Chapter 2

Literature Review

Proper design of any robotic system requires knowledge of available technology. Design of any robotic system should pay close consideration to existing mechanical architectures, sensors, and sensor fusion and navigation strategy options. Knowledge of the advantages and shortcomings of existing systems will result in the selection of components and architectures best suited to any given task. This chapter presents an overview of existing mechanical architectures, sensors, and sensor fusion and navigation strategies.

2.1 Mobile Robot Mechanical Architectures

The purpose of this section is to provide an overview of common vehicle architectures used in mobile robot applications. A focus is placed on the mechanical and control issues associated with each design. The three methods of locomotion that have been considered are wheels, tracks and legs. Each of these approaches will be reviewed in the following sections.

2.1.1 Mobility and Maneuverability

A vehicle design can be categorized upon its mobility and maneuverability. Thus, it is important to define and differentiate between vehicle mobility and maneuverability. In the strictest sense, mobility is the number of possible movements, or degrees of freedom, that an object may have with respect to a given coordinate frame (McKerrow, 1991). A free-floating object is capable of six degrees of freedom: three translational and three rotational. Typically a ground vehicle is limited to three degrees of freedom: two translational and one rotational. Bekker's (1969) interpretation of mobility also considers a vehicle's ability to handle varying terrain and soil. Thus, even though a vehicle may have three degrees of freedom, any or all of these degrees may be limited by the vehicle's inability to deal with rough terrain or soft soil. Power, size, and type of locomotion all effect a vehicle's mobility.

The maneuverability of a vehicle is closely linked to its design. A vehicle's maneuverability is characterized by its ability to perform movements that combine multiple degrees of freedom. A standard car is capable of three degrees of freedom, however; in order to rotate the vehicle must also translate. This configuration limits the flexibility of the vehicle path due to coupled degrees of freedom (Critchlow, 1985). This also introduces the concept of holonomic versus nonholonomic systems. The state of holonomic systems, such as many serial manipulators, may be described using a set of displacement variables, or a set of velocity variables that may be integrated as a whole to produce displacement variables. Wheeled vehicles, however, fall into the realm of nonholonomic systems. Nonholonomic systems can not be described by the integration of its individual actuator velocities (Angeles, 1997).

Legged, omni-directional, and differential drive designs decouple their multiple degrees of freedom. The ability to rotate without translation makes such designs more maneuverable. Given two vehicles of equal mobility but differing maneuverability, the more maneuverable vehicle will typically be advantageous in cluttered environments.

2.1.2 Vehicle Stability

The ability of a vehicle to maintain stability is an important consideration during design selection. A vehicle's conceptual design can be extremely maneuverable but remain infeasible due to low stability. The basis of static stability lies in a vehicle's *support pattern*. A vehicle's support pattern can be described by a convex polygon projected onto the horizontal plane. The convex polygon is the shortest bounding polygon that contains the vertical projections of all of a vehicle's support structures such as wheels or feet (Messuri and Klein, 1985; McKerrow, 1991). Examples are shown in Figure 2.1. Note that a support structure may not necessarily be included in the determination of the support pattern. This could be especially true in the case of legged vehicles with legs momentarily lifted off the terrain. The stability of vehicle designs may be compared using each vehicle's *static stability margin*. The magnitude of the static stability margin is the shortest distant between the vertical projection of the center of gravity and any side of the support pattern (Messuri and Klein, 1985).

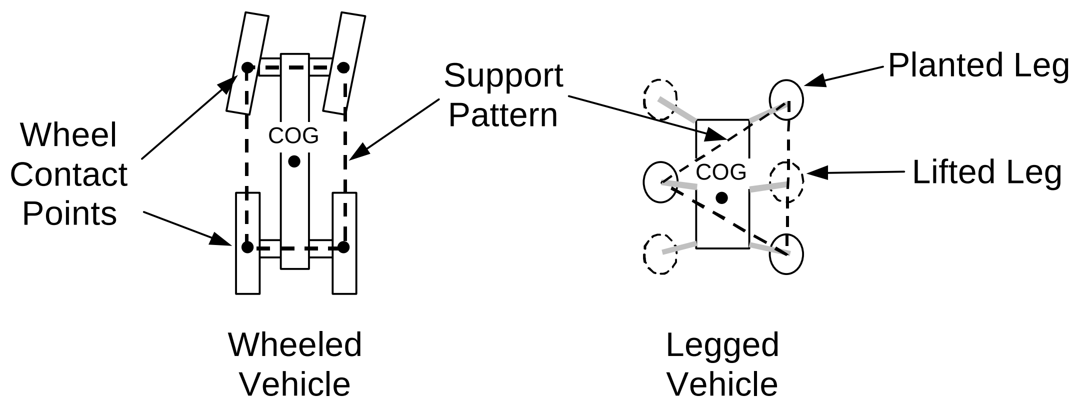

Figure 2.1 Vehicle Support Patterns

The static stability margin is obviously dependent upon vehicle configuration and size. This approach also only considers vehicle's operating over smooth terrain. Operation in rough environments requires the use of more complex approaches such as the *energy stability margin*. This technique considers the minimum work required rotate a vehicle about its support pattern for a given orientation on uneven terrain (Messuri and Klein, 1985). This method accounts for the effects of vehicle height on stability. Thus, the energy stability margin indicates the chance of vehicle tipping or rollover.

