ϕ, the aircraft altitude is available from an onboard altimeter. This assumption results in decoupling of Equations (4.13) and (4.14). Two such equations can be written for each light observed in the image. Adding the resulting expressions:

$$\sum_{1\leq p\leq M} x_p = Mx_b + h \sum_{1\leq p\leq M} \frac{(s_{2p}s_{6p} - s_{3p}s_{5p})}{(s_{1p}s_{5p} - s_{4p}s_{2p})} \tag{5.1}$$

$$\sum_{1\leq p\leq M} y_p = My_b + h \sum_{1\leq p\leq M} \frac{(s_{1p}s_{6p} - s_{3p}s_{4p})}{(s_{2p}s_{4p} - s_{1p}s_{5p})} \tag{5.2}$$

Here M is the number of lights. In the ideal case, the location of the lights in the airport lighting model x_i and y_i should be same as that observed. That is:

$$\sum_{1\leq i\leq M} x_i = \sum_{1\leq p\leq M} x_p \tag{5.3}$$

$$\sum_{1\leq i\leq M} y_i = \sum_{1\leq p\leq M} y_p \tag{5.4}$$

Thus,

$$\sum_{1\leq i\leq M} x_i = Mx_b + h \sum_{1\leq p\leq M} \frac{(s_{2p}s_{6p} - s_{3p}s_{5p})}{(s_{1p}s_{5p} - s_{4p}s_{2p})} \tag{5.5}$$

$$\sum_{1\leq i\leq M} y_i = My_b + h \sum_{1\leq p\leq M} \frac{(s_{1p}s_{6p} - s_{3p}s_{4p})}{(s_{2p}s_{4p} - s_{1p}s_{5p})} \tag{5.6}$$

In reality, the left hand sides are known exactly from the airport lighting model, while the right hand sides are not known exactly due to observation errors in the image. For example, perspective projection causes distant edge lights and centerline lights merge together to form lines in the image in Figure 2.2. Due to this and other effects, the exact location of lights in the image plane cannot be accurately ascertained. In order to make the left and right sides comparable, the left hand side is normalized with the number of N lights in the airport lighting model, while the right hand sides are normalized using the number of M distinct lights observed in the image. Thus,

$$\frac{1}{N}\sum_{1\leq i\leq N} x_i = x_b + \frac{h}{M} \sum_{1\leq p\leq M} \frac{(s_{2p}s_{6p} - s_{3p}s_{5p})}{(s_{1p}s_{5p} - s_{4p}s_{2p})} \tag{5.7}$$

$$\frac{1}{N}\sum_{1\leq i\leq N} y_i = y_b + \frac{h}{M} \sum_{1\leq p\leq M} \frac{(s_{1p}s_{6p} - s_{3p}s_{4p})}{(s_{2p}s_{4p} - s_{1p}s_{5p})} \tag{5.8}$$

In this form, the inertial aircraft position components x_b and y_b can be computed simply by subtracting the image based arithmetic means from the model based arithmetic means.

Results obtained using Algorithm I and II in Chapter 4 suggested that only the nearer lights should be used for obtaining the along-track position of the aircraft. Therefore, only N_1 nearer model lights and M_1 nearer image lights should be used in Equation (5.7). Note that $N_1 < N$ and $M_1 < M$. Using the nearer lights with scaled indices between zero and 40, Equation (5.7) can be re-written as:

$$\frac{1}{N_1}\sum_{1\leq i\leq N_1} x_i = x_b + \frac{h}{M_1} \sum_{1\leq p\leq M_1} \frac{(s_{2p}s_{6p} - s_{3p}s_{5p})}{(s_{1p}s_{5p} - s_{4p}s_{2p})} \tag{5.9}$$

It may be observed from Figure 5.2 that the approach lights and the threshold bar lights in the model have scaled indices between zero and 40. The cross-track component of the aircraft position can be obtained by using all the model and image lights in Equation (5.8).

An additional fact that deserves consideration is that only a portion of the airport lighting is visible within the camera field-of-view as the aircraft follows its descent trajectory. Thus, the comparison between model lighting and actual lighting observed in the image may yield incorrect position estimates. In order to partially offset the errors caused by this, the position estimation algorithm employs the previously estimated aircraft position to determine the model lights within the field-of-view. This information is then used in the calculations using Equations (5.8) and (5.9).

This does not cause any problems for initialization because the complete airport lighting is visible to the camera at the beginning of the descent path even with substantial differences between the actual and assumed aircraft positions.

The modified airport lighting model x and y coordinates arranged in a non-decreasing order are shown in Figures 5.3 and 5.4. These figures were generated using an assumed along-track position which was in error by 1000 feet compared to the true aircraft position. No errors were assumed in the cross-track position and altitude. It may be observed from the figures that the threshold bar lights, edge lights and the centerline lights are correctly represented in the model layout graphs. The figures also show the observed lighting x and y coordinates arranged in a non-decreasing order. Since the observed lighting is with respect to the aircraft, the true along-track and cross-track positions of the aircraft can be obtained by shifting the observed layout graphs to the model layout graphs. Figures 5.3 and 5.4 show that the true along-track and cross-track positions are -6633 feet and 0 feet with respect to the inertial coordinate system.

Algorithm III is summarized in Table 5.1.

Some of the salient features of Algorithm III are as follows. Firstly, since

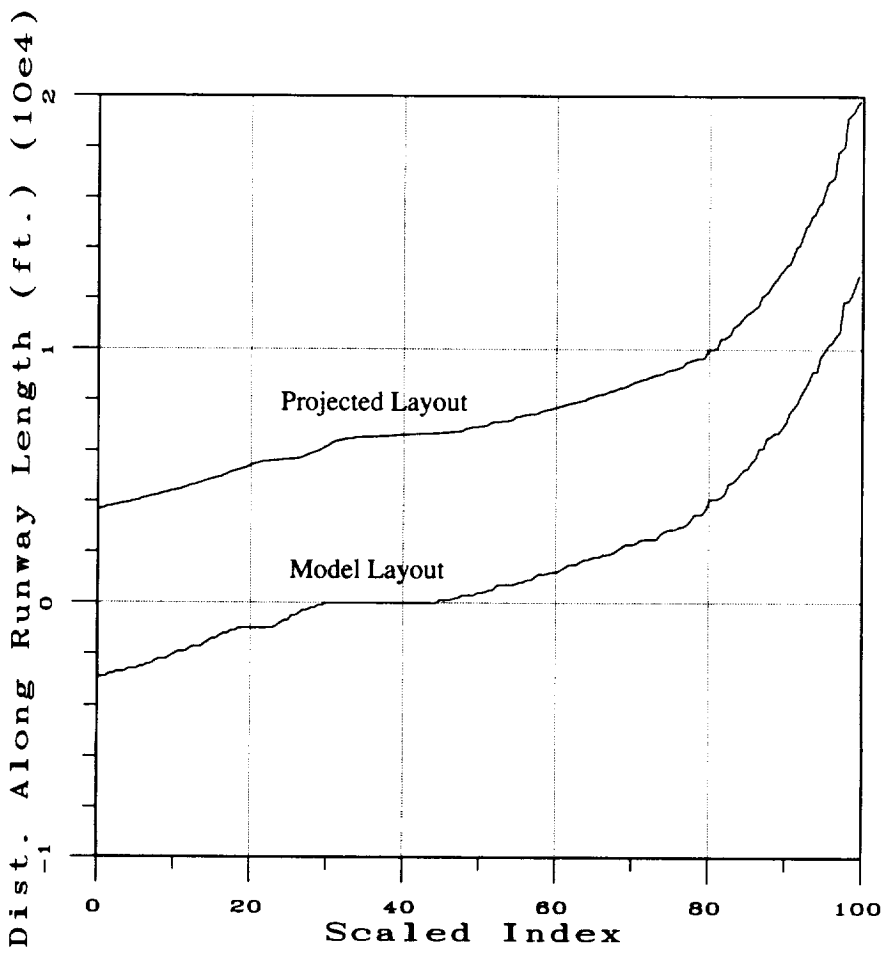

Figure 5.3: x–coordinates of adapted model and image lights in a non–decreasing order.

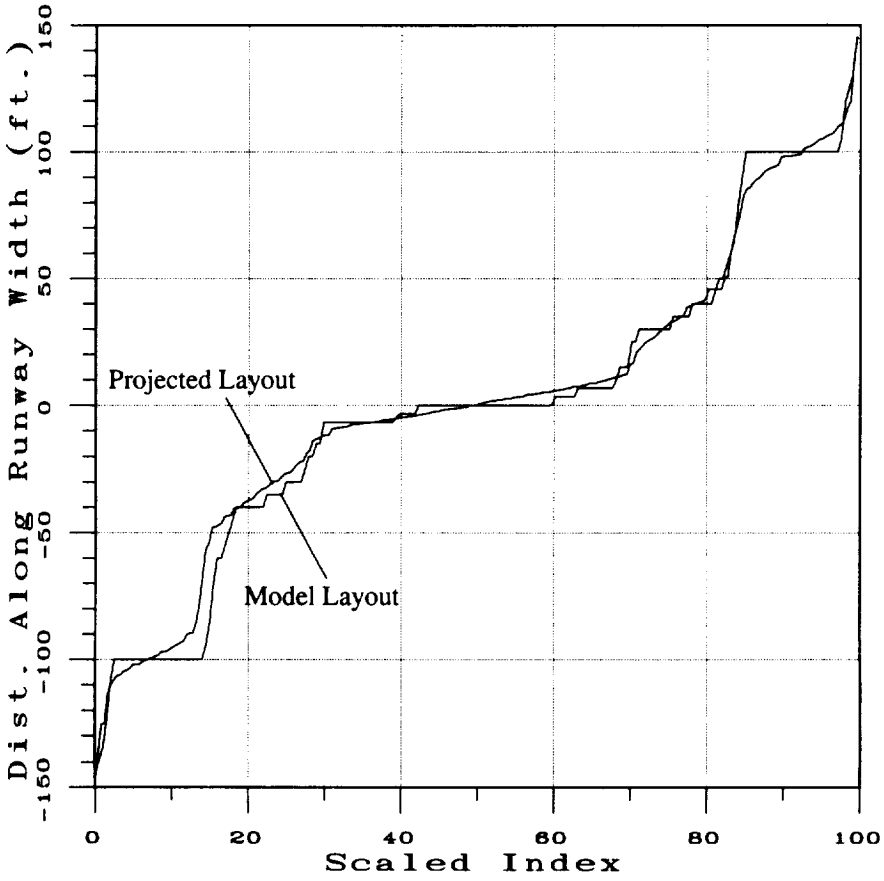

Figure 5.4: y-coordinates of adapted model and image lights in a non-decreasing order.

Table 5.1: Summary of Algorithm III

1. Arrange x_i coordinates of all the N lights from the airport lighting geometry model in non–decreasing order.

2. Compute the scaled indices $k_i = 100i/N$; $1 \leq i \leq N$.

3. Select the ordered x_i $\forall$ i such that $0 \leq k_i \leq 40$. Compute the mean of the N_1 selected x_i in Equation (5.9).

4. Compute the mean using all y_i, $1 \leq i \leq N$, in Equation (5.8).

5. For every light in the scene compute U_p and V_p using Equations (4.1) and (4.2) with $p = 1, 2, \ldots, M$ where, M is the number of lights detected in the image.

6. Use ψ, θ and ϕ to compute r_1 through r_9 using Equation (3.3).

7. Compute s_{1p} through s_{6p} using Equations (4.7) through (4.12).

8. Compute $\tilde{x}_p = (s_{2p}s_{6p} - s_{3p}s_{5p})/(s_{1p}s_{5p} - s_{4p}s_{2p})$
   and $\tilde{y}_p = (s_{1p}s_{6p} - s_{3p}s_{4p})/(s_{2p}s_{4p} - s_{1p}s_{5p})$ $\forall p$; $1 \leq p \leq M$.

9. Arrange $\tilde{x}_p$ coordinates of M image lights in non–decreasing order.

10. Compute the scaled indices $k_p = 100p/M$; $1 \leq p \leq M$.

11. Select the ordered $\tilde{x}_p$ $\forall$ p such that $0 \leq k_p \leq 40$. Compute the mean of the M_1 selected $\tilde{x}_p$ in Equation (5.9).

12. Compute the mean using all $\tilde{y}_p$, $1 \leq p \leq M$, in Equation (5.8).

13. Use the known altitude h in Equations (5.9) and (5.8) and subtract the image–based means from the model–based means in these equations to compute the aircraft position components x_b and y_b.

the method uses information from multiple airport lights, it is more robust when compared with the envelope matching methods discussed in the last chapter. Aircraft position is determined without resorting to the traditional correspondence methods such as the local correlation methods.

Major limitations of the algorithm are that it is unable to account for the runway elevation and that knowledge of aircraft altitude is required. Due to the importance of accurate altitude information during landing, it is desirable to modify this algorithm to provide the altitude information in addition to the lateral and longitudinal positions. This is the motivation for the next algorithm for aircraft position determination described in Section 5.2.

As in the case of Algorithms I and II, the position measurements from individual frames can be made consistent with the aircraft kinematics using a six–state Kalman filter described in Appendix A.

5.1.1 Results Using Algorithm III

The position and velocity estimates generated using the six–state Kalman filter driven by the measurements from Algorithm III are described in this section. These estimates were obtained using the aircraft landing flight trajectory and the images along the glide slope simulated using the procedures described in Section 3.6. Errors of 1000 feet in the along–track position and 100 feet in the cross–track position are assumed for initializing the Kalman filter. The inertial velocity components are all initialized to zero.

The along–track position error residual is shown in Figure 5.5. It may be seen from the figure that the along–track position estimate converges to within ± 100 feet in less than one second.

The cross–track position error residual presented in Figure 5.6 shows that the cross–track position estimate converges to within ± 5 feet in less than one second.

The along–track velocity error residual is shown in Figure 5.7 and the cross–track velocity error residual is shown in Figure 5.8. It may be observed from the

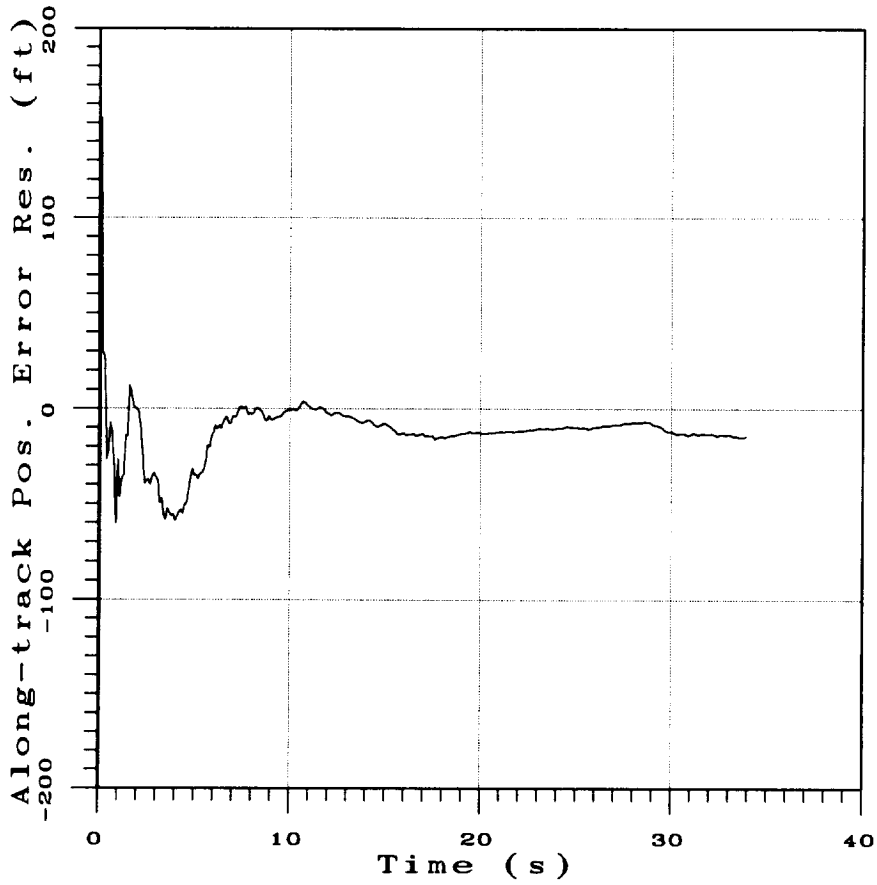

Figure 5.5: Along-track position error using Algorithm III.

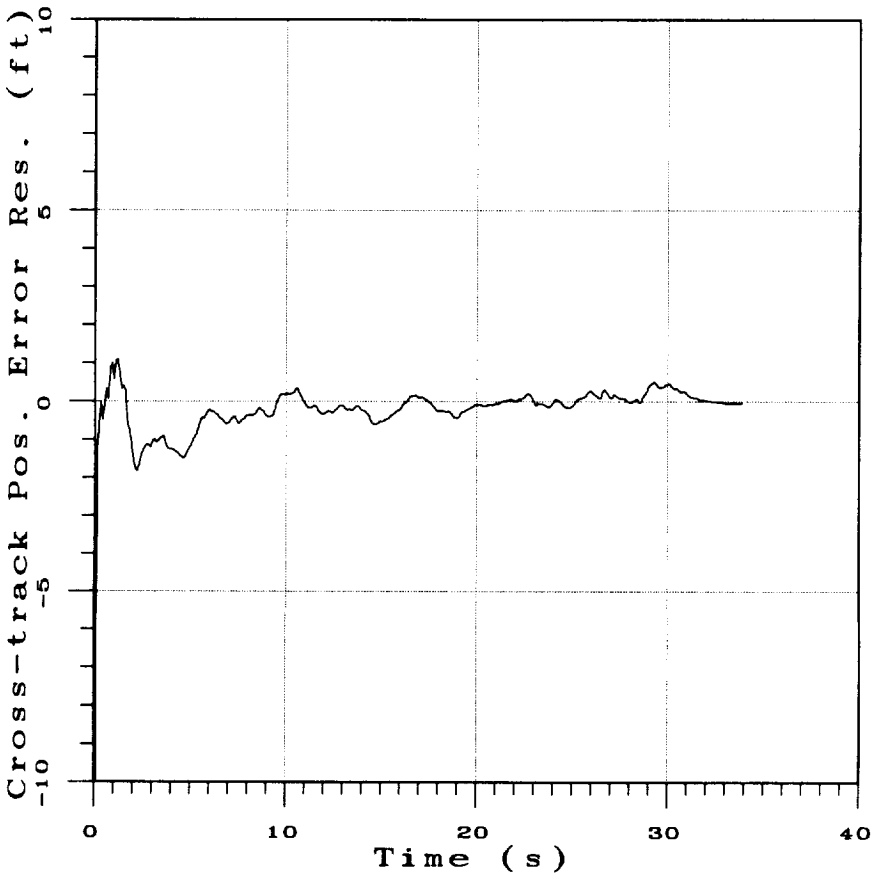

Figure 5.6: Cross–track position error using Algorithm III.

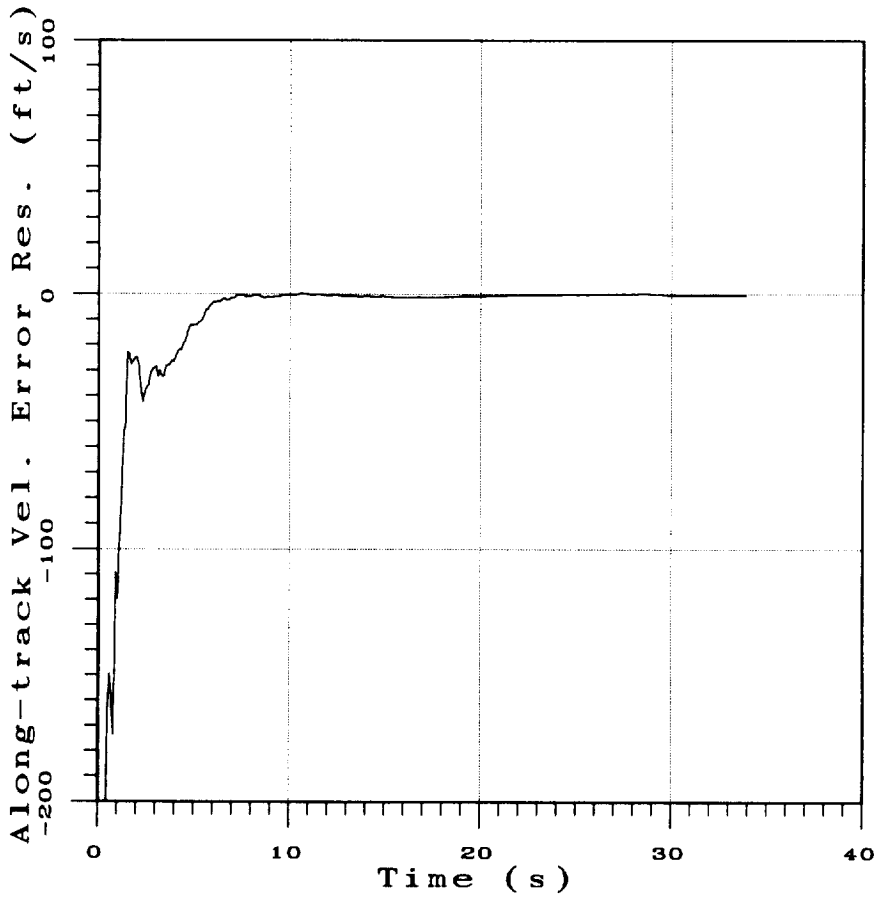

Figure 5.7: Along–track velocity error using Algorithm III.

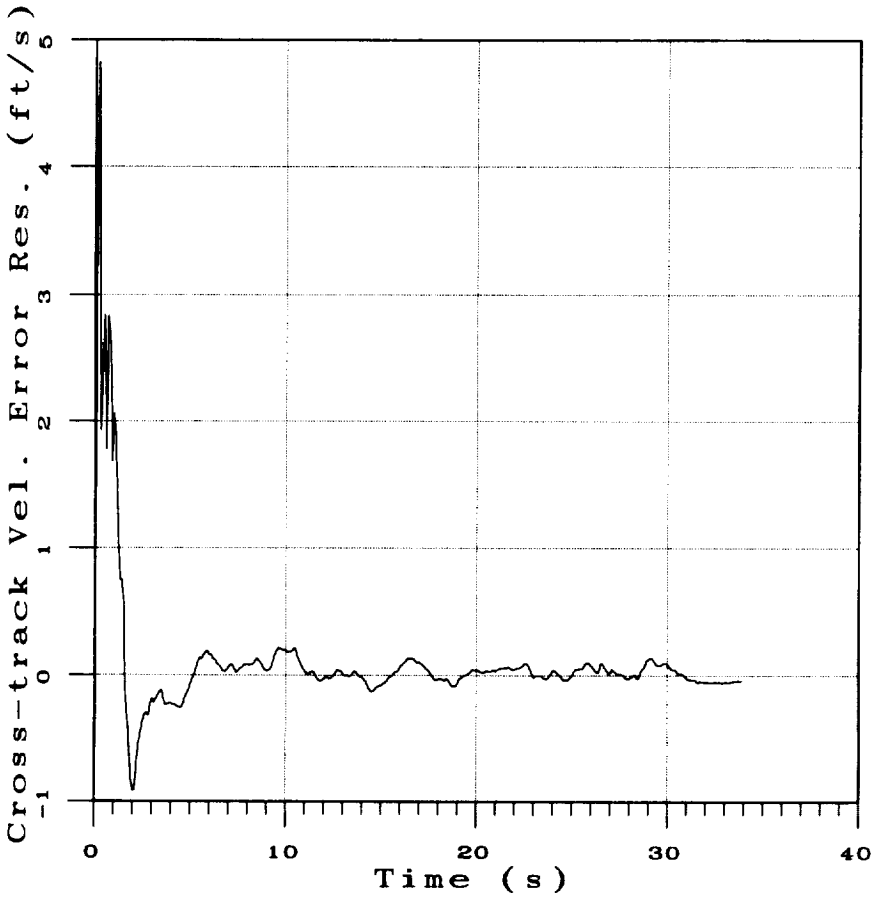

Figure 5.8: Cross-track velocity error using Algorithm III.

figures that the along–track velocity estimate settles to within ±10 feet/second in less than six seconds and the cross–track velocity settles to within ±5 feet/second in less than one second.

The position and velocity error residuals are summarized in Table 5.2. In this table the lateral component is in the units of feet. Comparing Table 5.2

Table 5.2: Algorithm III Results

Category	Lateral (y_b)
CAT I	±0.61
CAT II	±0.45
CAT IIIa	±0.51
CAT IIIb & c	±0.35

to Table 3.2, it may be seen that Algorithm III along with the Kalman filter generates position estimates which meet the navigation accuracy requirements for all Categories. It may be seen by comparing Tables 4.2, 4.3 and 5.2 that Algorithm III results are of comparable accuracy as Algorithm I with $w_1 = 1$ and $w_2 = 0$. The results are better for Category III.

5.2 Algorithm IV

This algorithm is motivated by the desire to estimate the three position components of the aircraft relative to the inertial coordinate system located on the runway. In Chapter 4 the three position components were estimated using Algorithm II which was based on an iterative scheme. The attempt here is to use the structure discovered in Algorithm III to compute all three components of the aircraft position without the use of an iterative scheme.

As discussed earlier, if any one of the aircraft inertial position components is known, the remaining position components can be recovered using Equations (5.8) and (5.9). However, in order to solve for three position components, at least one

additional equation relating the image quantities to the model quantities in terms of the aircraft position components is required.

Several such equations can be formed by using the two basic Equations (5.7) and (5.8) on groups of lights that represent well defined structures in the ordered graphs. For example, an additional equation can be constructed from Equation (5.8) as follows. Scaled indices between zero and 15 of the ordered y-coordinates of the model lights and the observed lights can be used in Equation (5.8) to yield:

$$\frac{1}{N_2}\sum_{1\le i\le N_2} y_i = y_b + \frac{h}{M_2}\sum_{1\le p\le M_2}\frac{(s_{1p}s_{6p} - s_{3p}s_{4p})}{(s_{2p}s_{4p} - s_{1p}s_{5p})} \tag{5.10}$$

Here, N_2 and M_2 are the number of lights that represent the left edge lights in the model lighting layout and in the observed lighting layout.

A similar equation can also be written for the right edge lights by using the scaled indices between 85 and 100 of the ordered sets as:

$$\frac{1}{(N - N_3 + 1)}\sum_{N_3\le i\le N} y_i = y_b + \frac{h}{M - M_3 + 1}\sum_{M_3\le p\le M}\frac{(s_{1p}s_{6p} - s_{3p}s_{4p})}{(s_{2p}s_{4p} - s_{1p}s_{5p})} \tag{5.11}$$

where, N_3 and M_3 are the number of scaled indices between zero and 85 of the ordered sets.

Re-writing Equations (5.9), (5.8), (5.10) and (5.11) as:

$$x_b + a_1 h = b_1 \tag{5.12}$$

$$y_b + a_2 h = b_2 \tag{5.13}$$

$$y_b + a_3 h = b_3 \tag{5.14}$$

$$y_b + a_4 h = b_4 \tag{5.15}$$

it may be seen that Equation (5.12) and any one of the remaining three equations form a linearly independent set of two equations. Note, a_1 to a_4 are the image based averages and b_1 to b_4 are the model based averages. Equations (5.14) and (5.15) are linearly dependent if $a_3 = a_4$. This can only happen if all the lights being considered lie along a single projection ray from the lens center. Such a situation cannot arise when the camera is above the plane of the runway as discussed in Section

2.3.7.1. Therefore, Equations (5.14) and (5.15) are always linearly independent for the viewing geometry considered here. Equation (5.13) can be obtained by a linear combination of Equations (5.14) and (5.15) with a perfect camera. This is because the left and right edge lights are symmetrically located about the centerline lights. Errors in the imaging process can cause Equation (5.13) also to be linearly independent.

Since Equations (5.14) and (5.15) are linearly independent, the aircraft altitude can be estimated by subtracting Equation (5.11) from Equation (5.10). Thus,

$$h = \frac{\frac{1}{N_2}\sum_{1\le i\le N_2} y_i - \frac{1}{(N-N_3+1)}\sum_{N_3\le i\le N} y_i}{\frac{1}{M_2}\sum_{1\le p\le M_2}\frac{(s_{1p}s_{6p}-s_{3p}s_{4p})}{(s_{2p}s_{4p}-s_{1p}s_{5p})} - \frac{1}{M-M_3+1}\sum_{M_3\le p\le M}\frac{(s_{1p}s_{6p}-s_{3p}s_{4p})}{(s_{2p}s_{4p}-s_{1p}s_{5p})}} \tag{5.16}$$

With the altitude so determined, Algorithm III can be used for determining the x_b and y_b components of the aircraft inertial position.

Alternatively, Equations (5.9), (5.8), (5.10) and (5.11) can be used for obtaining a least squares solution for the aircraft inertial position components. Following the notation of Equations (5.12) through (5.15), the resulting Least Squares solution [55] is given by:

$$[x_b, y_b, h]^T = (A^T A)^{-1} A^T [b_1, b_2, b_3, b_4]^T \tag{5.17}$$

where,

$$A = \begin{bmatrix} 1 & 0 & a_1 \\ 0 & 1 & a_2 \\ 0 & 1 & a_3 \\ 0 & 1 & a_4 \end{bmatrix} \tag{5.18}$$

It is important to point out that the proposed method estimates all the three position components without resorting to any iterative calculations. Thus, this approach can be expected to be faster and more robust than the Algorithm II described in Chapter 4. It may be noted that as in Algorithm III, the six–state Kalman filter driven by the outputs generated by Algorithm IV can be used for estimating the aircraft position and velocity components.

Algorithm IV is summarized in Table 5.3.

Table 5.3: Summary of Algorithm IV

1. Arrange x_i and y_i coordinates of all the N lights from the airport lighting geometry model in non-decreasing order.

2. Compute the scaled indices $k_i = 100i/N$; $1 \leq i \leq N$.

3. Select the ordered x_i $\forall$ i such that $0 \leq k_i \leq 40$. Compute the mean b_1 for Equation (5.12) using the N_1 selected x_i in Equation (5.9).

4. Compute b_2 for Equation (5.13) using all y_i, $1 \leq i \leq N$, in Equation (5.8). Compute b_3 and b_4 for Equations (5.14) and (5.15) by selecting the ordered y_i such that $0 \leq k_i \leq 15$ and $85 \leq k_i \leq 100$ respectively and using them in Equations (5.10) and (5.11).

5. Compute s_{1_p} through s_{6_p} using steps 5, 6 and 7 of Algorithm III in Table 5.1.

6. Compute $\tilde{x}_p = (s_{2_p}s_{6_p} - s_{3_p}s_{5_p})/(s_{1_p}s_{5_p} - s_{4_p}s_{2_p})$
and $\tilde{y}_p = (s_{1_p}s_{6_p} - s_{3_p}s_{4_p})/(s_{2_p}s_{4_p} - s_{1_p}s_{5_p})$ $\forall p$; $1 \leq p \leq M$.

7. Arrange $\tilde{x}_p$ and $\tilde{y}_p$ coordinates of M image lights in non-decreasing order.

8. Compute the scaled indices $k_p = 100p/M$; $1 \leq p \leq M$.

9. Select the ordered $\tilde{x}_p$ $\forall$ p such that $0 \leq k_p \leq 40$. Compute the mean a_1 for Equation (5.12) using the M_1 selected $\tilde{x}_p$ in Equation (5.9).

10. Compute a_2 for Equation (5.13) using all $\tilde{y}_p$, $1 \leq i \leq M$, in Equation (5.8). Compute a_3 and a_4 for Equations (5.14) and (5.15) by selecting the ordered $\tilde{y}_p$ such that $0 \leq k_p \leq 15$ and $85 \leq k_p \leq 100$ respectively and using them in Equations (5.10) and (5.11).

11. Use Equation (5.17) to compute the aircraft position components x_b, y_b and h.

5.2.1 Results Using Algorithm IV

The position and velocity estimates generated by the six–state Kalman filter driven by outputs of Algorithm IV are described in this section. The simulation scenario and the initial conditions were same as in Algorithm II in Chapter 4.

The along–track position error residual is shown in Figure 5.9. It may be seen

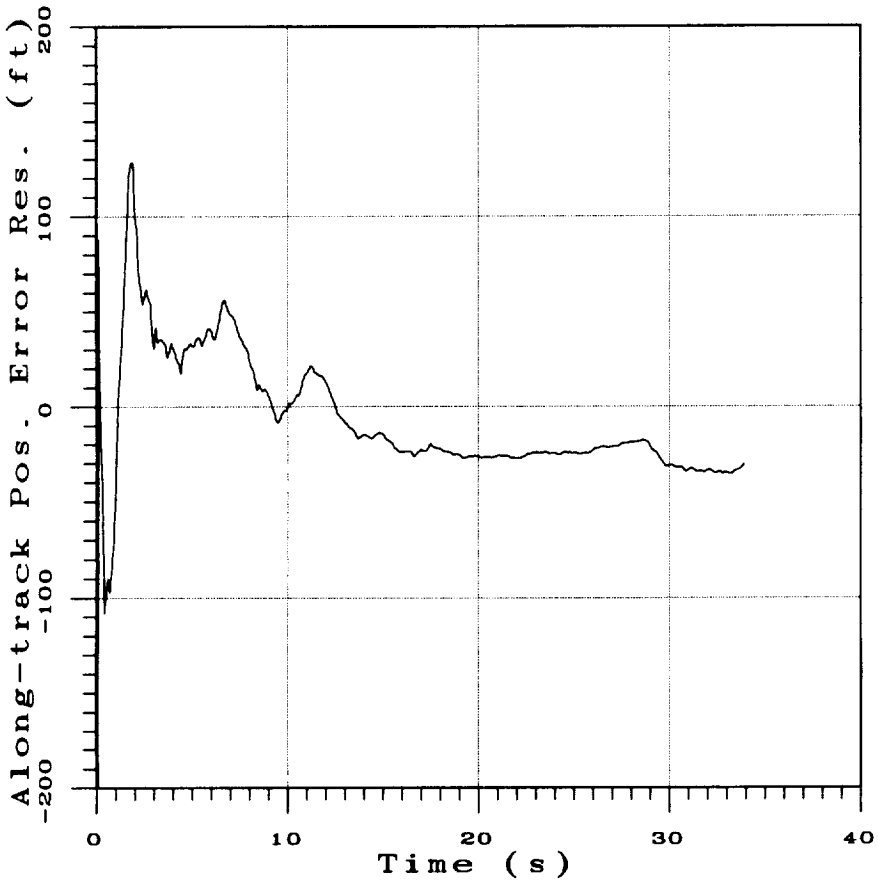

Figure 5.9: Along–track position error using Algorithm IV.

from the figure that the along–track position estimate converges to within ± 100 feet in less than two seconds.

The cross–track position error residual given in Figure 5.10 shows that the cross–track position estimate converges to within ± 5 feet in less than one second.

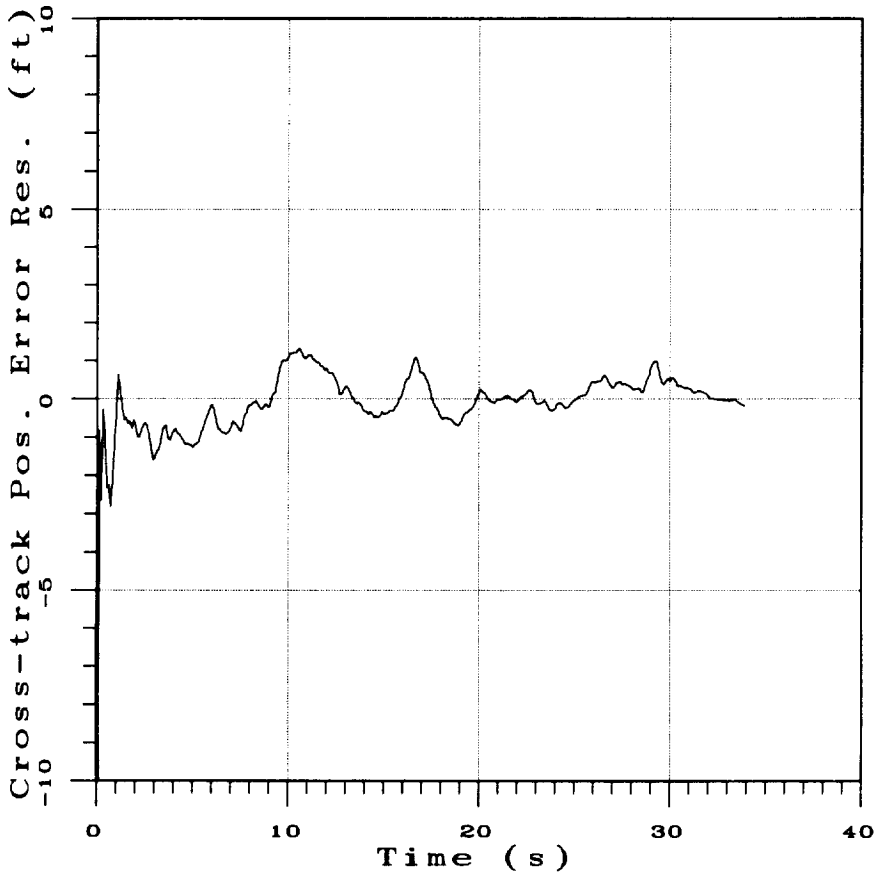

Figure 5.10: Cross-track position error using Algorithm IV.

Figure 5.11 illustrates the altitude error residual. It may be seen that it converges to within ±5 feet in less than six seconds.

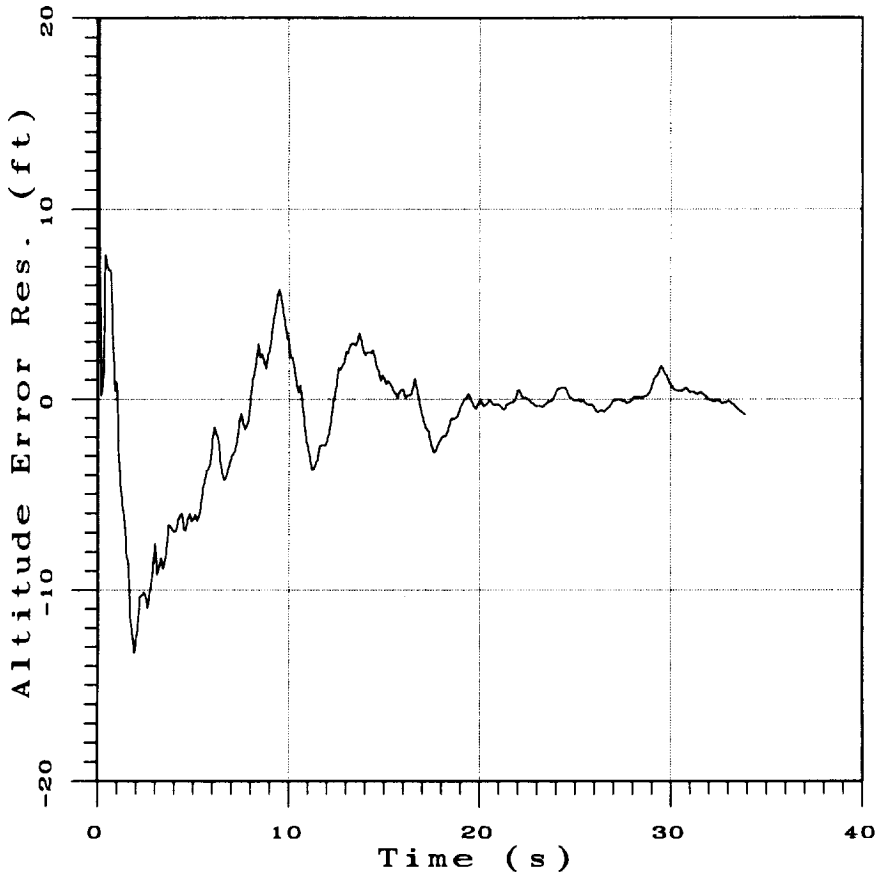

Figure 5.11: Altitude error using Algorithm IV.

The along–track and cross–track velocity error residuals are shown in Figures 5.12 and 5.13. These figures show that the along–track velocity estimate settles to within ±10 feet/second in less than three seconds and the cross–track velocity settles to within ±5 feet/second in less than a second.

The sink rate error residual is shown in Figure 5.14. It may be observed that the sink rate error is reduced to within ±5 feet/second in less than three seconds.

The position error residuals are summarized in Table 5.4. The notation and

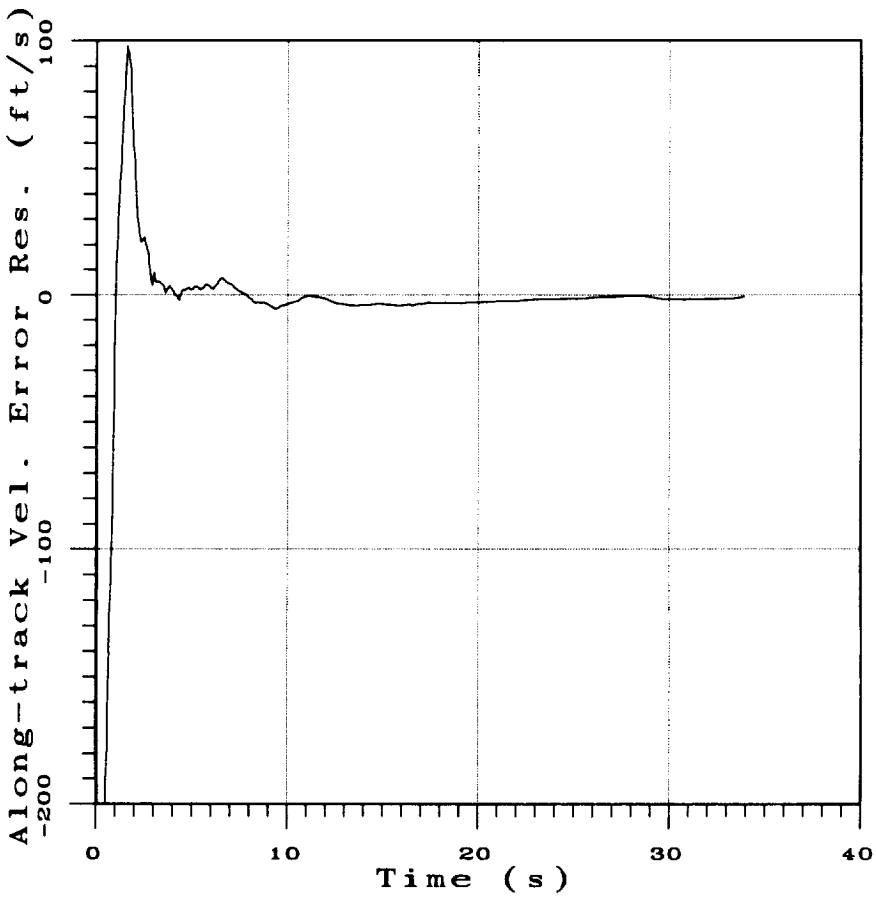

Figure 5.12: Along–track velocity error using Algorithm IV.

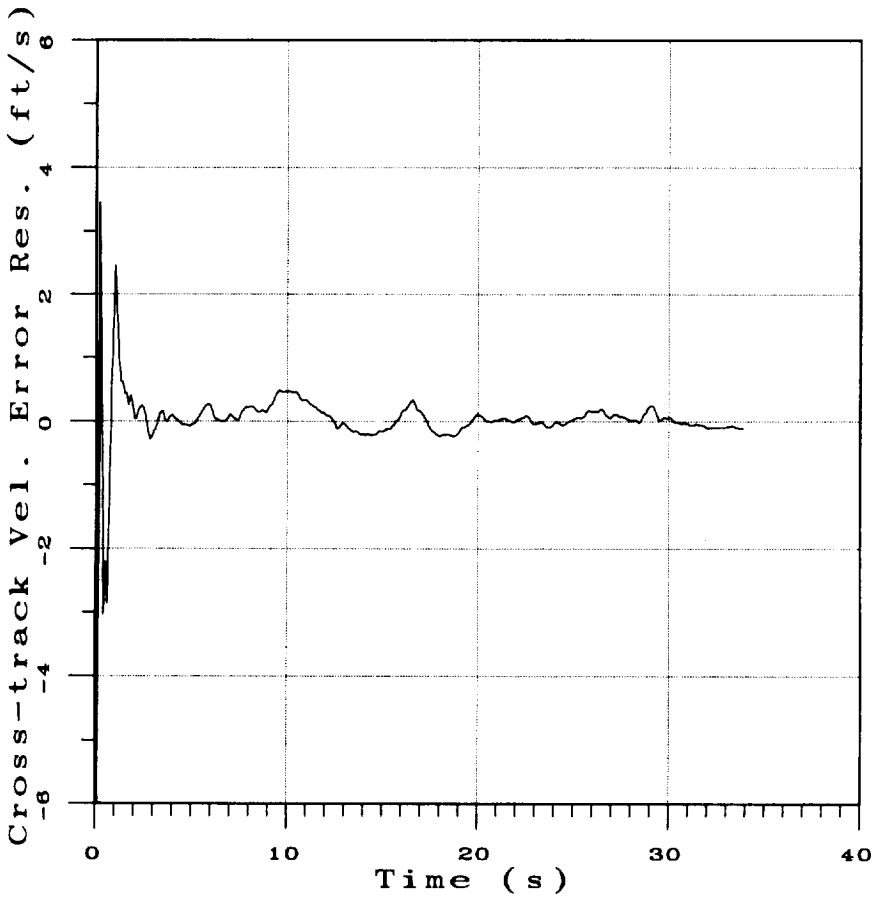

Figure 5.13: Cross-track velocity error using Algorithm IV.

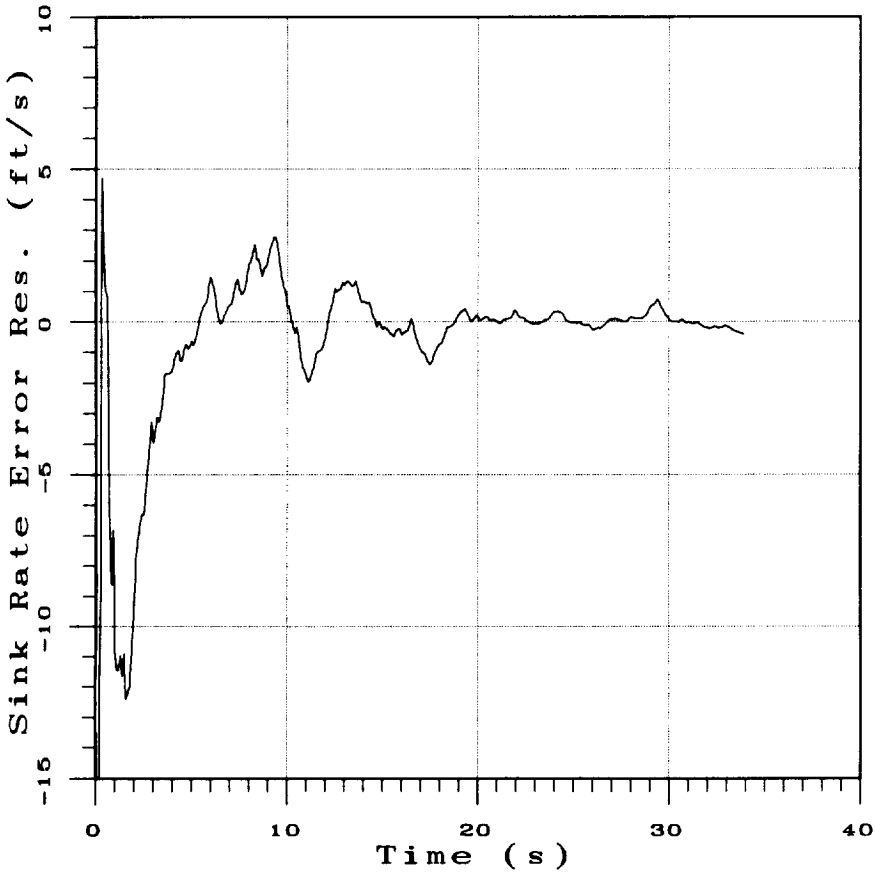

Figure 5.14: Sink rate error using Algorithm IV.

units are same as those in Table 5.2. Comparing Table 5.4 to Table 3.2, it may be

Table 5.4: Algorithm IV Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±1.32	±5.77
CAT II	±0.69	±2.82
CAT IIIa	±0.99	±1.74
CAT IIIb & c	±0.34	±0.77

noted that Algorithm IV along with the Kalman filter generates position estimates which meet the navigation accuracy requirements for all Categories. Moreover, Algorithm IV results are better than Algorithm II results with $w_1 = 1$ and $w_2 = 0$ at lower altitudes and they are far superior to the results obtained using Algorithm I with $w_1 = 1$ and $w_2 = 1$ at all altitudes.

5.3 Summary

The main theme of this chapter was to exploit the structures in the model and in the observed lighting layouts to enable direct estimation of the position components. The structures formed by the threshold bar lights, left and right edge lights and the centerline lights were discovered by arranging the lights in a non–decreasing order. Two methods which utilize the correspondence between the structures in the model and in the observed lighting layouts were described. The first method required an onboard altimeter and the yaw, pitch and roll angles. The second method only required knowledge of the aircraft attitude angles. It was shown that the second algorithm is able to estimate all three position components without any iterative computations. A six–state Kalman filter was used for integrating the information derived from these methods to improve the aircraft position estimates and to estimate velocity components along the descent path.

Chapter 6

Kalman Filter Integrated Methods

The algorithms described in this chapter are based on the second solution family described in Chapter 3. Note that the position determination algorithms in the two previous chapters were based on the first solution family. The reason for using the second solution family is that they are suitable for use as the basis for developing predictor/corrector methods. The techniques are based on matching the features between camera image and model based image. Since the model based image is synthesized by using the model of the airport lighting, camera model and the estimated aircraft position and orientation, the difference between the model image based and camera image based measurements can be used in a feedback scheme for driving the differences to zero, thereby recovering the position and orientation of the aircraft. This concept, coupled with the fact that the aircraft equations of motion temporally relate the aircraft states, allows the state estimation problem to be cast as a Kalman filtering problem. A summary of the Kalman filtering algorithm is given in Appendix A of this report.

In the previous two chapters, a six–state Kalman filter was used for integrating the aircraft motion with the position estimates generated by the algorithms. The position estimates were used as inputs to the Kalman filter. In this sense, the

integration of the aircraft dynamics with the position determination algorithms was external. The Kalman filters used in this chapter attempt a much deeper level of integration between the vehicle dynamics and the observed image sequence. Unlike the algorithms in the previous chapters, correspondence between airport lighting layout model and the observed images is achieved by minimizing the difference between the camera image and the model image using the Kalman filter.

The Kalman filters employed in the present work have their basis in the research reported in Reference [90]. In that work, three different Kalman filters were computed for the position determination problem. Kalman filters in the sensor coordinate system and in the inertial coordinate system were set up and compared for an image based ranging problem. It was shown that the Kalman filters in the sensor coordinate system and in the inertial coordinate system had comparable accuracies. However, the formulation in the inertial coordinate system was much easier to implement. An additional advantage is that since the translational states are linear in the inertial frame, the process update part of the Kalman filter is linear when formulated in this coordinate system.

The ensuing sections describe three Kalman filtering algorithms. The first and second algorithms described in this chapter, Algorithm V and Algorithm VI, assume that the orientation angles ψ, θ and ϕ are known. These algorithms are designed to estimate the three runway relative position components. Algorithm V uses information from the camera image and the airport lighting model for position determination.

Algorithm VI fuses the image and airport lighting model information with aircraft position estimates obtained from the Global Positioning System (GPS). GPS is an emerging technology for navigation. It is a satellite based navigation system that determines aircraft position. There is a strong movement within the aeronautical community to incorporate GPS receivers in every aircraft. GPS based position estimates can be integrated with the image based position determination algorithms to improve the accuracy and robustness of position estimates. Such

an integration can also help develop a navigation instrument that synergistically exploits all the information sources in an airport environment. This forms the motivation for the development of Algorithm VI. Algorithm VII is designed to provide both runway relative position and orientation. Unlike Algorithms V and VI, this algorithms only assumes the availability of the roll orientation angle. Earlier versions of algorithms VI and VII have been reported in [16] and [17].

Since several measurement equations and computations are common to all the algorithms, they are discussed first in this chapter. The raw data derived from the image consists of the image coordinates of every light visible in the image. Thus, the coordinates of a light p, $[\tilde{u}_p, \tilde{v}_p]^T$, are available in the raw data. The raw measurements have a position uncertainty due to the errors introduced during the imaging process. These errors can be modeled as:

$$\tilde{u}_p = u_p + \eta_u \tag{6.1}$$

$$\tilde{v}_p = v_p + \eta_v \tag{6.2}$$

where, η_u and η_v represent pixel position uncertainty. For modeling purposes, η_u and η_v can be assumed to be independent scalar white noise processes.

Clearly, the coordinates of individual lights contain little or no information regarding the shape or size of lighting layout. In order to incorporate this information, secondary measurements obtained by combining the coordinates of some or all the lights are needed. One of the ways of generating information regarding the shape and size is to assume that the image coordinates are random variables and construct aggregation formulae that characterize the shape and size. With this notion, size and shape can be related to the distribution of the two random variables. In that case, the characteristics of the distribution such as mean, variance and higher–order and moments can be used to establish the size and shape of the object in the image. Additional measures such as the correlation coefficient and the eigen values of the covariance matrix or matrix singular values can also be used.

Extensive numerical experiments revealed that, the following six image–based measurements are found to be useful for characterizing the shape and size of the airport lighting layout:

$$z_1 = \sum_{1 \le p \le M} \tilde{u}_p / M \tag{6.3}$$

$$z_2 = \sum_{1 \le p \le M} \tilde{v}_p / M \tag{6.4}$$

$$z_3 = \sum_{1 \le p \le M} (\sqrt{\tilde{u}_p^2 + \tilde{v}_p^2}) / M \tag{6.5}$$

$$z_4 = \sqrt{\sum_{1 \le p \le M} (\tilde{v}_p - z_2)^2 / M} \tag{6.6}$$

$$z_5 = \sqrt{\sum_{1 \le p \le M} (\tilde{u}_p - z_1 + \tilde{v}_p - z_2)^2 / (2M)} \tag{6.7}$$

$$z_6 = \sqrt{\sum_{1 \le p \le M} (\tilde{u}_p - z_1)^2 / M} \tag{6.8}$$

In these equations, M is the number of lights detected in the image; $\tilde{u}_p$ and $\tilde{v}_p$ are the measured coordinates of the individual light sources. The last three measurements, Equations (6.6) through (6.8) may be compactly written as:

$$z_j = \sqrt{\sum_{1 \le p \le M} [(\tilde{u}_p - z_1) \sin \tau + (\tilde{v}_p - z_2) \cos \tau]^2 / M} \tag{6.9}$$

where, $j = 4, \ldots, 6$ correspond to $\tau = 0$, $\tau = 45$ and $\tau = 90$ degrees. The secondary measurements in Equations (6.3) through (6.8) aggregate the size and shape information about the airport lighting layout.

Physically, the first and second measurements are the arithmetic means of positions of the observed airport lights in the image plane. They can also be thought of as the coordinates of the centroid of the observed light distribution in the image plane. The third measurement gives the the mean distance of the observed lights from the origin of the image coordinate system. The fourth measurement is the moment about an axis parallel to the u image axis and passing through the centroid defined by the mean. Similarly, the fifth measurement is the moment about an axis passing through the cluster centroid and is inclined at 45 degrees to both the u and v image axes. Finally, the sixth measurement is the moment

about an axis which passes through the mean and is parallel to the v image axis. The fourth, fifth and sixth measurements are the square roots of the elements of the covariance matrix. Specifically, the fourth and sixth measurements are the standard deviations of the v_p and u_p coordinates. The fifth measurement is the square root of the weighted sum of the variances of u_p and v_p and their covariance.

It may be noted that the six measurements z_1 through z_6 in Equations (6.3) through (6.8) can only assume positive values. The first two measurements are always positive due to the choice of the origin of the image plane defined in Figure 3.6. The other four measurements are always positive due to the use of sums of squares. Note that the squaring operation tends to decrease the effects of small numbers, while amplifying the influence of large numbers. Use of square root operation in the secondary measurements z_3 through z_6 prevent the position estimation algorithms from being biased towards large errors.

As described in Appendix A, in order to utilize these secondary measurements using a Kalman filter, the measurements have to be related to the aircraft states as follows:

$$\mathbf{Z} = \mathbf{h} + \zeta_{\mathbf{z}} \tag{6.10}$$

where, $\mathbf{Z}$ is the 6×1 vector of secondary measurements z_1 through z_6, $\mathbf{h}$ is the 6×1 state dependent measurement model vector and $\zeta_{\mathbf{z}}$ is the measurement noise vector. The components of the $\mathbf{h}$ vector, h_1 through h_6, are:

$$h_1 = \sum_{1 \le p \le N} u_p/N \tag{6.11}$$

$$h_2 = \sum_{1 \le p \le N} v_p/N \tag{6.12}$$

$$h_3 = \sum_{1 \le p \le N} (\sqrt{u_p^2 + v_p^2})/N \tag{6.13}$$

$$h_4 = \sqrt{\sum_{1 \le p \le N} (v_p - h_2)^2/N} \tag{6.14}$$

$$h_5 = \sqrt{\sum_{1 \le p \le N} (u_p - h_1 + v_p - h_2)^2/(2N)} \tag{6.15}$$

$$h_6 = \sqrt{\sum_{1 \le p \le N} (u_p - h_1)^2/N} \tag{6.16}$$

where N is the number of model lights within the synthesized image and u_p and v_p are the image coordinates of the lights.

For computation of the Kalman gain and propagation of the state error covariance matrix, a linearized measurement matrix $\mathbf{H}(k)$, is required. The elements of the $\mathbf{H}(k)$ matrix can be obtained by evaluating the partial derivatives of the measurement model vector with respect to the estimated states. The partial derivatives used in the computation of the measurement matrix $\mathbf{H}(k)$ can be obtained using the perspective projection Equations (3.41) and (3.42) as:

$$\frac{\partial u_p}{\partial x_b} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial x_b}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial x_b}\right)}{x_{cp}^2} \tag{6.17}$$

$$\frac{\partial u_p}{\partial y_b} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial y_b}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial y_b}\right)}{x_{cp}^2} \tag{6.18}$$

$$\frac{\partial u_p}{\partial z_b} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial z_b}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial z_b}\right)}{x_{cp}^2} \tag{6.19}$$

$$\frac{\partial u_p}{\partial \psi} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial \psi}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial \psi}\right)}{x_{cp}^2} \tag{6.20}$$

$$\frac{\partial u_p}{\partial \theta} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial \theta}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial \theta}\right)}{x_{cp}^2} \tag{6.21}$$

$$\frac{\partial u_p}{\partial \phi} = f\frac{x_{cp}\left(\frac{\partial y_{cp}}{\partial \phi}\right) - y_{cp}\left(\frac{\partial x_{cp}}{\partial \phi}\right)}{x_{cp}^2} \tag{6.22}$$

$$\frac{\partial v_p}{\partial x_b} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial x_b}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial x_b}\right)}{x_{cp}^2} \tag{6.23}$$

$$\frac{\partial v_p}{\partial y_b} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial y_b}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial y_b}\right)}{x_{cp}^2} \tag{6.24}$$

$$\frac{\partial v_p}{\partial z_b} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial z_b}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial z_b}\right)}{x_{cp}^2} \tag{6.25}$$

$$\frac{\partial v_p}{\partial \psi} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial \psi}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial \psi}\right)}{x_{cp}^2} \tag{6.26}$$

$$\frac{\partial v_p}{\partial \theta} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial \theta}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial \theta}\right)}{x_{cp}^2} \tag{6.27}$$

$$\frac{\partial v_p}{\partial \phi} = f\frac{x_{cp}\left(\frac{\partial z_{cp}}{\partial \phi}\right) - z_{cp}\left(\frac{\partial x_{cp}}{\partial \phi}\right)}{x_{cp}^2} \tag{6.28}$$

where the six position and orientation states are x_b, y_b, z_b, ψ, θ and ϕ. f is the focal length of the camera. The partial derivatives of x_{cp}, y_{cp} and z_{cp} with respect to the position states are obtained from Equations (3.7) through (3.9) as:

$$\frac{\partial x_{cp}}{\partial x_b} = -r_1 \tag{6.29}$$
$$\frac{\partial x_{cp}}{\partial y_b} = -r_2 \tag{6.30}$$
$$\frac{\partial x_{cp}}{\partial z_b} = -r_3 \tag{6.31}$$
$$\frac{\partial y_{cp}}{\partial x_b} = -r_4 \tag{6.32}$$
$$\frac{\partial y_{cp}}{\partial y_b} = -r_5 \tag{6.33}$$
$$\frac{\partial y_{cp}}{\partial z_b} = -r_6 \tag{6.34}$$
$$\frac{\partial z_{cp}}{\partial x_b} = -r_7 \tag{6.35}$$
$$\frac{\partial z_{cp}}{\partial y_b} = -r_8 \tag{6.36}$$
$$\frac{\partial z_{cp}}{\partial z_b} = -r_9 \tag{6.37}$$

Similarly, the partial derivatives with respect to the orientation states are also obtained from Equations (3.7) through (3.9) as:

$$\frac{\partial x_{cp}}{\partial \psi} = (x_p - x_b)\frac{\partial r_1}{\partial \psi} + (y_p - y_b)\frac{\partial r_2}{\partial \psi} + (z_p - z_b)\frac{\partial r_3}{\partial \psi} \tag{6.38}$$
$$\frac{\partial x_{cp}}{\partial \theta} = (x_p - x_b)\frac{\partial r_1}{\partial \theta} + (y_p - y_b)\frac{\partial r_2}{\partial \theta} + (z_p - z_b)\frac{\partial r_3}{\partial \theta} \tag{6.39}$$
$$\frac{\partial x_{cp}}{\partial \phi} = (x_p - x_b)\frac{\partial r_1}{\partial \phi} + (y_p - y_b)\frac{\partial r_2}{\partial \phi} + (z_p - z_b)\frac{\partial r_3}{\partial \phi} \tag{6.40}$$
$$\frac{\partial y_{cp}}{\partial \psi} = (x_p - x_b)\frac{\partial r_4}{\partial \psi} + (y_p - y_b)\frac{\partial r_5}{\partial \psi} + (z_p - z_b)\frac{\partial r_6}{\partial \psi} \tag{6.41}$$
$$\frac{\partial y_{cp}}{\partial \theta} = (x_p - x_b)\frac{\partial r_4}{\partial \theta} + (y_p - y_b)\frac{\partial r_5}{\partial \theta} + (z_p - z_b)\frac{\partial r_6}{\partial \theta} \tag{6.42}$$
$$\frac{\partial y_{cp}}{\partial \phi} = (x_p - x_b)\frac{\partial r_4}{\partial \phi} + (y_p - y_b)\frac{\partial r_5}{\partial \phi} + (z_p - z_b)\frac{\partial r_6}{\partial \phi} \tag{6.43}$$

$$\frac{\partial z_{cp}}{\partial \psi} = (x_p - x_b)\frac{\partial r_7}{\partial \psi} + (y_p - y_b)\frac{\partial r_8}{\partial \psi} + (z_p - z_b)\frac{\partial r_9}{\partial \psi} \quad (6.44)$$

$$\frac{\partial z_{cp}}{\partial \theta} = (x_p - x_b)\frac{\partial r_7}{\partial \theta} + (y_p - y_b)\frac{\partial r_8}{\partial \theta} + (z_p - z_b)\frac{\partial r_9}{\partial \theta} \quad (6.45)$$

$$\frac{\partial z_{cp}}{\partial \phi} = (x_p - x_b)\frac{\partial r_7}{\partial \phi} + (y_p - y_b)\frac{\partial r_8}{\partial \phi} + (z_p - z_b)\frac{\partial r_9}{\partial \phi} \quad (6.46)$$

Here, r_1 through r_9 are the components of the transformation matrix from inertial frame to camera frame given by Equation (3.3). It may be noted that the partial derivatives in Equations (6.29) through (6.46) can be computed for each light **p** in the image, synthesized from the model of the airport lighting using the estimated position and orientation states.

The partial derivatives of h_1, h_2 and h_3 measurements can be computed using Equations (6.17) through (6.28) and Equations (6.11) through (6.13) as follows:

$$\frac{\partial h_1}{\partial x_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial x_b} \quad (6.47)$$

$$\frac{\partial h_1}{\partial y_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial y_b} \quad (6.48)$$

$$\frac{\partial h_1}{\partial z_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial z_b} \quad (6.49)$$

$$\frac{\partial h_1}{\partial \psi} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial \psi} \quad (6.50)$$

$$\frac{\partial h_1}{\partial \theta} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial \theta} \quad (6.51)$$

$$\frac{\partial h_1}{\partial \phi} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial u_p}{\partial \phi} \quad (6.52)$$

$$\frac{\partial h_2}{\partial x_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial v_p}{\partial x_b} \quad (6.53)$$

$$\frac{\partial h_2}{\partial y_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial v_p}{\partial y_b} \quad (6.54)$$

$$\frac{\partial h_2}{\partial z_b} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial v_p}{\partial z_b} \quad (6.55)$$

$$\frac{\partial h_2}{\partial \psi} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial v_p}{\partial \psi} \quad (6.56)$$

$$\frac{\partial h_2}{\partial \theta} = \frac{1}{N}\sum_{1\le p\le N}\frac{\partial v_p}{\partial \theta} \quad (6.57)$$

$$\frac{\partial h_2}{\partial \phi} = \frac{1}{N} \sum_{1 \le p \le N} \frac{\partial v_p}{\partial \phi} \tag{6.58}$$

$$\frac{\partial h_3}{\partial x_b} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial x_b} + v_p \frac{\partial v_p}{\partial x_b}}{\sqrt{u_p^2 + v_p^2}} \tag{6.59}$$

$$\frac{\partial h_3}{\partial y_b} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial y_b} + v_p \frac{\partial v_p}{\partial y_b}}{\sqrt{u_p^2 + v_p^2}} \tag{6.60}$$

$$\frac{\partial h_3}{\partial z_b} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial z_b} + v_p \frac{\partial v_p}{\partial z_b}}{\sqrt{u_p^2 + v_p^2}} \tag{6.61}$$

$$\frac{\partial h_3}{\partial \psi} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial \psi} + v_p \frac{\partial v_p}{\partial \psi}}{\sqrt{u_p^2 + v_p^2}} \tag{6.62}$$

$$\frac{\partial h_3}{\partial \theta} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial \theta} + v_p \frac{\partial v_p}{\partial \theta}}{\sqrt{u_p^2 + v_p^2}} \tag{6.63}$$

$$\frac{\partial h_3}{\partial \phi} = \frac{1}{N} \sum_{1 \le p \le N} \frac{u_p \frac{\partial u_p}{\partial \phi} + v_p \frac{\partial v_p}{\partial \phi}}{\sqrt{u_p^2 + v_p^2}} \tag{6.64}$$

where, N is the number of lights. Similarly, the partial derivatives of h_4, h_5 and h_6 can be evaluated as:

$$\frac{\partial h_j}{\partial x_b} = \frac{\sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial x_b} - \frac{\partial h_1}{\partial x_b})\sin\tau_j + (\frac{\partial v_p}{\partial x_b} - \frac{\partial h_2}{\partial x_b})\cos\tau_j]}{\sqrt{N \sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.65}$$

$$\frac{\partial h_j}{\partial y_b} = \frac{\sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial y_b} - \frac{\partial h_1}{\partial y_b})\sin\tau_j + (\frac{\partial v_p}{\partial y_b} - \frac{\partial h_2}{\partial y_b})\cos\tau_j]}{\sqrt{N \sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.66}$$

$$\frac{\partial h_j}{\partial z_b} = \frac{\sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial z_b} - \frac{\partial h_1}{\partial z_b})\sin\tau_j + (\frac{\partial v_p}{\partial z_b} - \frac{\partial h_2}{\partial z_b})\cos\tau_j]}{\sqrt{N \sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.67}$$

$$\frac{\partial h_j}{\partial \psi} = \frac{\sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial \psi} - \frac{\partial h_1}{\partial \psi})\sin\tau_j + (\frac{\partial v_p}{\partial \psi} - \frac{\partial h_2}{\partial \psi})\cos\tau_j]}{\sqrt{N \sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.68}$$

$$\frac{\partial h_j}{\partial \theta} = \frac{\sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial \theta} - \frac{\partial h_1}{\partial \theta})\sin\tau_j + (\frac{\partial v_p}{\partial \theta} - \frac{\partial h_2}{\partial \theta})\cos\tau_j]}{\sqrt{N \sum_{1 \le p \le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.69}$$

$$\frac{\partial h_j}{\partial \phi} = \frac{\sum_{1\le p\le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j][(\frac{\partial u_p}{\partial \phi} - \frac{\partial h_1}{\partial \phi})\sin\tau_j + (\frac{\partial v_p}{\partial \phi} - \frac{\partial h_2}{\partial \phi})\cos\tau_j]}{\sqrt{N\sum_{1\le p\le N}[(u_p - h_1)\sin\tau_j + (v_p - h_2)\cos\tau_j]^2}} \tag{6.70}$$

with $\tau_j = 0$, 45 and 90 degrees for $j = 4$, 5 and 6 respectively.

The complete observation matrix $\mathbf{H}(k)$ can now be written in terms of the partial derivatives of h_1 through h_6. The details of the $\mathbf{H}(k)$ matrix for each of the three algorithms will be described in the ensuing sections.

In addition to the measurement model, a linearized discrete time dynamic model is required for implementing the Kalman filter described in Appendix A. All the three algorithms described in this chapter use a discrete time dynamic equation of the form:

$$\mathbf{X}(k+1) = \mathbf{\Phi}(k)\mathbf{X}(k) \tag{6.71}$$

$\mathbf{X}(k)$ is the current state vector and $\mathbf{X}(k+1)$ is the state vector at the next sample instant. $\mathbf{\Phi}(k)$ is the state transition matrix. Since different state vectors are employed in each of the three algorithms, the state transition matrices are given separately in the following sections.

6.1 Algorithm V

This algorithm assumes that the aircraft yaw, pitch and roll orientations ψ, θ and ϕ are known. This algorithm estimates the three aircraft position components x_b, y_b, z_b, and the three aircraft velocity components v_{bx}, v_{by} and v_{bz}. The 6×6 state transition matrix is obtained by assuming that the aircraft velocity vector components remain constant, and that the velocity to position integration can be adequately approximated by the Euler integration [35] method. Thus, the state

transition matrix is of the form:

$$\mathbf{\Phi}(k) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.72}$$

Δt is the update time step.

The observation matrix is:

$$\mathbf{H}(k) = \begin{bmatrix} \frac{\partial h_1}{\partial x_b} & \frac{\partial h_1}{\partial y_b} & \frac{\partial h_1}{\partial z_b} & 0 & 0 & 0 \\ \frac{\partial h_2}{\partial x_b} & \frac{\partial h_2}{\partial y_b} & \frac{\partial h_2}{\partial z_b} & 0 & 0 & 0 \\ \frac{\partial h_3}{\partial x_b} & \frac{\partial h_3}{\partial y_b} & \frac{\partial h_3}{\partial z_b} & 0 & 0 & 0 \\ \frac{\partial h_4}{\partial x_b} & \frac{\partial h_4}{\partial y_b} & \frac{\partial h_4}{\partial z_b} & 0 & 0 & 0 \\ \frac{\partial h_5}{\partial x_b} & \frac{\partial h_5}{\partial y_b} & \frac{\partial h_5}{\partial z_b} & 0 & 0 & 0 \\ \frac{\partial h_6}{\partial x_b} & \frac{\partial h_6}{\partial y_b} & \frac{\partial h_6}{\partial z_b} & 0 & 0 & 0 \end{bmatrix} \tag{6.73}$$

In addition to the state transition matrix and the observation matrix, several other matrices are required for implementing the Kalman filter as described in Appendix A. These are defined below.

The control input vector is a 6×1 null vector. For convenience, the input distribution matrix $\mathbf{\Gamma}(k)$ and the disturbance distribution matrix $\mathbf{\Gamma}_{\mathbf{d}}(k)$ are chosen to be 6×6 identity matrices. The process noise covariance matrix $\mathbf{Q}(k)$ is a 6×6 null matrix. The dimension of the measurement noise covariance matrix $\mathbf{R}$ is 6×6. In the current implementation, the diagonal elements are set to the variance of $.25/\Delta t$ corresponding to the standard deviation of pixel position uncertainty of 0.5 pixels. Δt is the measurement update time step. Note that the variance of pixel position uncertainty has been divided by Δt to convert to the discrete time case.

In order to begin state estimation, the 6×1 state vector $\hat{\mathbf{X}}(k)$ and its error

covariance matrix $\mathbf{P}(k)$, of size 6×6, have to be initialized. The Kalman filtering algorithm described in Appendix A is then used for recursive state estimation.

6.1.1 Results Using Algorithm V

The position and velocity estimates generated by Algorithm V are described in this section. As before, the simulation scenario and the initial conditions were taken to be same as those used for the previous algorithms described in Chapters 4 and 5.

The along-track position error residual is shown in Figure 6.1. This figure

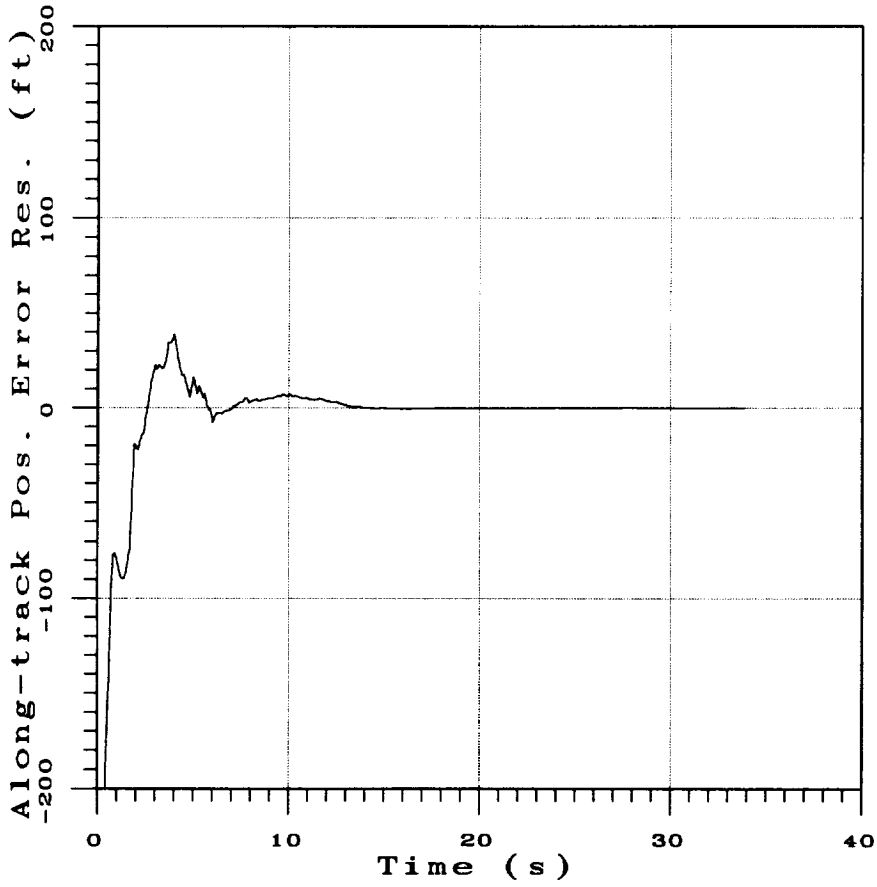

Figure 6.1: Along-track position error using Algorithm V.

shows that the along–track position estimates converge to within ±100 feet in less than two seconds.

The cross–track position error residual portrayed in Figure 6.2 shows that the cross–track position estimate converges to within ±5 feet in less than one second.

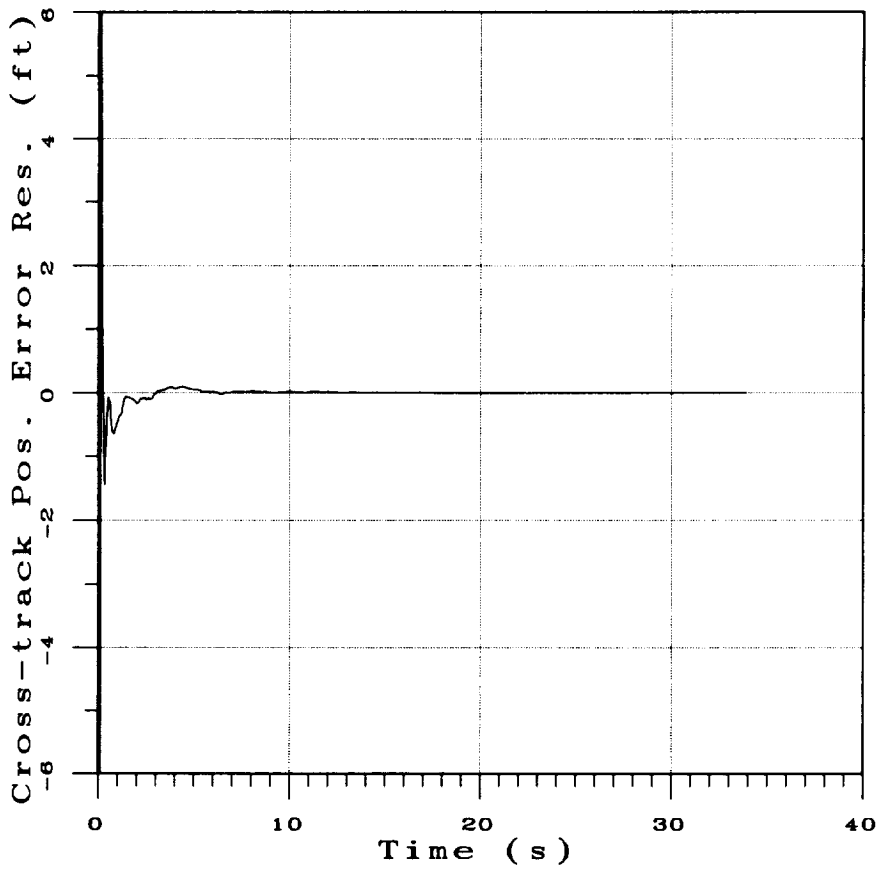

Figure 6.2: Cross–track position error using Algorithm V.

Figure 6.3 illustrates that the altitude estimates converge to within ±5 feet in less than two seconds.

The along–track velocity error residual is shown in Figure 6.4. It may be seen that the along–track velocity estimates settle to within ±10 feet/second in less than six seconds.

The cross–track velocity error residual in Figure 6.5 shows that the cross–track

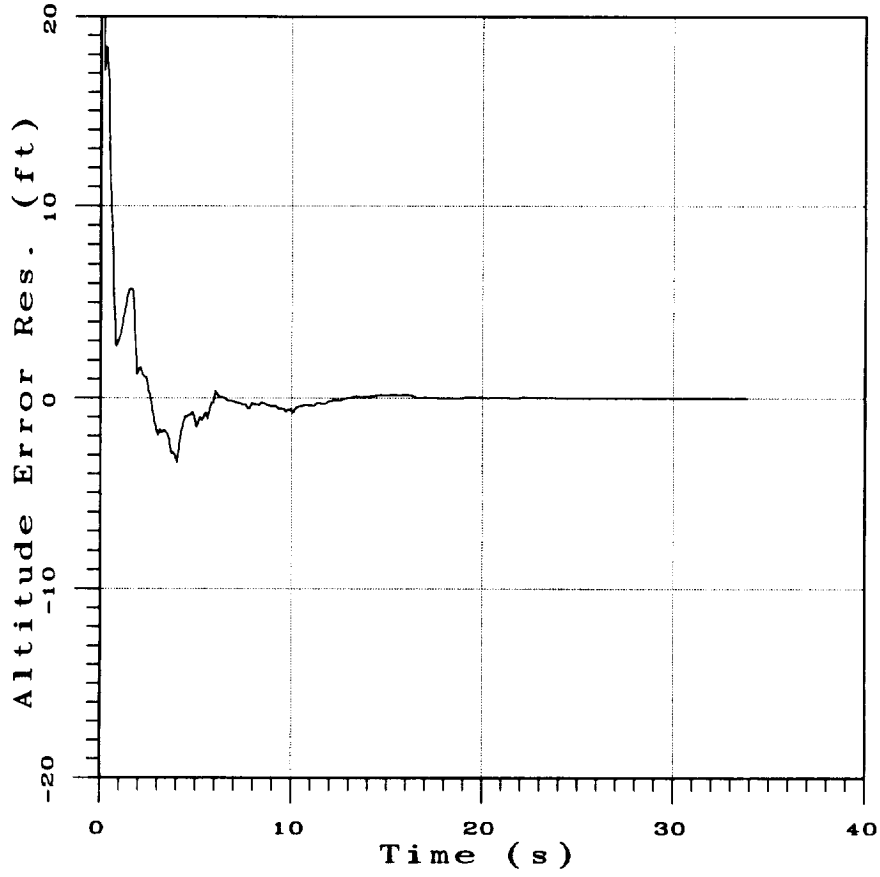

Figure 6.3: Altitude error using Algorithm V.

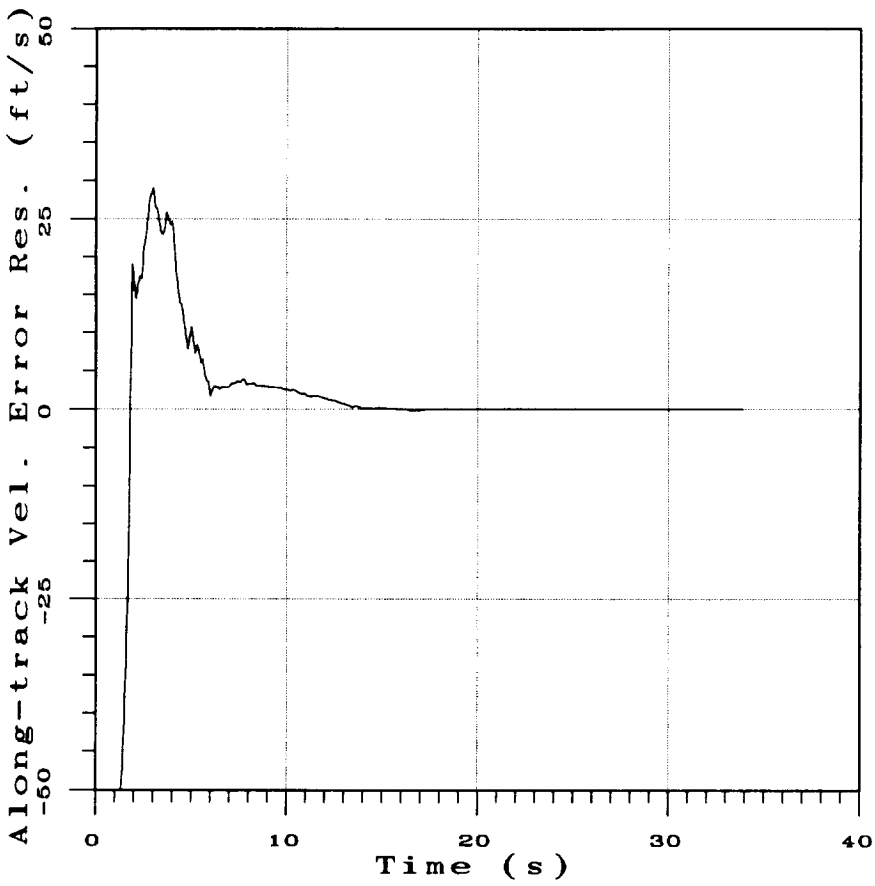

Figure 6.4: Along–track velocity error using Algorithm V.

velocity settles to within ±5 feet/second in less than a second.

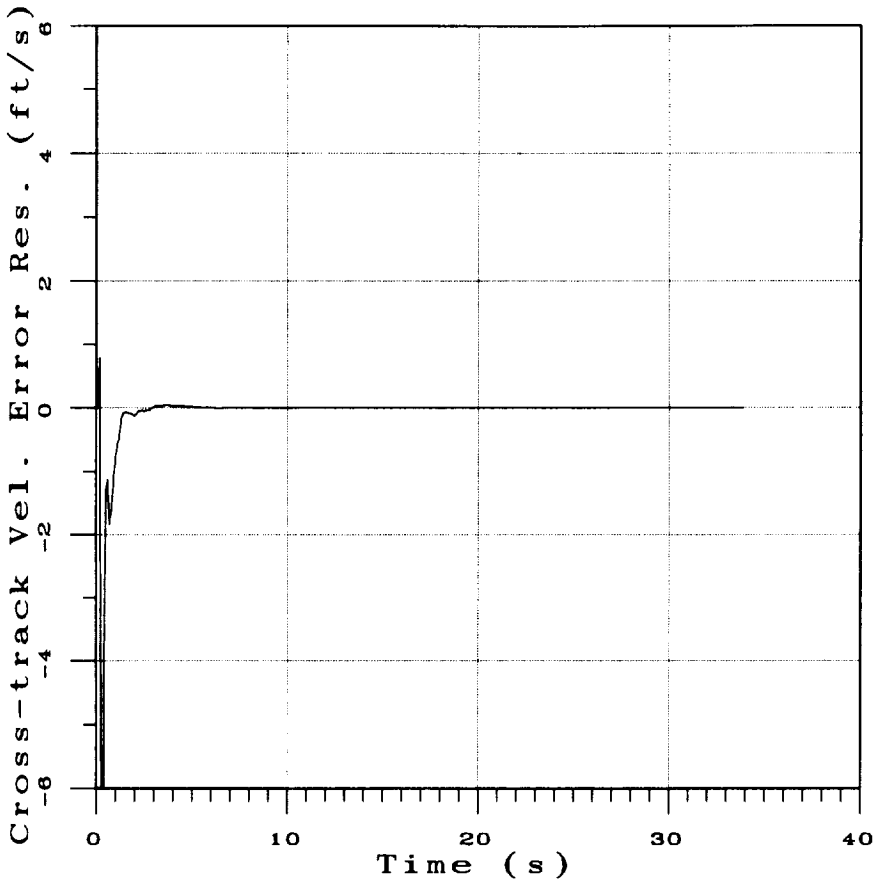

Figure 6.5: Cross-track velocity error using Algorithm V.

The sink rate error residual in Figure 6.6 shows that the sink rate error also is reduced to within ±5 feet/second in less than a second.

The position error residuals are summarized in Table 6.1. The notation and units are same as those in the previous tables. Comparing Table 6.1 to Table 3.2, it may be seen that the position estimates resulting from Algorithm V meet the navigation accuracy requirements for all Categories. Table 6.1 shows that amongst the five algorithms discussed so far in this report, Algorithm V provides the most accurate aircraft position estimates.

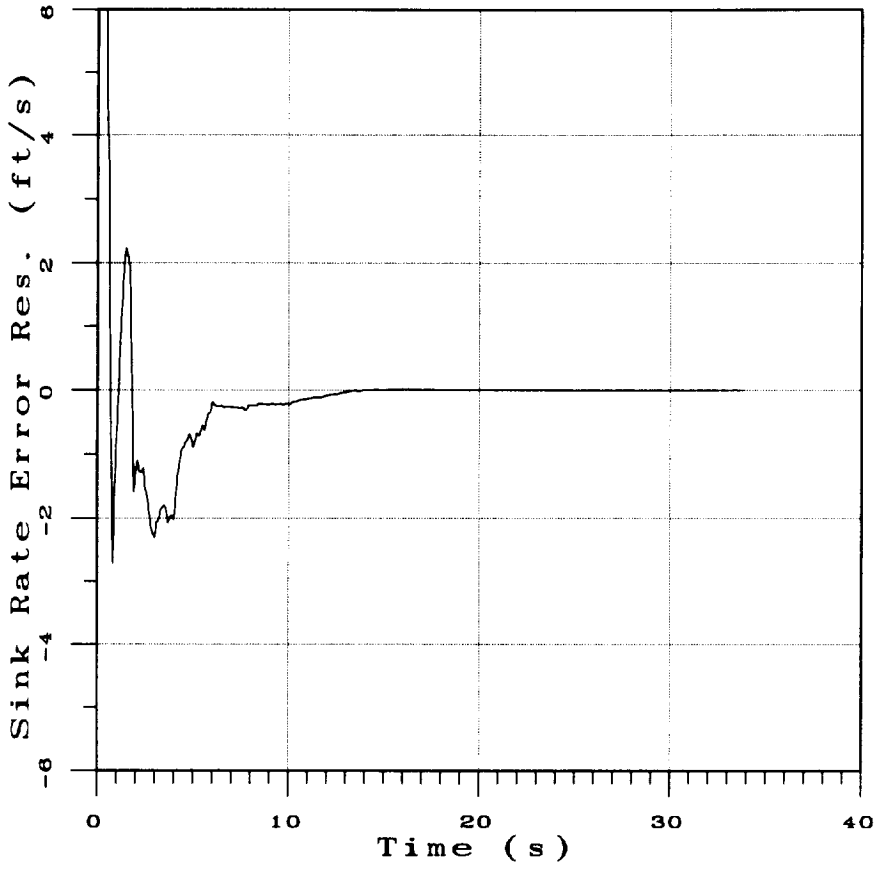

Figure 6.6: Sink rate error using Algorithm V.

Table 6.1: Algorithm V Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±0.02	±0.76
CAT II	±0.004	±0.04
CAT IIIa	±0.003	±0.01
CAT IIIb & c	±0.0	±0.003

6.2 Algorithm VI

Like the previous algorithm, this algorithm also assumes that the aircraft orientation angles are known. In addition to the six measurements described in Equations (6.3) through (6.8), this algorithm assumes that three position components are provided by an onboard Global Positioning System (GPS) receiver.

The reasons for integrating the GPS with the vision based position determination algorithm are as follows. The GPS provided position of the aircraft can be used to initialize the Kalman filter. If during descent, airport lights are cut off due to foreground occlusion, the integrated algorithm would continue to provide estimates of the aircraft position using GPS measurements. If the GPS signals are blocked due to terrain obstacles such as mountains or buildings, the integrated system would continue to estimate aircraft position using the vision based system. Since commercial GPS systems with standard position service have an accuracy of 325 feet horizontally 95 percent of the time [30] and 560 feet vertically, the bias and the noise in the GPS position can be reduced by the integrated system. Thus, integration of the vision system with the GPS is motivated by robustness. Moreover, the integrated navigation system synergistically exploits the available data sources for position estimation.

Since the location of the airport is known, the GPS–based aircraft position can be used for estimating the runway relative aircraft position components $\tilde{x}_b$, $\tilde{y}_b$ and $\tilde{z}_b$. These can be modeled as:

$$\tilde{x}_b = x_b + b_x + \eta_x \tag{6.74}$$

$$\tilde{y}_b = y_b + b_y + \eta_y \tag{6.75}$$

$$\tilde{z}_b = z_b + b_z + \eta_z \tag{6.76}$$

The GPS measurement model includes the bias terms b_x, b_y and b_z, and white noise terms, η_x, η_y and η_z. The GPS–based position components are directly used

as three additional measurements z_7, z_8 and z_9 in the present formulation. Thus,

$$z_7 = \tilde{x}_b \tag{6.77}$$

$$z_8 = \tilde{y}_b \tag{6.78}$$

$$z_9 = \tilde{z}_b \tag{6.79}$$

Based on Equations (6.74) through (6.76), the components of the measurement model vector h_7, h_8 and h_9 are:

$$h_7 = x_b + b_x \tag{6.80}$$

$$h_8 = y_b + b_y \tag{6.81}$$

$$h_9 = z_b + b_z \tag{6.82}$$

Assuming the bias terms to be constant during approach and landing, the following state equations can be used to model the bias terms:

$$b_x(k+1) = b_x(k) \tag{6.83}$$

$$b_y(k+1) = b_y(k) \tag{6.84}$$

$$b_z(k+1) = b_z(k) \tag{6.85}$$

The state vector for the algorithm consists of aircraft position and velocity components, together with the bias vector in the GPS. Thus,

$$\mathbf{X} = [\mathbf{X_b^i}, \mathbf{V_b^i}, \mathbf{B}]^T \tag{6.86}$$

$\mathbf{X_b^i}$ is the aircraft position vector, $\mathbf{V_b^i}$ is the aircraft inertial velocity vector and $\mathbf{B}$ is the GPS bias vector. Assuming constant inertial velocity and Euler integration

for state vector propagation results in the following 9×9 state transition matrix:

$$\mathbf{\Phi}(k) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.87}$$

Δt is the update time step.

With 9 elements of the measurement model vector, six image–based measurements defined in Equations (6.11) through (6.16) and three GPS positions in Equations (6.80) through (6.82), the complete observation matrix is:

$$\mathbf{H}(k) = \begin{bmatrix} \frac{\partial h_1}{\partial x_b} & \frac{\partial h_1}{\partial y_b} & \frac{\partial h_1}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial h_2}{\partial x_b} & \frac{\partial h_2}{\partial y_b} & \frac{\partial h_2}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial h_3}{\partial x_b} & \frac{\partial h_3}{\partial y_b} & \frac{\partial h_3}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial h_4}{\partial x_b} & \frac{\partial h_4}{\partial y_b} & \frac{\partial h_4}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial h_5}{\partial x_b} & \frac{\partial h_5}{\partial y_b} & \frac{\partial h_5}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial h_6}{\partial x_b} & \frac{\partial h_6}{\partial y_b} & \frac{\partial h_6}{\partial z_b} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.88}$$

The partial derivatives of h_1 through h_6 with respect to position components were discussed earlier in Equations (6.47) through (6.70).

The implementation of this Kalman filter differs significantly from Algorithm V because the image–based and GPS–based measurements are available at two different rates. The image–based measurements are typically available at the rate

of ten times a second, while the GPS measurements are available once a second. Hence, during one second, six image–based measurements z_1 through z_6 are available ten times, and the three GPS–based measurements z_7 through z_9 are available once. A multi–rate formulation of the Kalman filter [63] is required to deal with the changing dimension of the measurement vector.

The multi–rate formulation requires the observation matrix $\mathbf{H}(k)$ and the measurement noise covariance matrix $\mathbf{R}(k)$ to change with the number of measurements available at any one measurement epoch. For instance, the last three rows of the $\mathbf{H}(k)$ matrix in Equation (6.88) are eliminated when GPS measurements are unavailable. The noise covariance matrix $\mathbf{R}(k)$ is a 9×9 diagonal matrix when both GPS–based and image–based measurements are available. The first six diagonal elements contain the pixel noise variances, while the remaining three diagonal elements contain GPS measurement noise variances. The standard deviation of the pixel noise is assumed to be 0.5 pixels and the GPS measurement noise is three meters. It may be noted that the pixel variance is scaled by 0.1 second while the GPS measurement variance is scaled by 1 second. When only the image–based measurements are available, the last three rows and columns of the $\mathbf{R}(k)$ are eliminated in order to reduce its dimension to 6. The remaining matrices required for implementing the Kalman filter are defined below.

The control input vector is a 9×1 null vector. The input distribution matrix $\mathbf{\Gamma}(k)$ is set to be a 9×9 identity matrix. The disturbance distribution matrix $\mathbf{\Gamma_d}(k)$ is a 9×9 identity matrix. The process noise covariance matrix $\mathbf{Q}(k)$ is a 9×9 diagonal matrix. The diagonal elements are chosen as "tuning parameters" for the Kalman filter [35].

In order to begin the state estimation process, the 9×1 state vector $\hat{\mathbf{X}}(k)$ and its error covariance matrix $\mathbf{P}(k)$, of size 9×9, have to be initialized. Following the standard procedure in Kalman filtering, the state error covariance matrix is initialized by placing large variance values in the diagonal locations and setting the off–diagonal terms to zero. The Kalman filtering algorithm in Appendix A is

then used for recursive state estimation.

On a final note, it may be verified from Equation (A.5) in Appendix A that the dimension of the Kalman gain matrix changes from 9×6 to 9×9 depending on whether image–based measurements and/or GPS measurements are available. Thus, the structure of the Kalman filter is changed to accommodate the change in the number of measurements. This way, the Kalman filter is always updated at the fastest measurement rate.

6.2.1 Results using Algorithm VI

The position, velocity and GPS bias estimates obtained by Algorithm VI are described in this section. The estimates were obtained for the same simulation scenario used in all the previous algorithms. The GPS position estimates were used for initializing the Kalman filter. In order to simulate GPS measurements, position bias with a uncertainty of ± 100 meters and Gaussian white noise with a standard deviation of three meters were added to the aircraft position vector. The GPS position bias errors of 82 feet in the along–track position, 177 feet in the cross–track position and 142 feet in the altitude were assumed for generating the results given in this section.

The along–track position error residual is shown in Figure 6.7. This figure shows that the along–track position estimates converge to within ± 100 feet in two seconds.

The cross–track position error residual given in Figure 6.8 shows that the cross–track position estimate converges to within ± 5 feet in less than one second.

Figure 6.9 illustrates that the altitude estimates converge to within ± 5 feet in less than two seconds.

The along–track velocity error residual is shown in Figure 6.10. It may be seen that the along–track velocity estimates settle to within ± 10 feet/second in less than two seconds.

The cross–track velocity error residual in Figure 6.11 shows that the cross–

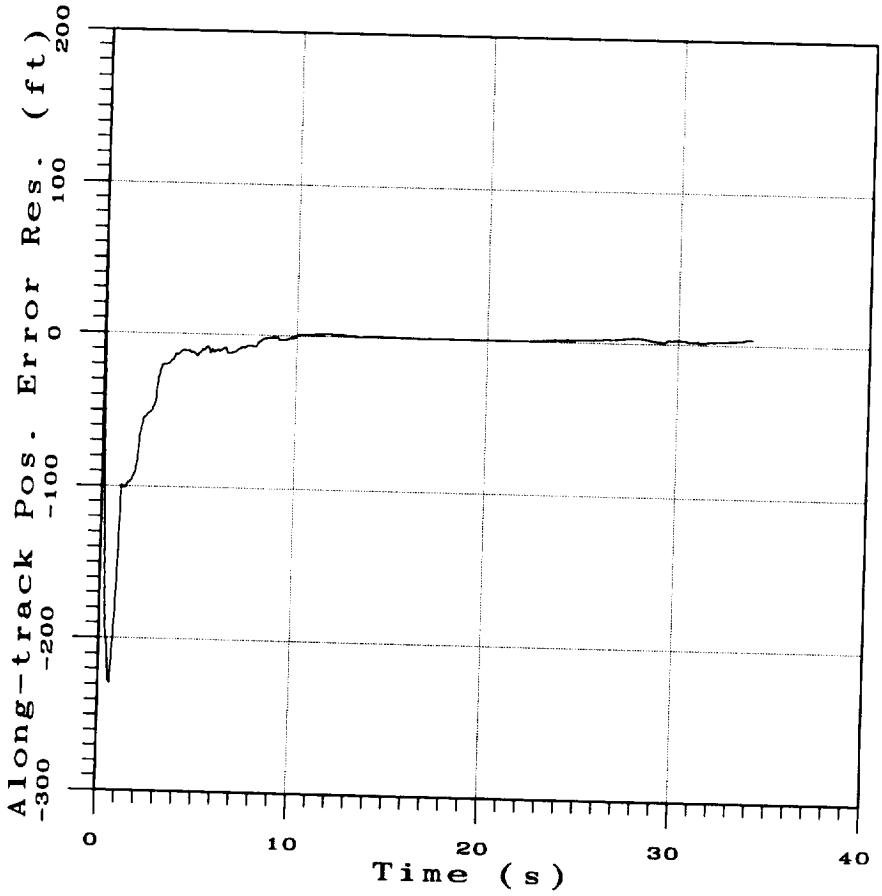

Figure 6.7: Along-track position error using Algorithm VI.

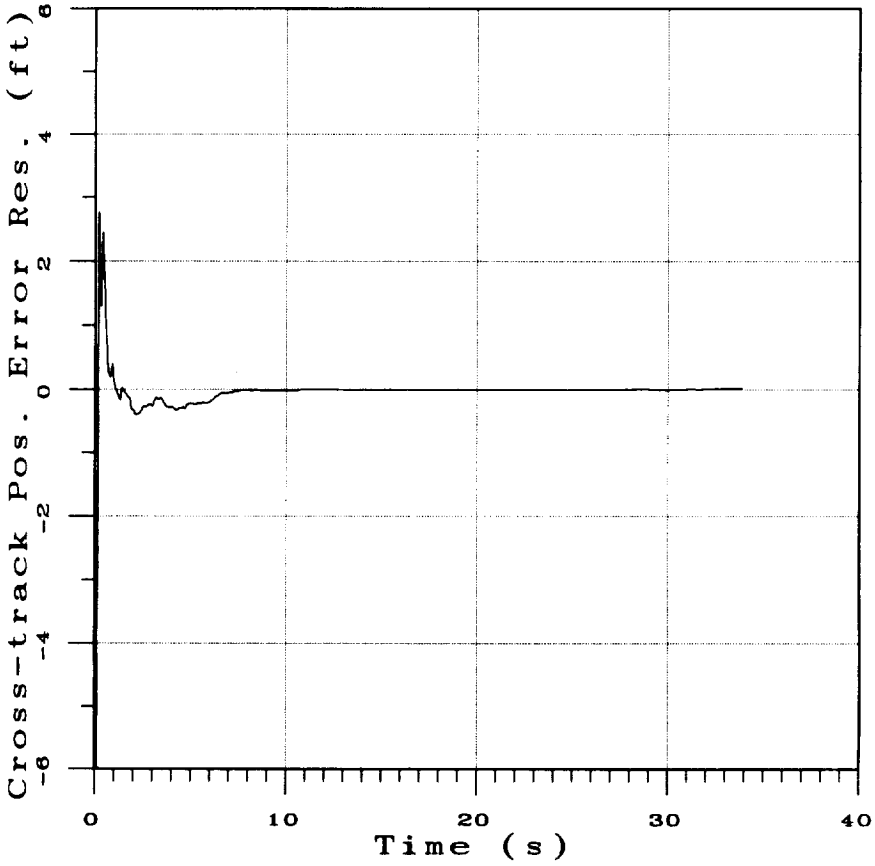

Figure 6.8: Cross-track position error using Algorithm VI.

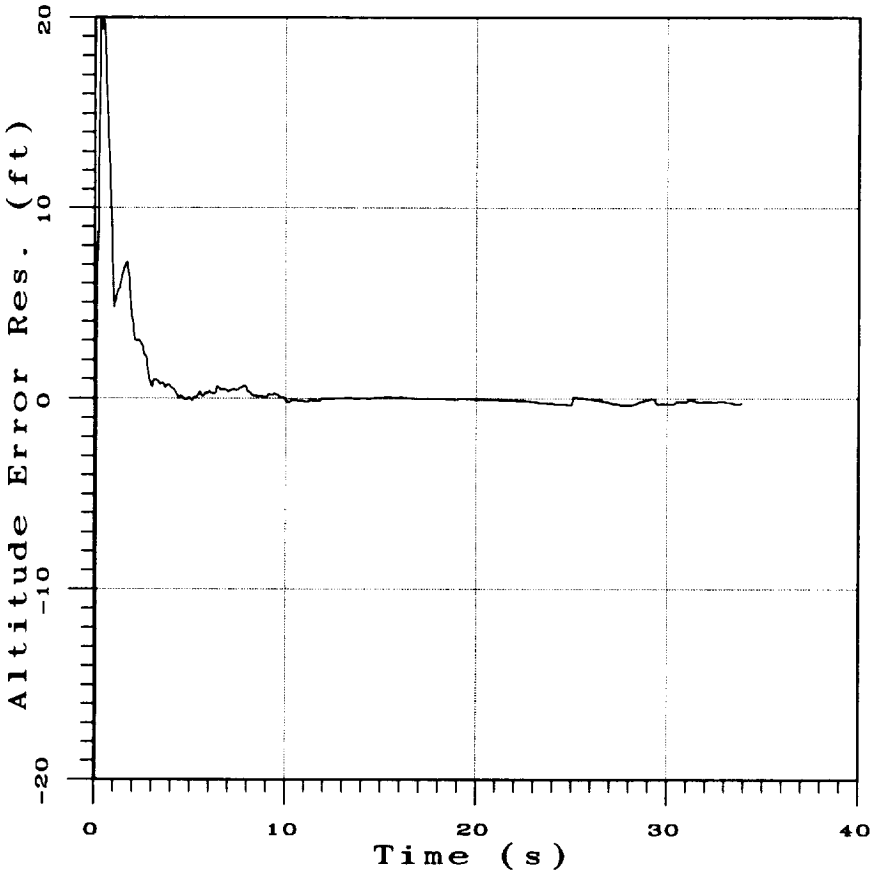

Figure 6.9: Altitude error using Algorithm VI.

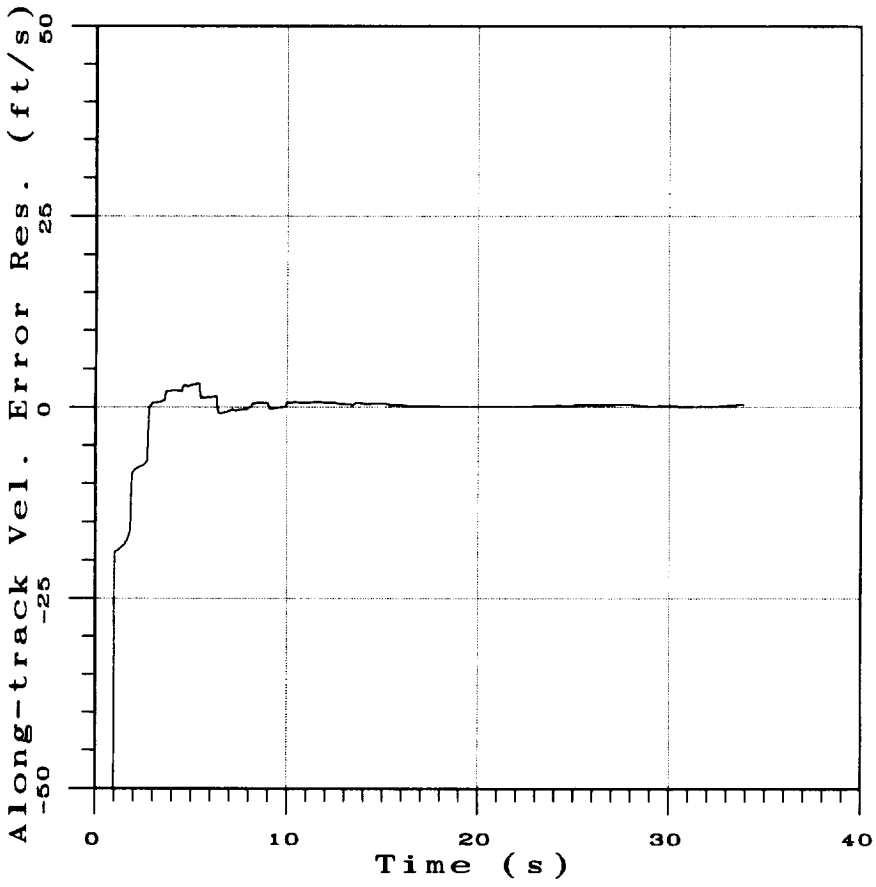

Figure 6.10: Along-track velocity error using Algorithm VI.

track velocity settles to within ±5 feet/second in less than a second.

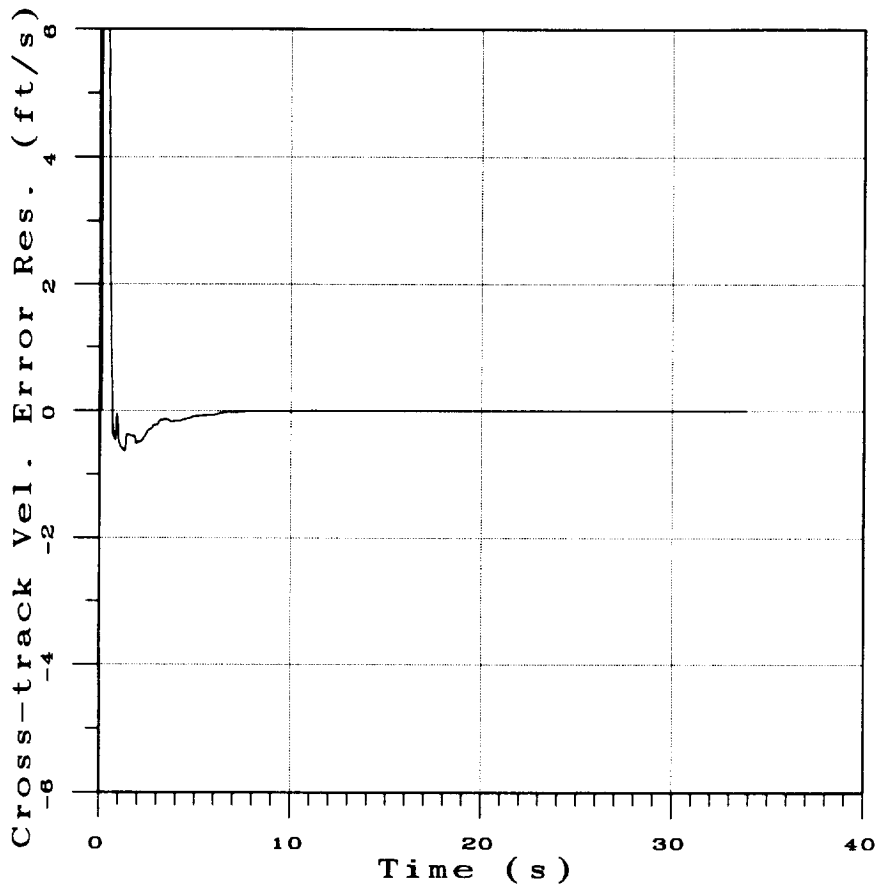

Figure 6.11: Cross-track velocity error using Algorithm VI.

The sink rate error residual shown in Figure 6.12 reveals that the sink rate error also is reduced to within ±5 feet/second in less than two seconds.

The GPS bias error residuals in the along-track and cross-track positions and in the altitude are shown in Figures (6.13), (6.14) and (6.15). The GPS bias estimates in the along-track position settles to within ±20 feet in less than six seconds. The GPS bias estimates in the cross-track position and altitude converge to within ±10 feet in less than 17 seconds and three seconds respectively. The altitude bias error increases beyond ±10 feet when the aircraft is 36 feet above the runway.

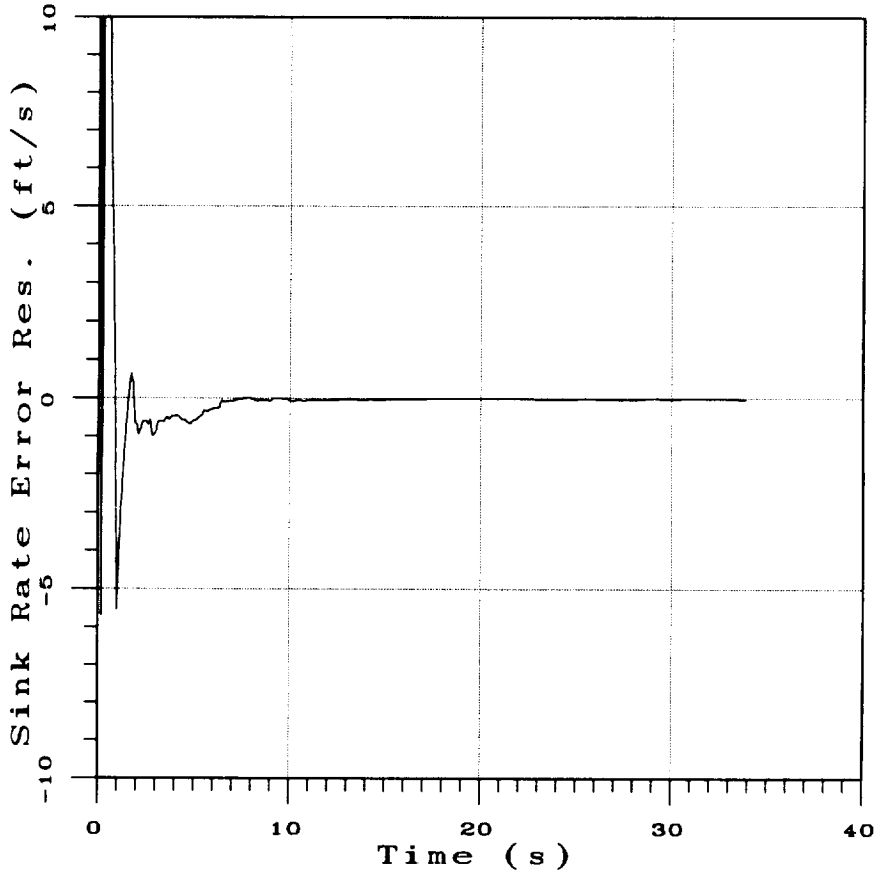

Figure 6.12: Sink rate error using Algorithm VI.

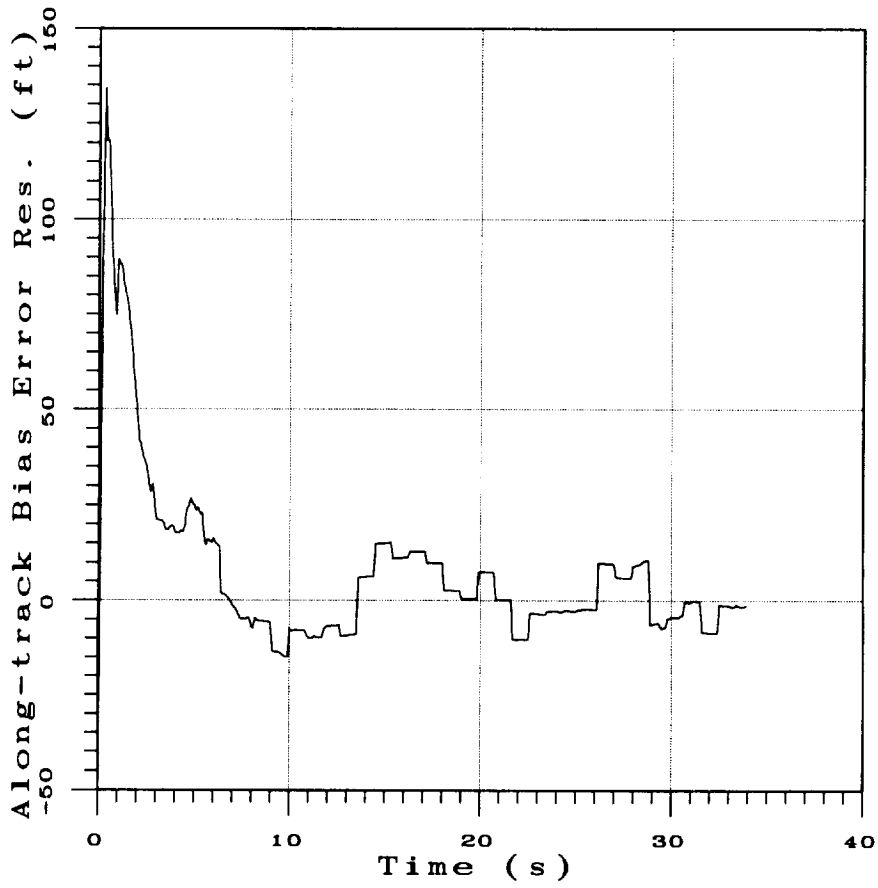

Figure 6.13: GPS bias in the along-track position using Algorithm VI.

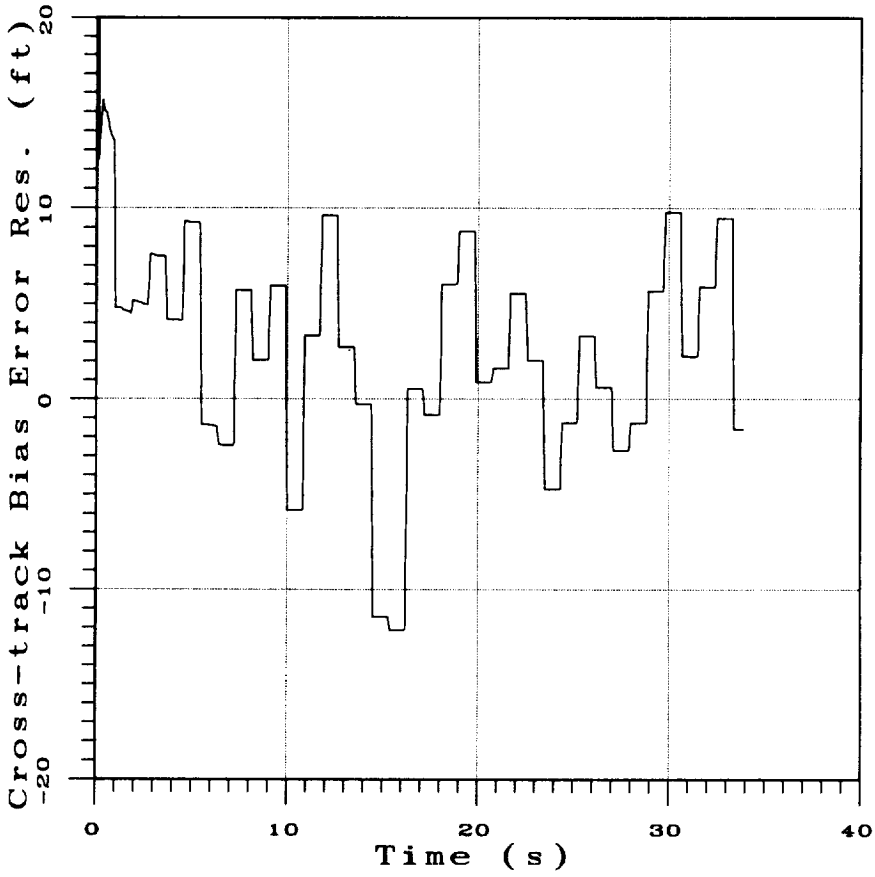

Figure 6.14: GPS bias in the cross-track position using Algorithm VI.

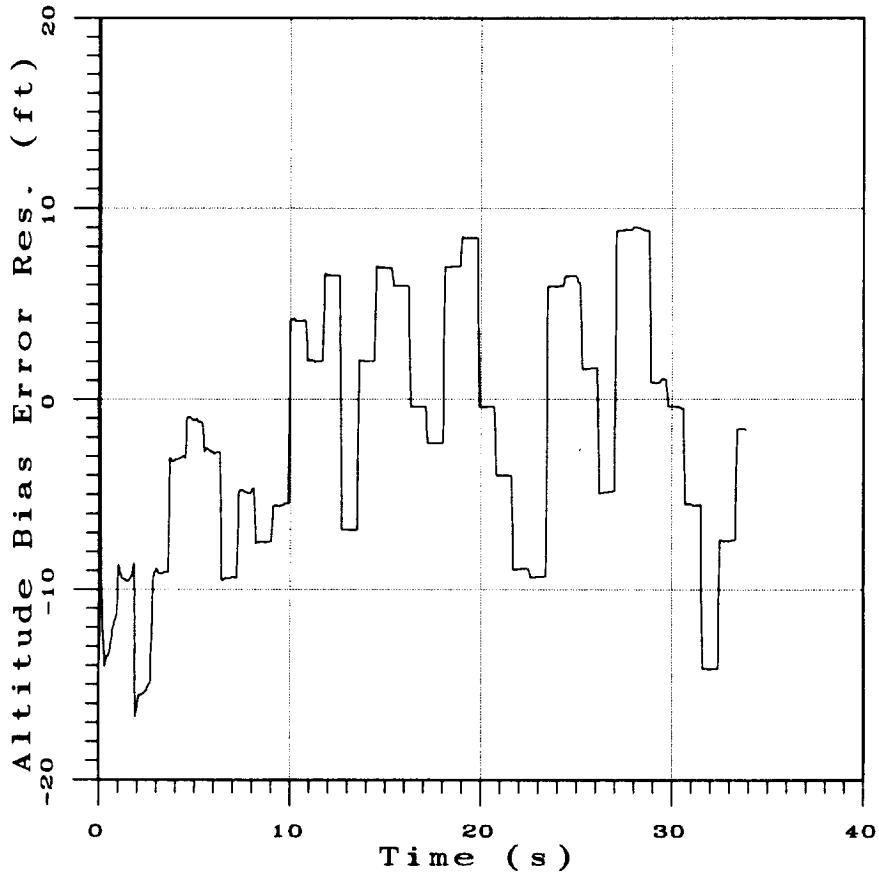

Figure 6.15: GPS bias in the altitude using Algorithm VI.

The position error residuals are summarized in Table 6.2. The notation and units are same as those in the previous tables. The usefulness of the integrated

Table 6.2: Algorithm VI Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±0.02	±0.24
CAT II	±0.01	±0.33
CAT IIIa	±0.01	±0.34
CAT IIIb & c	±0.02	±0.28

navigation system in aircraft operations can be assessed by comparing the achieved accuracy with the desired accuracies listed in Table 3.2. It may be observed from Table 6.2 that the present navigation scheme meets the objectives for all the Categories listed in Table 3.2. Table 6.2 shows that Algorithm VI is able to provides accurate aircraft position estimates. Moreover, Algorithm VI improves the GPS bias estimates considerably.

The results presented here show that the integrated GPS and image based algorithm is as accurate as the pure image based algorithm. The accuracy of the integrated algorithm can be further improved with a GPS receiver with a faster update rate. The integrated algorithm is expected to be fault tolerant in the event of imaging system failure.

6.3 Algorithm VII

In the two previous algorithms, the orientation angles were assumed to be known. In this final algorithm of the present work, the orientation angles are included as additional states to be estimated. The objective is to examine the degree to which the image based position determination concept can be extended. As discussed in Section 2.3.7.1, it is unlikely that all three attitudes can be reliably estimated without explicit correspondence between several points in the camera–

based and model–based images. Hence, in the present work, the attempt will be to estimate the aircraft pitch and yaw angles, with the assumption that the roll angle is available from measurements. It needs to be mentioned here that the attempts at including all the three attitudes did not yield useful results. With the addition of orientation states, this algorithm can be considered to be an extension of Algorithm V.

Using Euler Equations (3.24) through (3.26) describing the attitude kinematics, the discrete time state equation for attitudes can be obtained as:

$$\begin{pmatrix} \psi(k+1) \\ \theta(k+1) \\ r(k+1) \\ q(k+1) \end{pmatrix} = \begin{bmatrix} 1 & 0 & \cos\phi(k)\sec\theta(k)\Delta t & \sin\phi(k)\sec\theta(k)\Delta t \\ 0 & 1 & -\sin\phi(k)\Delta t & \cos\phi(k)\Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \psi(k) \\ \theta(k) \\ r(k) \\ q(k) \end{pmatrix} \tag{6.89}$$

where r and q are the yaw and pitch angular rates, and Δt is the time step. The yaw and pitch angular rates are assumed to be constant during the landing phase. The Euler equation shown in Equation (6.89) was discretized using forward differences which resulted in an explicit scheme. Note that the Euler equation can also be discretized using backward differences to yield an implicit scheme. The advantage of an implicit scheme is that it is unconditionally stable with respect to step size [45]. An explicit scheme becomes unstable for large step sizes. The disadvantage of an implicit scheme is that it requires several computations at each time step. Due to this reason, the explicit scheme has been used here.

Since the state vector is formed by combining the translational position and velocity states x_b, y_b, z_b, v_{bx}, v_{by} and v_{bz} with the rotational states: ψ, θ, r and q,

the 10×10 state transition matrix is:

$$\mathbf{\Phi}(k) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cos\phi(k)\sec\theta(k)\Delta t & \sin\phi(k)\sec\theta(k)\Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\sin\phi(k)\Delta t & \cos\phi(k)\Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.90}$$

Since the quasi-steady state approximation in the orientation angles influence the state transition matrix, a further approximation is introduced in the propagation of the error covariance matrix during the time update step described in Equation (A.8) in Appendix A.

Using the measurement Equations (6.11) through (6.16) and the ten states, the 6×10 observation matrix can be constructed as:

$$\mathbf{H}(k) = \begin{bmatrix} \frac{\partial h_1}{\partial x_b} & \frac{\partial h_1}{\partial y_b} & \frac{\partial h_1}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_1}{\partial \psi} & \frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial \phi} & 0 & 0 \\ \frac{\partial h_2}{\partial x_b} & \frac{\partial h_2}{\partial y_b} & \frac{\partial h_2}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_2}{\partial \psi} & \frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial \phi} & 0 & 0 \\ \frac{\partial h_3}{\partial x_b} & \frac{\partial h_3}{\partial y_b} & \frac{\partial h_3}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_3}{\partial \psi} & \frac{\partial h_3}{\partial \theta} & \frac{\partial h_3}{\partial \phi} & 0 & 0 \\ \frac{\partial h_4}{\partial x_b} & \frac{\partial h_4}{\partial y_b} & \frac{\partial h_4}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_4}{\partial \psi} & \frac{\partial h_4}{\partial \theta} & \frac{\partial h_4}{\partial \phi} & 0 & 0 \\ \frac{\partial h_5}{\partial x_b} & \frac{\partial h_5}{\partial y_b} & \frac{\partial h_5}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_5}{\partial \psi} & \frac{\partial h_5}{\partial \theta} & \frac{\partial h_5}{\partial \phi} & 0 & 0 \\ \frac{\partial h_6}{\partial x_b} & \frac{\partial h_6}{\partial y_b} & \frac{\partial h_6}{\partial z_b} & 0 & 0 & 0 & \frac{\partial h_6}{\partial \psi} & \frac{\partial h_6}{\partial \theta} & \frac{\partial h_6}{\partial \phi} & 0 & 0 \end{bmatrix} \tag{6.91}$$

Other vectors and matrices needed for implementation of the Kalman filter are as follows. The control input vector is a 10×1 null vector. The input distribution matrix $\mathbf{\Gamma}(k)$ is a 10×10 identity matrix. The disturbance distribution matrix $\mathbf{\Gamma_d}(k)$ is also a 10×10 identity matrix. The dimension of the process noise covariance matrix $\mathbf{Q}(k)$ is 10×10.

Just as in algorithm V, six measurements z_1 through z_6 defined in Equations (6.3) through (6.8) are used. Thus, the 6×6 measurement noise covariance matrix $\mathbf{R}(k)$ in Algorithm V is also employed here.

The 10×1 state vector $\hat{\mathbf{X}}(k)$ and its error covariance matrix $\mathbf{P}(k)$, of size 10×10 are initialized using an approach identical to that in the six previous algorithms. The Kalman filtering algorithm is then used for recursive state estimation.

6.3.1 Results using Algorithm VII

The estimates of position, velocity, yaw and pitch orientation angles, and yaw and pitch body rates generated by Algorithm VII are described in this section. The estimates are obtained for the same simulation scenario used in all the previous algorithms except that the yaw orientation angle is set to -10 degrees. As before, the pitch orientation angle is set to -3 degrees. Errors of 1000 feet in the along–track position x_b and 100 feet in the cross–track position y_b and in the altitude $-z_b$ are assumed for initializing the Kalman filter. Initial values of the velocity components, the yaw and pitch orientation angles, and the yaw and pitch body rates are set to zero.

The along–track position error residual is shown in Figure 6.16. It may be observed that the along–track position estimates converge to within ± 100 feet in five seconds.

The cross–track position error residual is shown in Figure 6.17 reveals that the cross–track position estimate converges to within ± 5 feet in less than one second.

Figure 6.18 illustrates that the altitude estimates converge to within ± 12 feet in about six seconds.

The along–track velocity error residual is shown in Figure 6.19. It may be observed that the along–track velocity estimates settle to within ± 10 feet/second in 11 seconds.

The cross–track velocity error residual in Figure 6.20 shows that the cross–

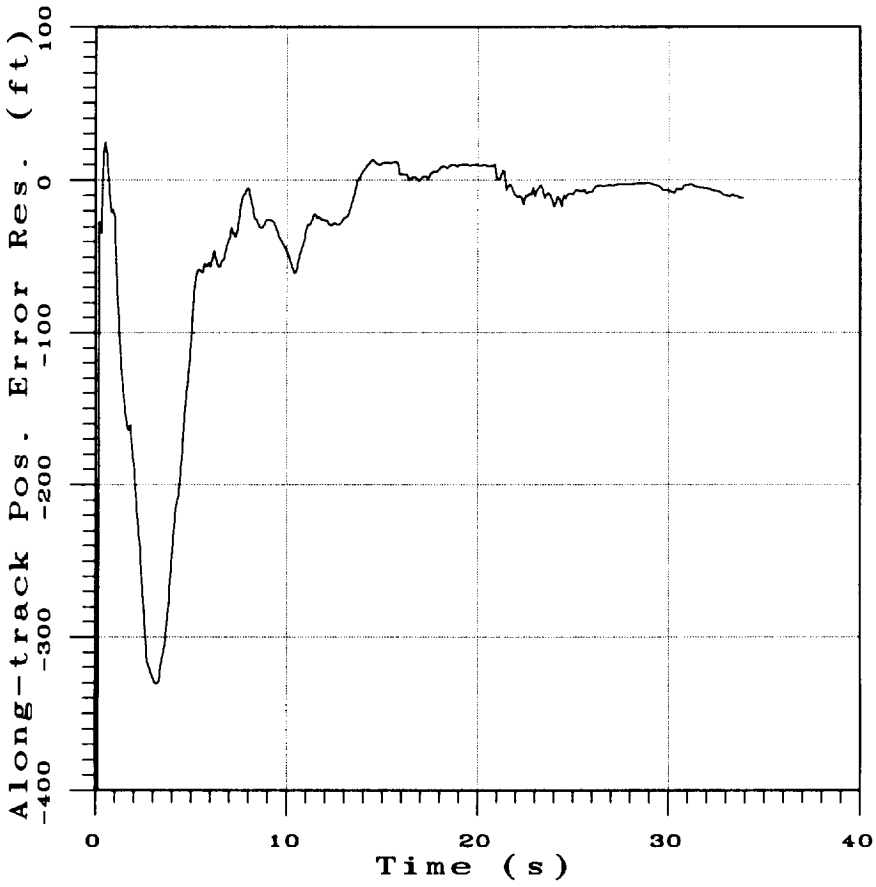

Figure 6.16: Along-track position error using Algorithm VII.

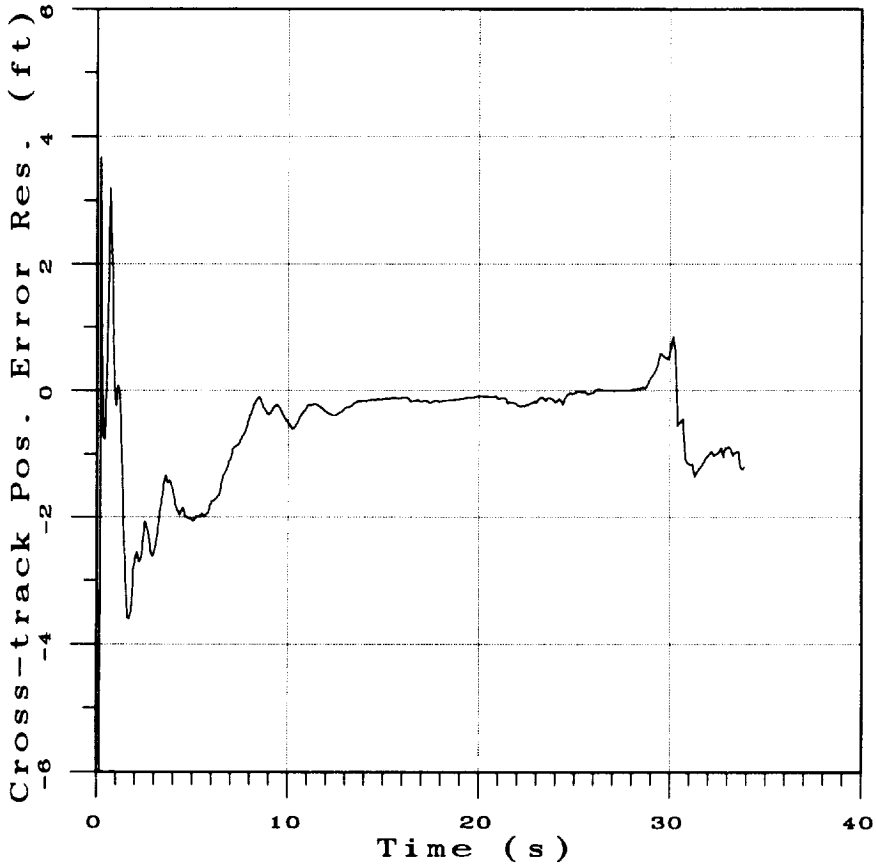

Figure 6.17: Cross-track position error using Algorithm VII.

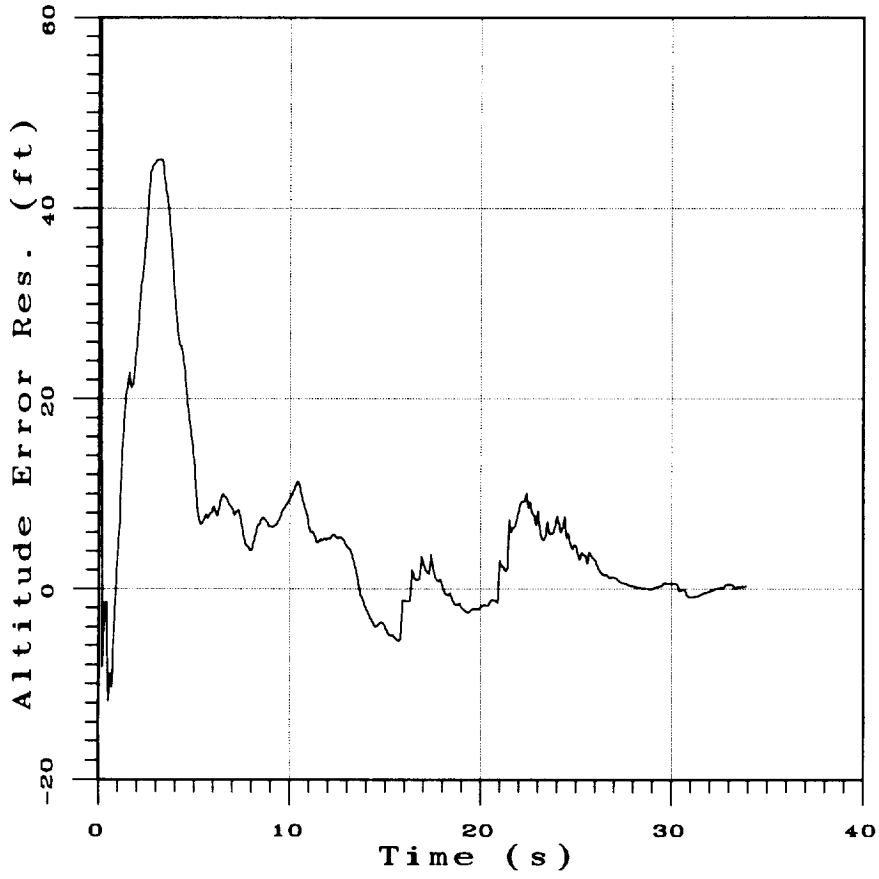

Figure 6.18: Altitude error using Algorithm VII.

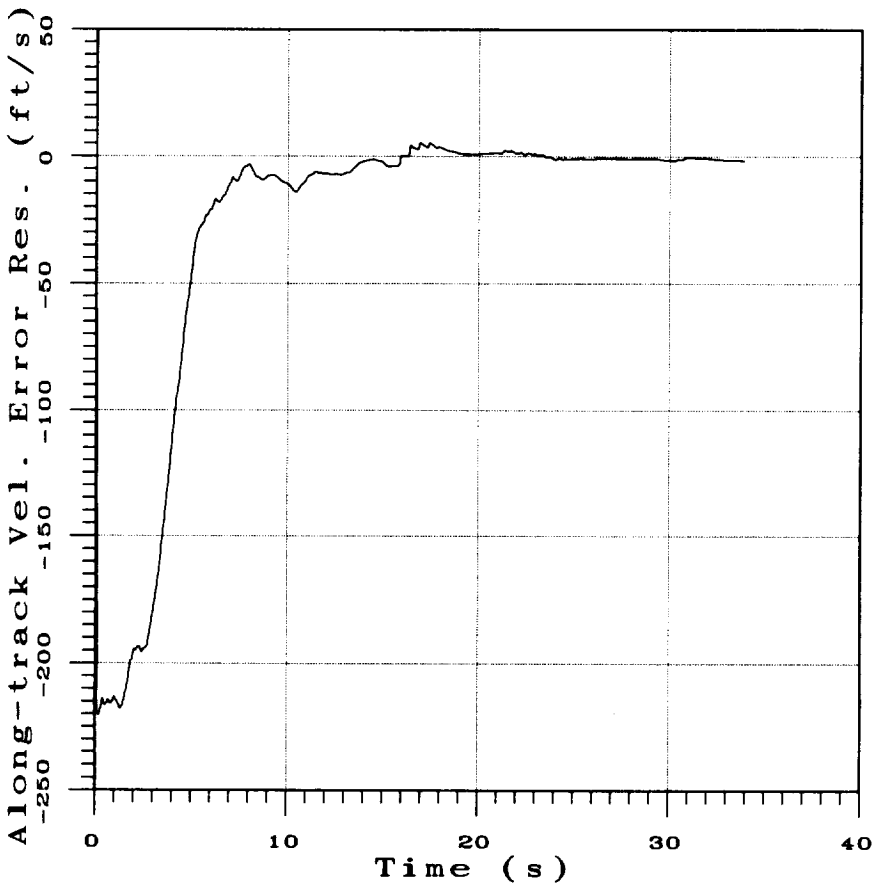

Figure 6.19: Along–track velocity error using Algorithm VII.

track velocity settles to within ±5 feet/second in about a second.

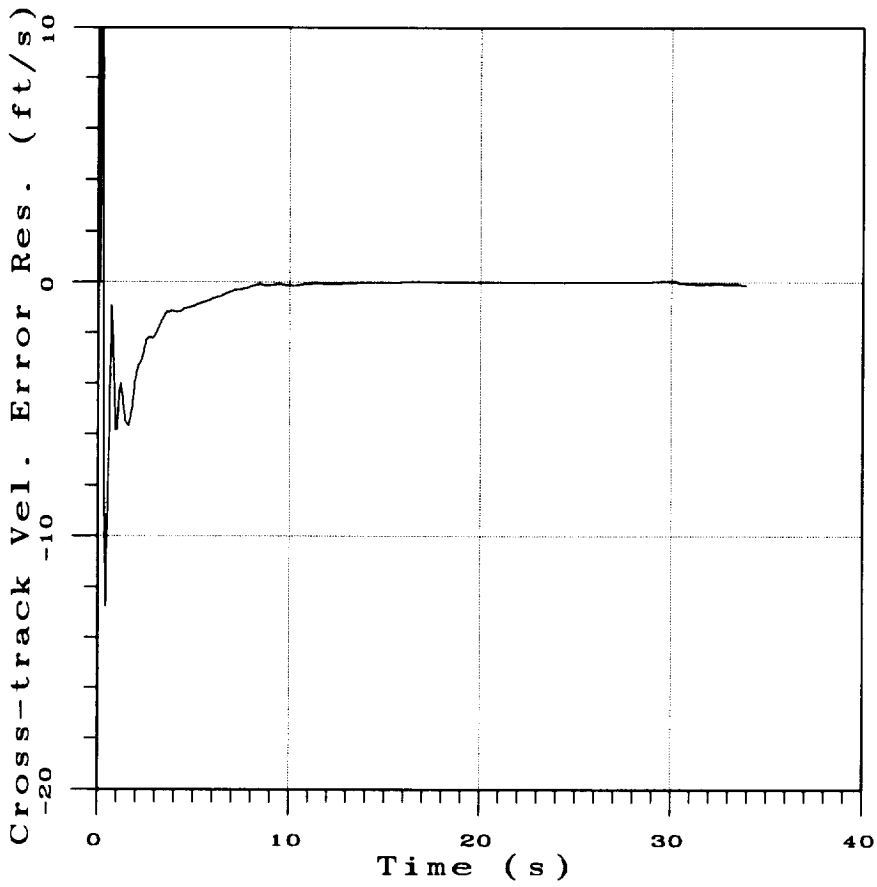

Figure 6.20: Cross–track velocity error using Algorithm VII.

The sink rate error is reduced to within ±5 feet/second in about five seconds as can be observed in Figure 6.21.

The aircraft yaw and pitch attitude error residuals are presented in Figure 6.22 and 6.23 respectively. Both these errors settle to within ±0.2 degree in less than a second.

Figures 6.24 and 6.25 show that the yaw and pitch body rate error residuals settle to within ±0.1 degrees/second in under two seconds.

The position error residuals are summarized in Table 6.3. The notation and units are same as those in the previous tables. By comparing Table 6.3 to Table

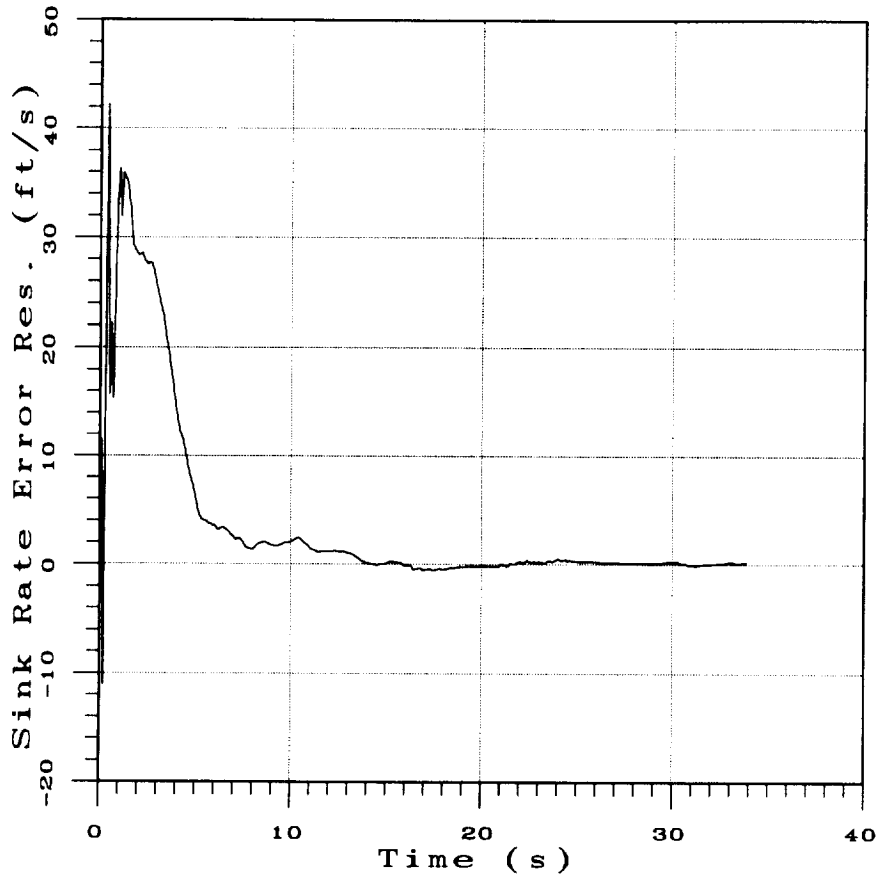

Figure 6.21: Sink rate error using Algorithm VII.

Table 6.3: Algorithm VII Results

Category	Lateral (y_b)	Vertical (h)
CAT I	±0.60	±11.29
CAT II	±0.25	±10.10
CAT IIIa	±0.85	±2.54
CAT IIIb & c	±1.36	±0.86

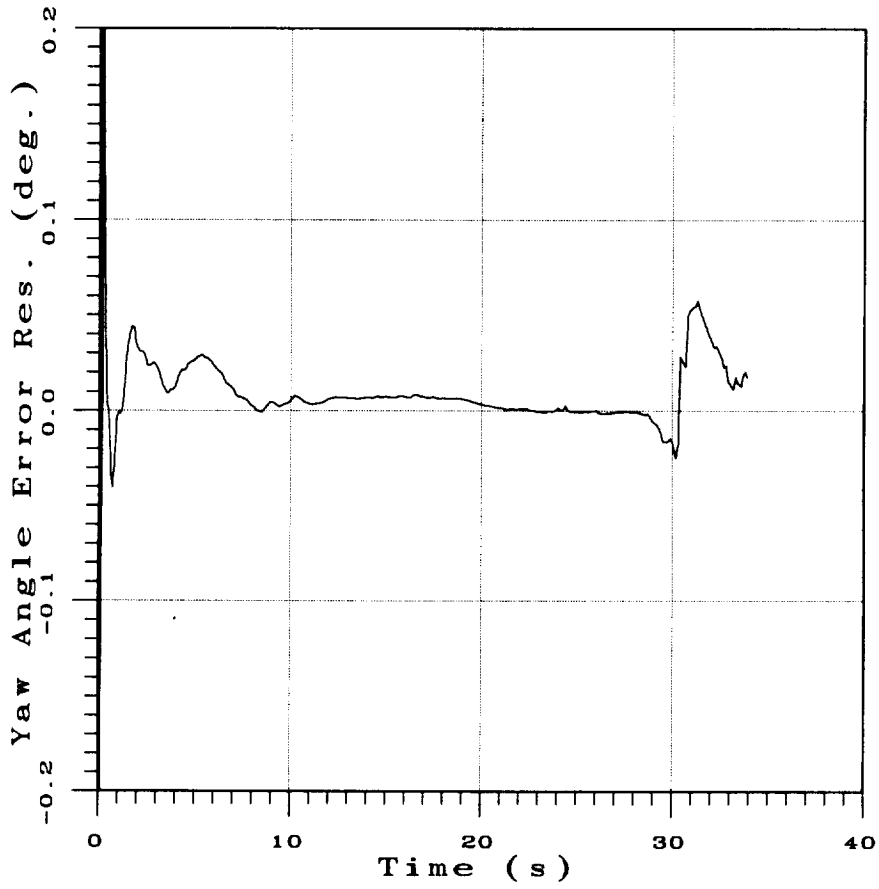

Figure 6.22: Yaw attitude estimates using Algorithm VII.

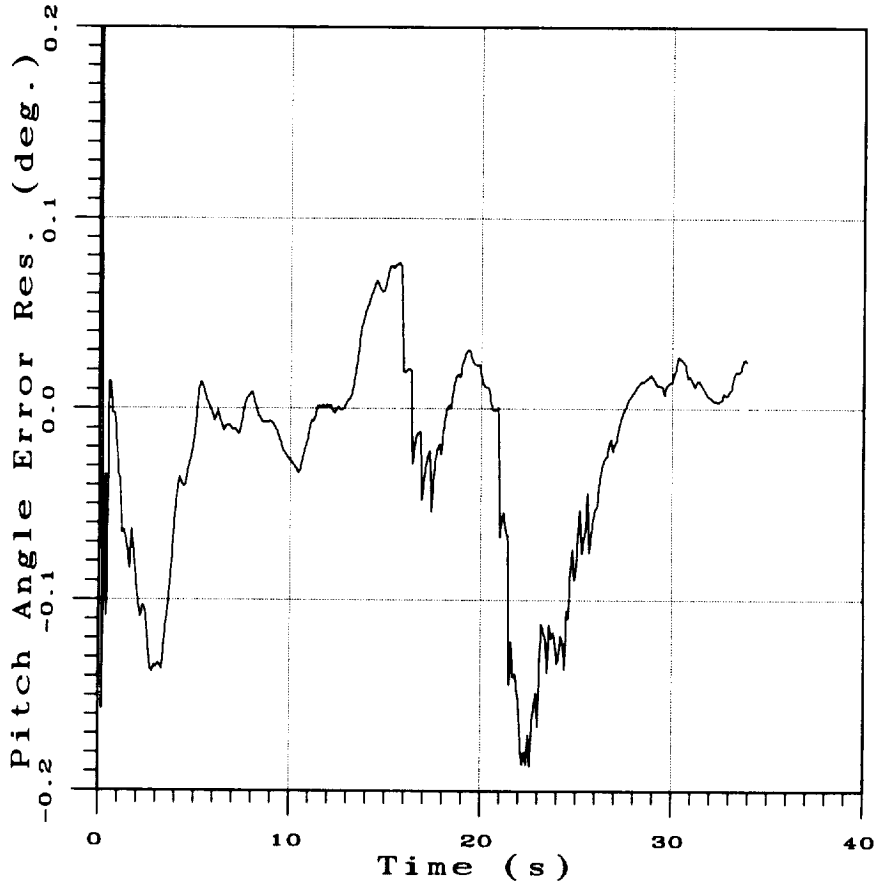

Figure 6.23: Pitch attitude estimates using Algorithm VII.

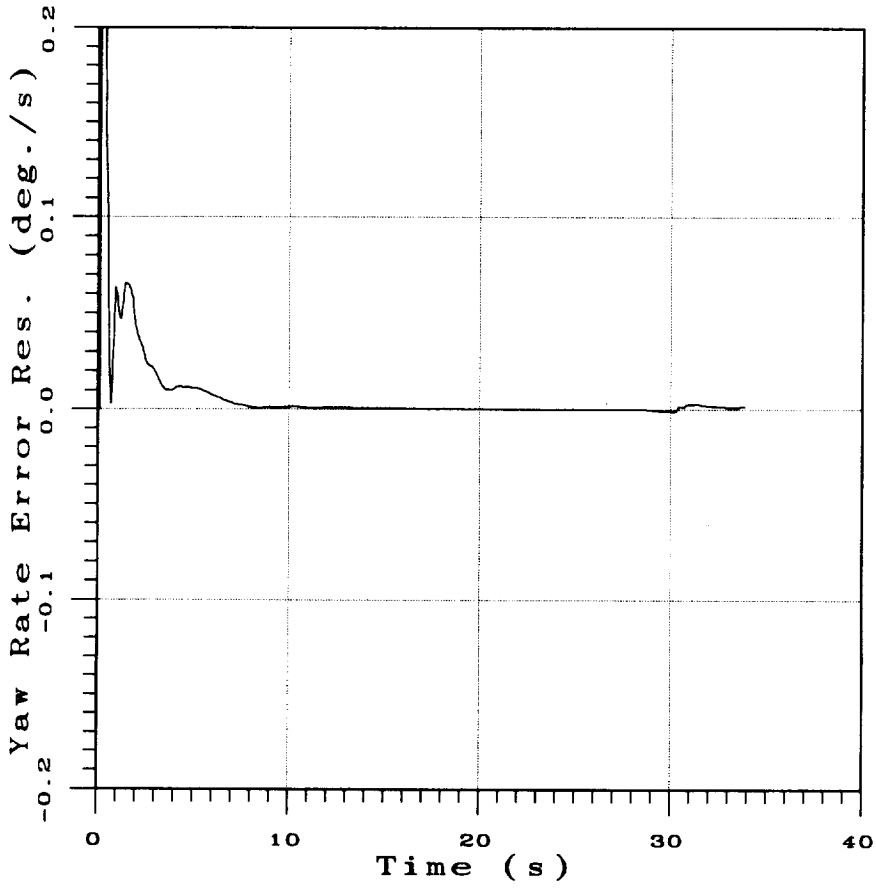

Figure 6.24: Yaw body rate using Algorithm VII.

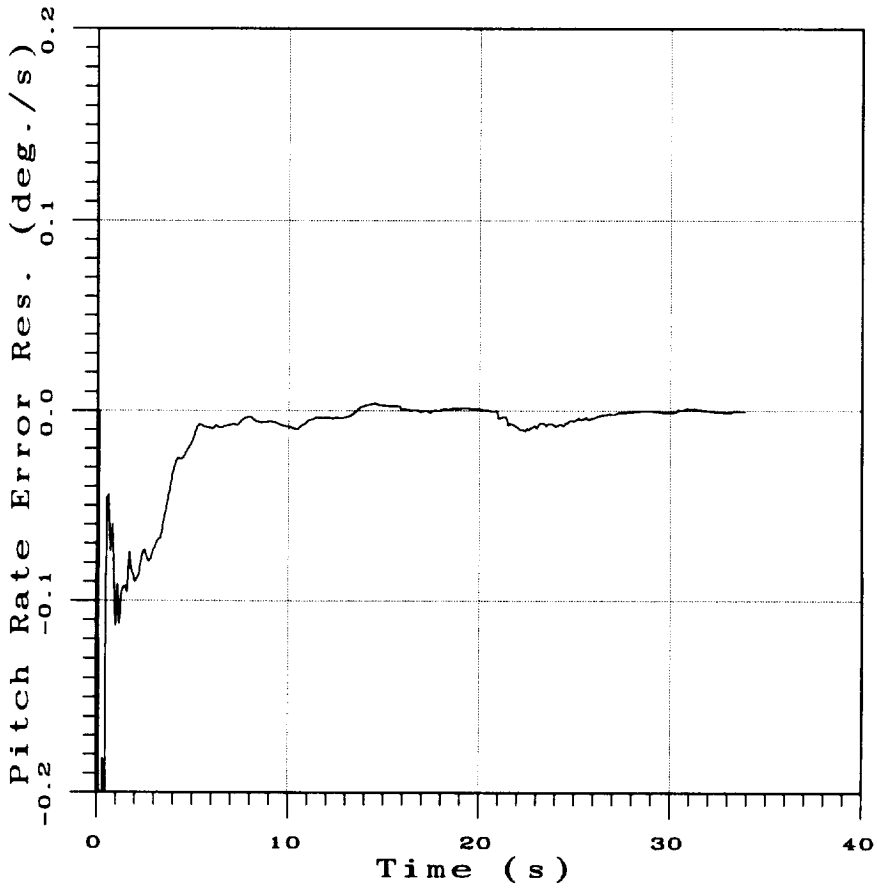

Figure 6.25: Pitch body rate using Algorithm VII.

3.2, it may be seen that the results obtained using Algorithm VII only satisfy Category I landing requirements. The results presented in this section show that Algorithm VII provides very accurate estimates of the aircraft attitude angles.

6.4 Summary

Three Kalman filtering centered algorithms were presented in this chapter. The first algorithm was designed to provide estimates of the aircraft position and velocity components. The second algorithm was designed to provide estimates of the aircraft position, velocity and GPS position bias components. Finally, the third algorithm was designed to provide estimates of the aircraft position and velocity components, yaw and pitch orientation angles, and yaw and pitch body rates.

All three algorithms used six shape features of the airport lighting layout as image-based measurements for the Kalman filters. Additionally, Algorithm VI used the three aircraft position components provided by the GPS receiver. Except for Algorithm VII, the other two algorithms also needed yaw, pitch and roll orientation angles. Algorithm VII only required the roll orientation angle.

Results were obtained to demonstrate the performance of the Kalman filters. It was shown that Algorithms V and VI are able to provide aircraft position estimates which meet Category I, II, IIIa, IIIb and IIIc navigation accuracy requirements. Aircraft position estimates generated using Algorithm VII were only able to meet Category I navigation accuracy requirements. However, Algorithm VII was able to provide highly accurate estimates of the yaw and pitch orientation angles.

Chapter 7

Contributions of the Report and Future Work

The contributions of this report and future research directions are discussed in this chapter.

7.1 Contributions of This Report

This report has explored the development of machine vision based pilot aids to help reduce night approach and landing accidents. The research focus was on developing an onboard instrument that complements the existing cockpit instrumentation.

The techniques developed during the course of this research were motivated by the desire to use the existing information sources to derive more precise aircraft position and orientation information. During night landing, the information source used by pilots for obtaining aircraft position and orientation information is the airport lighting layout. The fact that airport lighting geometry is known and since the images of the airport lighting can be acquired from an onboard camera, machine vision technology can be used for synthesizing a landing aid. Use of a machine vision system has several advantages. Firstly, such systems are not susceptible to

optical illusions. Moreover since the camera is a passive imaging device, it does not cause interference with the ground based equipment or with equipment onboard other aircraft. Finally, lowering costs of electro–optical cameras and real–time computer systems have made this technology attractive. Even if Global Positioning System (GPS) receivers become cheaper and more accurate, an integrated machine vision and GPS system would be a much more robust landing aid. The machine vision based system could also serve as a back–up landing system.

The main contribution of this research are the synthesis of seven navigation algorithms based on two broad families of solutions. The first family of solution methods comprise of techniques that reconstruct the airport lighting layout from the camera image and then estimate the aircraft position components by comparing the reconstructed lighting layout with the known model of the airport lighting layout. The second family of methods consist of techniques that synthesize the image of the airport lighting using a camera model and the known model of the airport lighting layout and then estimate the aircraft position components by comparing this synthesized image with the actual image of the airport lighting acquired by the onboard camera.

Algorithms I through IV belong to the first family of solutions while Algorithms V through VII belong to the second family of solutions. These algorithms can further be classified as parameter optimization methods, feature correspondence methods and Kalman filter centered methods respectively. Algorithms I and II are parameter optimization methods. Algorithms III and IV are feature correspondence methods. Algorithms V, VI and VII are Kalman filter centered methods. Figure 7.1 summarizes the algorithm classification.

Figure 7.1 shows the two classes of machine vision based landing aid developed in this report. First class provide only position information and second category provide both position and orientation information. Algorithms I and III provide the aircraft x and y inertial position components. They assume that the altitude information is available from an onboard altimeter. Algorithms II, IV, V and VI

Figure 7.1: Classification of the seven algorithms developed in this report.

compute all three aircraft inertial position components. Additionally, Algorithms V and VI provide estimates of the aircraft inertial velocity. Since Algorithm VI integrates image–based measurements with the position measurements from a GPS receiver, it also provides estimates of GPS position bias components. These can be used to improve the accuracy of GPS measurements.

Algorithms I through VI all assume that the yaw, pitch and roll attitude angles are available. Algorithm VII provides estimates of all three runway relative aircraft position and velocity components, the yaw and pitch orientation angles, and the yaw and pitch body rates. This algorithm only assumes that the roll attitude angle is available. Table 7.1 summarizes the aircraft states estimated by the seven algorithms developed in this report. The estimated quantities are marked with bullet. Note that velocity estimates for Algorithms I, II, III and IV are obtained by using a six–state Kalman filter driven by these algorithms. In Table 7.1 x_b, y_b and z_b are the inertial position components, v_{bx}, v_{by} and v_{bz} are the inertial velocity components, b_x, b_y and b_z are the GPS position bias components, ψ and θ are the yaw and pitch orientation angles, and r and q are the yaw and pitch body rates. Table 7.1 also indicates the nature of the computations required for each of these algorithms. Algorithms I, III and IV are direct computational schemes that do not require iterative computations. Algorithm II, V, VI and VII use iterative computational schemes.

In order to take advantage of the aircraft dynamics and the multiple images available along the glide path, the estimates provided by Algorithms I, II, III and IV were used for driving a six–state Kalman filter for providing estimates of the aircraft position and inertial velocity components. Algorithms V, VI and VII are Kalman filter centered algorithms and were designed to implicitly utilize the aircraft dynamics and multiple images available along the glide path.

Results were presented to demonstrate the performance of all the seven algorithms developed in this report. It was shown that all the algorithms are able to meet some or all of the Federal Aviation Administration specified navigation

Table 7.1: States Estimated by the Seven Algorithms

State	Alg. I	Alg. II	Alg. III	Alg. IV	Alg. V	Alg. VI	Alg. VII
x_b	•	•	•	•	•	•	•
y_b	•	•	•	•	•	•	•
z_b		•		•	•	•	•
v_{bx}	•	•	•	•	•	•	•
v_{by}	•	•	•	•	•	•	•
v_{bz}		•		•	•	•	•
b_x						•	
b_y						•	
b_z						•	
ψ							•
θ							•
r							•
p							•
Iterative		•			•	•	•

accuracy requirements for various landing categories. Table 7.2 summarizes the performance of the seven algorithms in meeting the navigation accuracy requirements for various FAA categories (CAT) of landing in Table 3.2.

Table 7.2: Performance Of The Seven Algorithms

CAT	Alg. I	Alg. II	Alg. III	Alg. IV	Alg. V	Alg. VI	Alg. VII
I	•	•	•	•	•	•	•
II	•	•	•	•	•	•	
IIIa	•		•	•	•	•	
IIIb, c	•		•	•	•	•	

7.2 Practical Considerations

The algorithms reported in this research have been validated using simulated image sequences. By comparing an actual image in Figure 2.2 with the simulated image in Figure 3.10 it may be observed that the only difference between the two

images is that lights in the actual image occupy regions in the image while lights in the simulated image appear as point sources. Thus, the actual image can be processed through low–level algorithms to transform it to appear similar to the simulated image. One may conclude that for Algorithms I, II, III and VI reconstruction of the airport lighting layout based on an actual image would result in lights occupying regions on the plane of the runway. To convert these regions to points, the centroids of the regions have to be found. A circular template of the physical dimensions of an airport light can be used to determine the centroids. Once the centroids are found for an initial image, the position and velocity estimates provided by the Kalman filters can be used to aid local template matching in subsequent images. For Algorithms V, VI and VII, the image constructed from the model of the airport lighting using propagated position estimates can be used for aiding the search for centroids of the lights detected in the actual image. Template matching schemes or local clustering schemes can be used for determining the centroids of lights in the actual image.

The computational requirements for all the algorithms are modest. Of the several Kalman filters developed in this report, the largest one is a ten–state filter with six measurements used in Algorithm VII. Experience with ranging algorithms which track three position components of several hundred objects in the image using three–state Kalman filters for every object has shown that these algorithms can be made to work in real–time using inexpensive hardware [84].

It may be noted that Algorithms II, V, VI and VII need initial values of the aircraft position components. In the case of Algorithm VI, the initial estimates are provided by an onboard GPS receiver. Algorithm I or III can be used with a barometric altimeter for initializing Algorithm II and Algorithm IV can be used for initializing Algorithms V through VII.

7.3 Future Work

Since the seven algorithms developed in this report have been verified only in simulation, the next logical step would be to verify the performance of these algorithms using actual images of the airport lighting layout obtained by an onboard camera.

This report has employed four lighting structures, the left and right runway edge lights, centerline lights and the threshold bar lights, in Algorithm IV. Techniques need to be developed for detecting other lighting structures in order to extend Algorithm IV to estimate the yaw, pitch and roll orientation angles.

It may be possible to improve the estimation accuracy of Algorithms V, VI and VII by extending them to iterated Kalman filtering algorithms. This should specially be investigated for Algorithm VII in order to improve its altitude estimation accuracy.

The six features used in Algorithm VII were found to have very low sensitivity to the roll orientation angle. Other features based on higher order moments should be investigated for possible estimation of the roll orientation angle. Shape features such as perimeter, area, eccentricity and thinness [27, 53] should also be investigated for improving the robustness of Algorithms V, VI and VII and for possible roll orientation angle estimation using Algorithm VII.

The focus of this research was to develop a pilot aid for flight on or near the glide slope. Hence, it was assumed the aircraft is headed in the approach direction of the runway and that only the airport lights are visible in the image. This report has not addressed the question of initial acquisition of the airport lighting layout when the aircraft is not lined up in the approach direction of the runway. Since the heading of the airport is known from published charts [54] and the heading of the airplane is known from cockpit instruments such as a gyro compass, the pilots may be able to identify the airport. Once they identify the airport and line up the aircraft in the approach direction present algorithms can be initiated. It may

be possible to develop a pilot aid for detecting the airport by using the heading information and images of the ground lighting layout. Color images may also be beneficial for airport detection because both approach and runway lights are color coded.

Bibliography

[1] Ahuja, N., and Rosenfeld, A., "A Note on the Use of Second-Order Gray-Level Statistics for Threshold Selection," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 8, No. 12, December 1978, pp. 895–898.

[2] Amini, A. A., Weymouth, T. E. and Jain, R. C., "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 9, September 1990, pp. 855–867.

[3] Anderson, B. D. O., and Moore, J. B., **Optimal Filtering**, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[4] Andrews, H. C., **Introduction to Mathematical Techniques in Pattern Recognition**, Wiley-Interscience, New York, NY, 1972.

[5] Baker, S. P., Lamb, M. W., Li, G., and Dodd, R. S., "Human Factors in Crashes of Commuter Airplanes," *Aviation, Space and Environmental Medicine*, January 1993, pp. 63–68.

[6] Ballard, D. H., and Brown, C. M., **Computer Vision**, Prentice-Hall, Englewood Cliffs, New Jersey, NJ, 1982.

[7] Barniv, Y., "Velocity Filtering Applied to Optical Flow Calculations," **Technical Memorandum 102802**, NASA Ames Research Center, Moffett Field, CA, August 1990.

[8] Bell, B. and Pau, L. F., "Contour Tracking and Corner Detection in a Logic Programming Environment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 9, September 1990, pp. 913–917.

[9] Bennett, C. T., and Schwirzke, M., "Analysis of Accidents During Instrument Approaches," *Aviation, Space and Environmental Medicine*, April 1992, pp. 253–261.

[10] Besl, P. J. and Jain, R. C., "Segmentation Through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, March 1988, pp. 167–190.

[11] Bhanu, B., Roberts, B., and Ming, J., "Inertial Navigation Sensor Integrated Motion Analysis," *Proceedings of Defense Advanced Research Projects Agency (DARPA) Image Understanding Workshop*, Palo Alto, CA, May 23–26, 1989, pp. 747–763.

[12] Binford, T. O.,"Survey of Model–Based Image Analysis Systems," *The International Journal of Robotics Research*, Vol. 1, No. 1, Spring 1982, pp. 18–64.

[13] Bryson, A. E., and Ho, Y. C., **Applied Optimal Control**, Blaisdell Pub. Co., Waltham, Massachusetts, 1969.

[14] Chatterji, G. B., and Sridhar, B., "Discrete Range Clustering Using Monte Carlo Methods," *Proceedings of IEEE Conference on Systems Man and Cybernetics*, Chicago, IL, October 18–21, 1992. Also in *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 6, November 1996, pp. 832–837.

[15] Chatterji G. B., Menon, P. K., and Sridhar, B., "Machine–Vision Based Pilot Aids for Night Landing and Takeoff," *Proceedings of AIAA Guidance, Navigation and Control Conference*, Scottsdale, AZ, August 1–3, 1994, pp. 1332–1339.

[16] Chatterji, G. B., Menon, P. K., and Sridhar, B., "Integrated GPS/Machine–Vision Navigation System for Aircraft Night Operations," *Proceedings of AIAA Guidance, Navigation and Control Conference*, Baltimore, MD, August 7–10, 1995, pp. 214–224.

[17] Chatterji, G. B., Menon, P. K., and Sridhar, B., "Vision-Based Position and Attitude Determination for Aircraft Night Landing," *AIAA Guidance, Navigation and Control Conference*, San Diego, CA, July 29–31, 1996.

[18] Chellappa, R., and Shekar, C., "Passive Ranging using a Moving Camera," *Journal of Robotic Systems*, 9, 000–000, October 1992.

[19] Chen, P.C., and Pavlidis, T. "Segmentation by Texture Using A Co–Occurrence Matrix and a Split–and–Merge Algorithm," *Computer Graphics and Image Processing*, Vol. 10, 1979, pp. 172–182.

[20] Chin, R. T., and Dyer, C. R., "Model–Based Recognition in Robot Vision," *Computing Surveys*, Vol. 18, No. 1, March 1986, pp. 67–107.

[21] Conners, R. W., and Harlow, C. A., "A Theoretical Comparison of Texture Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, No. 3, May, 1980, pp. 204–222.

[22] Defense Mapping Agency Aerospace Center, "IFR – Supplement United States," *Defense Mapping Agency Aerospace Center*, 3200 South Second Street, St. Louis, Missouri, 63118–3399.

[23] De Floriani, L., "Surface Representations Based on Triangular Grids," *The Visual Computer*, Vol. 3, 1987, pp. 27–50.

[24] Department of the Air Force, **Flying Training: Instrument Flying**, AFM 51–37, Headquarters US Air Force, Washington, DC 20330–5000.

[25] Dickmanns, E. D., and Graefe, V., **Dynamic Monocular machine Vision and Applications of Dynamic Monocular Machine Vision**, Report No. UniBwm/LRT/WE 12/FB/88-3, Universität der Bundeswehr München, Germany, July 1988.

[26] Dickmanns, E. D., "Machine Perception Exploiting High-Level Spatio-Temporal Models," *AGARD Lecture Series 185: machine Perception*, Hampton, VA, September 1992.

[27] Duda, R. O., and Hart, P. E., **Pattern Classification and Scene Analysis**, John Wiley, New York, NY, 1973.

[28] Faugeras, O., and Maybank, S. J., *IEEE Workshop on Visual Motion*, Newport Beach, CA, March 1989.

[29] Federal Aviation Administration, "Airport Design Standards-Site Requirements for Terminal Navigational Facilities," *Advisory Circular*, AC No: 150/5300-2D, Washington, DC, 1980.

[30] Federal Aviation Administration, "Airworthiness Approval of Global Positioning System (GPS) Navigation Equipment for Use as a VFR and IFR Supplemental Navigation System,"*Advisory Circular*, AC No: 20-138, U.S. Department of Transportation, May 25, 1994.

[31] Federal Aviation Administration, "Criteria for Approval of Category III Landing Weather Minima," *Advisory Circular 120-28C*, March 9, 1984.

[32] Federal Aviation Administration, "Criteria for Approving Category I and II Landing Minima for FAR 121 Operators," *Advisory Circular 120-29*, September 25, 1970.

[33] Federal Aviation Administration, "Runway and Taxiway Edge Lighting System," *Advisory Circular*, AC No: 150/5340-24, Washington, DC, 1977.

[34] Federal Aviation Administration, "Installation Details for Runway Centerline and Touchdown Zone Lighting Systems," *Advisory Circular*, AC No: 150/5340-4C, Washington, DC, 1978.

[35] Gelb, A., (ed.), **Applied Optimal Estimation**, The M. I. T. Press, Massachusetts Institute of Technology, Cambridge, MA, 1986.

[36] Geman, D., Geman, S., Graffigne, C. and Dong, P., "Boundary Detection by Constrained Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, July 1990, pp. 609–628.

[37] Gonzales, R. C., and Tou, J. T., **Pattern Recognition Principles**, Addison-Wesley, Reading, MA, 1981.

[38] Graham, R. L., Knuth, D. E., and Patashnik, O., **Concrete Mathematics**, Addison-Wesley, Menlo park, California, October 1992.

[39] Haddon, J. F., and Boyce, J. F., "Image Segmentation by Unifying Region and Boundary Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No.10, October 1990, pp. 929–948.

[40] Haines, R. F., and Flatau, C., **Night Flying**, McGraw-Hill, Blue Ridge Summit, PA, 1992.

[41] Hancock, E. R. and Kittler, J., "Edge-Labeling Using Dictionary-Based Relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, January 1990, pp. 165–181.

[42] Harlick, R. M., Shanmugam, R., and Dienstein, I., "Textural features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 6, November 1973, pp. 610–621.

[43] Harlick, R. M., "Statistical and Structural Approach to Texture," *Proceedings of IEEE*, Vol. 67, No. 5, May 1979, pp. 764–772.

[44] Harlick, R. M., "Monocular Vision Using Inverse Perspective Projection Geometry: Analytic Relations," CVPR, June 4–8, 1989, pp. 370–379.

[45] Hoffman, J. D., **Numerical Methods for Engineers and Scientists**, McGraw–Hill, New York, NY, 1992.

[46] Horn, B. K. P., and Schunck B. G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, 1981, pp. 185–203.

[47] Horn, B. K. P., **Robot Vision**, The MIT Press, McGraw–Hill, Cambridge, MA, 1987.

[48] Horn, B. K. P., "Motion Fields are Hardly Ever Ambiguous," *International Journal of Computer Vision*, Vol. 1, October 1987.

[49] Horonjeff, R., and McKelvey, F. X., **Planning and Design of Airports**, 3rd, ed., McGraw–Hill, San Francisco, CA, 1983, pp. 503–534.

[50] Horowitz, S. L. and Pavlidis, T., "Picture Segmentation by a Tree Traversal Algorithm," *Journal of the Association for Computing Machinery*, Vol. 23, No. 2, April 1976, pp. 368–388.

[51] Hussien, B. and Soursa, R., "Clustering Methods for Removing Outliers from Vision–Based Range Estimates," *Proceedings of SPIE Intelligent Robots and Computer Vision X: Neural, Biological and 3–D Methods*, Vol. 1608, Boston, MA, November 14–15, 1991, pp. 326–337.

[52] Inglis, A. F., **Video Engineering**, McGraw–Hill, New York, NY, 1993.

[53] Jain, A. K., **Fundamentals of Digital Image Processing** Prentice–Hall, Englewood Cliffs, NJ, 1989.

[54] **Jeppesen Airway Manual**, Jeppesen Sanderson, Inc., 1992.

[55] Johnson, L. W., Riess, R. D., and Arnold, J. T., **Introduction to Linear Algebra**, Addison–Wesley, Menlo Park, CA, 1993.

[56] Knuth, D. E., **Sorting and Searching; Volume 3 of The Art of Computer Programming**, Addison–Wesley, Menlo Park, CA, 1975.

[57] Kohler, R., "A Segmentation System Based on Thresholding," *Computer Graphics and Image Processing*, Vol. 15, 1981, pp. 319–338.

[58] Krishnapuram, R. and Casasent, D., "Determination of Three–dimensional object location and orientation from Range Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 11, November 1989, pp. 1158–1167.

[59] Kruskal, J. B., Jr., "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of American Mathematical Society*, No. 7, 1956, pp. 48-50.

[60] Kumar, R., and Hansen, A. R., "Model Extension and Refinement using Pose Recovery Techniques," *Journal of Robotic Systems*, 9, 000–000, October 1992.

[61] Kweon, I.S. and Kanade, T., "High–Resolution Terrain Map from Multiple Sensor Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, February 1992, pp. 278–293.

[62] Lacroix, V., "A Three–Module Strategy for Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, November 1988, pp. 803–810.

[63] Lewis, F. L., **Optimal Estimation with an Introduction to Stochastic Control Theory**, Wiley, New York, NY, 1986.

[64] McLain, D. H., "Two Dimensional Interpolation from Random Data," *Computer Journal*, Vol. 19, No. 2, 1976, pp. 178–181.

[65] Meer, P., Sher, C. A. and Rosenfeld, A., "The Chain Pyramid: Hierarchical Contour Processing," *IEEE Transactions Pattern Analysis Machine Intelligence*, Vol. 12, No. 4, April 1990, pp. 363–376.

[66] Menon, P. K. A., and Sridhar, B., "Passive Navigation Using Image Irradiance Tracking," *Proceedings of AIAA Guidance, Navigation and Control Conference*, Boston, MA, August 14–16, 1989, pp. 639–646.

[67] Menon, P. K. A., and Sridhar, B., "Image Based Range Determination," *Proceedings of AIAA Guidance, Navigation and Control Conference*, Portland, OR, August 20–22, 1990, pp. 786–796.

[68] Menon, P. K. A., Chatterji, G. B., and Sridhar, B., "A Fast Algorithm for Image–Based Ranging," *Proceedings of SPIE International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, Orlando, FL, April 1–5, 1991, pp. 190–200.

[69] Menon, P. K. A., Chatterji, G. B., and Cheng, V. H. L., "A Two–Time–Scale Autopilot for High–Performance Aircraft," *AIAA Guidance, Navigation and Control Conference*, New Orleans, LA, August 12–14, 1991.

[70] Menon, P. K. A., Chatterji, G. B., and Sridhar, B., "Passive Obstacle Location for Rotorcraft Guidance," *AIAA Guidance, Navigation and Control Conference*, New Orleans, LA, August 12–14, 1991. Also appeared as: "Electro–Optical Navigation for Aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 3, July 1993, pp. 825–833.

[71] Menon, P. K. A., Sridhar, B., and Chatterji, G. B., "Vision–Based Stereo Ranging as an Optimal Control Problem," *Proceedings of AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, August 10–12, 1992, pp. 594–604.

[72] Milgram, M. and Pierre, T. D. S., "Boundary Detection and Skeletonization with a Massively Parallel Architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, January 1990, pp. 74–78.

[73] Negahdaripour, S., "Multiple Interpretations of the Shape and Motion of Objects from Two Perspective Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 11, November 1990, pp. 1025-1039.

[74] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., **Numerical Recipes in C; The Art of Scientific Computing**, Cambridge University Press, New York, NY, 1990.

[75] Prim, R. C., "Shortest connection networks and some generalizations," *Bell Systems Technical Journal*, November 1957, pp. 1389–1401.

[76] Roberts, B., and Bhanu, B., "Inertial Navigation Sensor Integrated Motion Analysis for Autonomous Vehicle Navigation," *Journal of Robotic Systems*, 9, 000–000, October 1992.

[77] Rounds, E. M., and Sutty, G., "Segmentation based on second–order statistics," *Proceedings of the SPIE, Image Understanding Systems II*, Vol. 205, 1979, pp. 126–132.

[78] Rosenfeld, A., and Kak, A. C., **Digital Picture Processing**, Academic Press, New York, NY, 1976.

[79] Rosenfeld, A., and Davis, L. S., "Image Segmentation and Image Models," *Proceedings of IEEE*, Vol. 67, No. 5, May 1979, pp. 764–772.

[80] Sergerlind, L. J., **Applied Finite Element Analysis**, John Wiley & Sons, New York, NY, 1976.

[81] Skifstadt, K. and Jain, R., "Range Estimation from Intensity Gradient Analysis," *Machine Vision and Applications*, Vol. 2, No. 1, 1989, pp. 81–102.

[82] Smith, P. N., "Flight Data Acquisition for Validation of Passive Ranging Algorithms for Obstacle Avoidance," *American Helicopter Society Forum*, Washington, DC, May 21–23, 1990, pp. 311–319.

[83] Smith, P. N., Sridhar, B., and Suorsa, R. E., "Multiple-Camera/Motion Stereoscopy for Range Estimation in Helicopter Flight," *Proceedings of American Control Conference*, San Francisco, CA, June 2–4, 1993, pp. 1339–1343.

[84] Smith, P. N., and Sridhar, B., "Realtime Implementation of a Passive Ranging System," *SPIE* 10^{th} *Annual International AeroScense Symposium*, Orlando, Florida, April 8–12, 1996.

[85] Spiegel, M. R., **Mathematical Handbook of Formulas and Tables**, Schaum's Outline Series, McGraw-Hill, New York, NY, 1968.

[86] Sridhar, B., Cheng, V. H. L., and Phatak, A. V., "Kalman Filter Based Range Estimation for Autonomous Navigation Using Imaging Sensors," *Proceedings of IFAC Automatic Control in Aerospace*, Japan, 1989, pp. 45–50.

[87] Sridhar, B., and Suorsa, R., "Comparison of Motion and Stereo Methods in Passive Ranging Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 27, No. 4, July 1991, pp. 741–746.

[88] Sridhar, B., Suorsa, R., and Hussien, B., "Passive Range-Estimation for Rotorcraft Low-Altitude Flight," *NASA Technical Memorandum 103897*, NASA, Washington, DC, 20546-0001, October 1991.

[89] Sridhar, B., Suorsa, R., and Smith, P. N., "Vision Based Techniques for Rotorcraft Low Altitude Flight," *Proceedings of SPIE International Symposium on Intelligent Robotics*, Banglore, India, January 3–5, 1991, pp. 27–37.

[90] Sridhar,B., and Phatak, A.V., "Simulation and Analysis of Image-based Navigation System for Rotorcraft low altitude flight," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 2, March/April 1992, pp. 290–299.

[91] Sridhar, B., Suorsa, R., Smith, P. N., and Hussien, B., "Vision-Based Obstacle Detection for Rotorcraft Flight," *Journal of Robotic Systems*, Vo. 9, No. 6, 1992, pp. 709–727.

[92] Sridhar, B., Smith, P. N., Suorsa, R., and Hussien, B., "Multirate and Event Driven Kalman Filters for Helicopter Passive Ranging," *Proceedings of IEEE Conference on Control Applications*, Dayton, OH, September 13–16, 1992, pp. 800–805.

[93] Sridhar, B., Phatak, A., and Chatterji, G. B., "Scene Segmentation of Natural Images Using Texture Measures and Back–Propagation," NASA TM–104003, NASA, Washington, DC 20546–0001, March 1993.

[94] Sridhar, B., Suorsa, R., and Hussien, B., "Passive Range Estimation for Rotorcraft Low Altitude Flight," *Machine Vision and Applications*, Vol. 6, No. 1, 1993, pp. 10–24.

[95] Sridhar, B., and Chatterji, G. B., "Vision Based Obstacle Detection and Grouping for Helicopter Guidance," *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, August 9–11, 1993, Also appeared in the *AIAA Journal of Guidance Control and Dynamics*, Vol. 17, No. 5, September–October 1994, pp. 908–915.

[96] Stevens, Brian L., and Lewis, Frank L., **Aircraft Control and Simulation**, John Wiley & Sons, New York, NY, 1992.

[97] Tsai, R. Y., and Huang, T. S., "Uniqueness and Estimation of Three–Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, January 1984.

[98] Tsai, R. Y., "A Versatile Camera Calibration Technique for High–Accuracy 3D machine Vision Metrology Using Off–the–Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, Vol. RA–3, No. 4, August 1987, pp. 323–344.

[99] Unser, M., "Sum and Difference Histograms for Texture Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, January 1986, pp. 118–125.

[100] U.S. Department of Transportation and U.S. Department of Defense, **1990 Federal Radionavigation Plan**, DOT-VNTSC-RSPA-90-3/DOD-4650.4, The National Technical Information Service, Springfield, Virginia 22161.

[101] Vickers, A. L., and Modestino, J. W., "A Maximum Likelihood Approach to Texture Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.4, No. 1, January 1982, pp. 61–68.

[102] Weszka, J. S, Dyer, C. R., and Rosenfeld, A., "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 6, No. 4, April 1976, pp. 269–285.

[103] Xu, G. Y., and Fu, K. S., "Natural Scene Segmentation Based on Multiple Threshold and Textural Measurement," *Proceedings of IEEE Conference*, 1984, pp. 1111–1113.

[104] Zahn, C. T., "Graph–Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, Vol. C–20, No. 1, January 1971, pp. 68–86.

[105] Zienkiewicz, O. C., and Taylor, R. L., **The Finite Element Method**, Vol. 1, McGraw–Hill, London, UK, 1989.

Appendix A

Kalman Filtering Algorithm

The Kalman Filter development is based on a linear discrete time dynamical model of the form:

$$\mathbf{X}(k+1) = \mathbf{\Phi}(k)\mathbf{X}(k) + \mathbf{\Gamma}(k)\mathbf{U}(k) + \mathbf{\Gamma_d}(k)\zeta_{\mathbf{x}}(k) \tag{A.1}$$

$\mathbf{X}(k)$ is the state vector, $\mathbf{U}(k)$ is the control input vector, $\zeta_{\mathbf{x}}(k)$ is a vector of discrete time white noise sequences with covariance $\mathbf{Q}(k)$ representing the process noise, $\mathbf{\Phi}(k)$ is the state transition matrix [35], $\mathbf{\Gamma}(k)$ is the input distribution matrix and $\mathbf{\Gamma_d}(k)$ is the process noise distribution matrix. The Measurement vector equation is given by:

$$\mathbf{Z}(k) = \mathbf{h}(\mathbf{X}(k)) + \zeta_{\mathbf{z}}(k) \tag{A.2}$$

Here, $\mathbf{Z}(k)$ is the measurement vector, $\mathbf{h}(\mathbf{X}(k))$ is the vector of nonlinear measurement functions and $\zeta_{\mathbf{z}}(k)$ is measurement noise vector with covariance $\mathbf{R}(k)$. Note that $\zeta_{\mathbf{z}}(k)$ is assumed to be a vector of white noise sequences.

The Kalman Filter [3] is a computational algorithm for computing optimal state estimates $\hat{\mathbf{X}}(k)$ using the linear discrete time dynamical model and the measurement equations. The Kalman filter is optimal in the sense of generating unbiased minimum variance estimates. The filter continuously generates the state estimate error covariance matrix $\mathbf{P}(k)$. The Kalman Filter consists of two steps: measurement update, which improves the state estimate based on the new mea-

surements, and process update, which propagates the state estimate according to the dynamical equations. Before every measurement update step, an estimate of the state $\hat{\mathbf{X}}(k)$, state error covariance matrix $\mathbf{P}(k)$, process noise covariance matrix $\mathbf{Q}(k)$ and measurement noise covariance matrix $\mathbf{R}(k)$ are known. The new measurements are used for improving the state estimate and its error covariance as:

$$\tilde{\mathbf{X}}(k) = \hat{\mathbf{X}}(k) + \mathbf{K}(k)\left[\mathbf{Z}(k) - \mathbf{h}(\hat{\mathbf{X}}(k))\right] \quad \text{(A.3)}$$

$$\tilde{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]\mathbf{P}(k) \quad \text{(A.4)}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{K}(k)$ is the Kalman gain matrix computed using

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{P}(k)\mathbf{H}^T(k) + \mathbf{R}(k)]^{-1} \quad \text{(A.5)}$$

$\mathbf{H}(k)$ is the matrix of partial derivatives, representing the linear approximation to the nonlinear measurement functions, computed as:

$$\mathbf{H}(k) = \partial \mathbf{h}(\mathbf{X})/\partial \mathbf{X}|_{\mathbf{X}=\hat{\mathbf{X}}} \quad \text{(A.6)}$$

The process update part of the Kalman Filter accounts for system dynamics and propagates the state and its error covariance until the next measurement is obtained. The propagated values are:

$$\hat{\mathbf{X}}(k+1) = \mathbf{\Phi}(k)\tilde{\mathbf{X}}(k) + \mathbf{\Gamma}(k)\mathbf{U}(k) \quad \text{(A.7)}$$

$$\mathbf{P}(k+1) = \mathbf{\Phi}(k)\tilde{\mathbf{P}}(k)\mathbf{\Phi}^T(k) + \mathbf{\Gamma_d}(k)\mathbf{Q}(k){\mathbf{\Gamma_d}}^T(k) \quad \text{(A.8)}$$

The steps A.3 through A.8 form the core of the Kalman filter. This algorithm is summarized in Table A.1.

The sequence of steps given in Table A.1 assume that the measurement and process updates are carried out at the same rate. The extension to the case of measurement update time step being an integer multiple of the process update time step is straight forward [35]. The procedure is more complicated when the

Table A.1: Summary of Kalman Filter Algorithm

1. Set k = 1.
2. Initialize $\hat{\mathbf{X}}(k)$, $\mathbf{P}(k)$, $\mathbf{Q}(k)$ and $\mathbf{R}(k)$.
3. Compute $\mathbf{h}(\hat{\mathbf{X}}(\mathbf{k}))$.
4. Compute $\mathbf{H}(k)$ using Equation (A.6).
5. Compute Kalman gain $\mathbf{K}(k)$ using Equation (A.5).
6. Compute $\tilde{\mathbf{X}}(k)$ using Equation (A.3).
7. Compute $\tilde{\mathbf{P}}(k)$ using Equation (A.4).
8. Compute $\hat{\mathbf{X}}(k+1)$ using Equation (A.7).
9. Compute $\mathbf{P}(k+1)$ using Equation (A.8).
10. Increment k = k + 1.
11. Return to step 3.

measurement update is done asynchronously. In this case the measurement arrives within the process update time step. One of the ways of dealing with this situation is to split the process update into two steps: one from the previous process update time to the measurement epoch and the other from the measurement epoch to the next scheduled process update time. This ensures that the state estimate and its error covariance are available synchronously. Further details of multi–rate Kalman filter implementation can be found in Reference [3].

Appendix B

Matrices Using Aircraft Kinematic Models

The matrices required for estimating the position and velocity of the aircraft using Algorithms I, II, III and IV along with the Kalman filter in Appendix A are described in this Appendix. Algorithms I and II are described in Chapter 4 and Algorithms III and IV are described in Chapter 5. The outputs generated by these algorithms are used as measurements for the Kalman filter described in Appendix A.

The position and velocity estimation problem can be stated as follows. Given noisy measurements of the aircraft position, estimate its position components x_b, y_b, and z_b, and its inertial velocity components v_{bx}, v_{by}, and v_{bz}.

The discrete time state transition matrix with position and velocity components as states can be found to be:

$$\mathbf{\Phi}(k) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

This development assumes that the velocity states are integrated using the Euler integration formula. Here, Δt is the update time step. The control input vector is a 6×1 null vector. For convenience, the input distribution matrix $\mathbf{\Gamma}(k)$ can be chosen to be a 6×6 identity matrix. The process noise distribution matrix $\mathbf{\Gamma_d}(k)$ is a 6×6 identity matrix. The process noise covariance matrix $\mathbf{Q}(k)$ is a 6×6 diagonal matrix. The diagonal elements of $\mathbf{Q}(k)$ are chosen as "tuning parameters" for the Kalman filter [35].

The 3×1 measurement vector $\mathbf{Z}(k)$ consists of the components of the aircraft inertial position vector, x_b, y_b and z_b, determined using Algorithms I, II, III and IV. With these measurements, position and velocity components as states, the measurement model matrix is:

$$\mathbf{H}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{B.2}$$

Since three measurements are used, the dimension of the measurement noise covariance matrix $\mathbf{R}(k)$ is 3×3.

The 6×1 state vector $\hat{\mathbf{X}}(k)$ and its 6×6 error covariance matrix $\mathbf{P}(k)$ have to be initialized to begin state estimation process. The Kalman filtering algorithm described in Appendix A can then be used for recursive state estimation.