Designers should be careful to consider the effects of vehicle stability on performance. This is especially true for walking vehicles where stability levels constantly change as the vehicle moves. Vehicle acceleration and deceleration may also cause radical changes in vehicle momentum that could lead to rollover. This must be considered in the vehicle's motion control system.

2.1.3 Wheeled Vehicles

Wheeled vehicles make up the largest group of mobile robots. Wheeled vehicles are typically simple to control, use a minimum of energy per unit distance movement, and can usually move faster than tracked or legged vehicles (McKerrow, 1991). Although the maneuverability of wheeled vehicles is highly dependent on physical design, vehicles can be loosely grouped as steer-drive or differential drive (Liu and Lewis, 1994). Steer-drive vehicles control steering and propulsion using individual actuators. Steering and propulsion may be performed by the same wheels (as in a front-wheel drive car) but speed and steering angle are controlled separately. Differential drive vehicles utilize the same wheels and actuators for drive and steering. Changes in direction are produced through proportional control of individual wheel velocities.

Wheeled vehicles, especially those used in outdoor environments, suffer from certain problems. The terrain a wheeled vehicle can negotiate is dependent on tire size. Wheeled vehicles cannot typically traverse obstacles larger than approximately one-half their tire diameter (McKerrow, 1991). Steep grades and trenches can also limit mobility due to loss of stability. The use of pneumatic tires, the most common type of wheel in outdoor applications, also creates problems. Manufacturing, inflation, and compression differences as well as tire misalignment and imbalance lead to errors in position and speed tracking of wheeled vehicles (Borenstein, 1995).

The next five sections will discuss various drive and steering configurations of wheeled vehicles. Configurations are based upon the nominal number of wheels used for drive and steering.

2.1.3.1 Three Wheels

Three wheeled vehicles are typically configured in one of four possible ways. The first three of these are shown in Figure 2.2. The first arrangement utilizes an actuated steering wheel with two drive wheels. Note that, in this arrangement, three wheel motions are being used to control only two independent degrees of freedom. This “tricycle” arrangement requires that the two drive wheels rotate at different speeds while the vehicle is turning (Borenstein, 1995). If the drive wheels are statically coupled to rotate at the same speed, “scrubbing” will occur. “Scrubbing” is the dragging or slippage of a wheel caused by a difference in required and actual wheel speed. “Scrubbing” leads to excessive tire wear and odometry errors. Using the second three-wheel arrangement can eliminate the problems of scrubbing. Using a single wheel for both drive and steering allows the two remaining wheels to rotate independently. This variant results in a more complex mechanical design for the single drive-steer wheel, which may lead to steering errors during implementation (McKerrow, 1991).

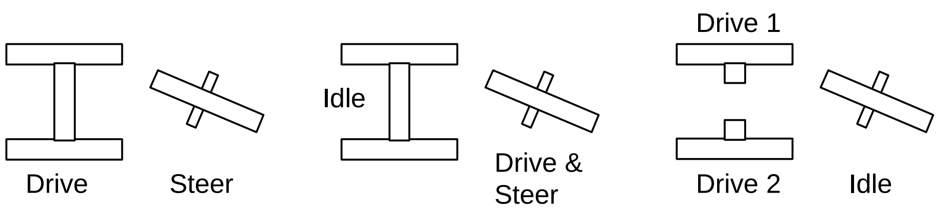

Figure 2.2: Three Wheel Vehicle Configurations (McKerrow, 1991)

The third possibility is the differential drive arrangement. Two independent drive wheels are used to control both speed and steering angle. The remaining wheel or caster is used only to maintain stability. Running the two drive wheels at equal speeds will result in straight motion. Changes in vehicle direction are achieved by operating the drive wheels at different speeds. Equal but opposite wheel velocities will cause the vehicle to rotate about the mid-point of its drive wheels, thus producing the possibility of a zero-turn-radius (ZTR) vehicle. The extreme maneuverability of ZTR vehicles has made this arrangement popular among researchers (Liu and Lewis, 1994). Precise motor control and feedback are necessary to accurately control ZTR vehicles. Mechanical factors such as wheel slip, misalignment, and unequal tire diameter can also complicate trajectory tracking (McKerrow, 1991).

A final variation of the three-wheel design is called synchronous drive. Three wheels placed 120° apart are linked via belts or gears to single drive and single steering motors. All three wheels are steered in the same direction at equal speed. While this design does provide good maneuverability, it is complicated to control. Control over the orientation of the vehicle body is also lost unless a separate "turret" motor is added (Borenstein, 1995).

All the three wheel designs have the advantage that all wheels inherently remain in contact with the ground. This fact eliminates the need for additional wheel suspension (McKerrow, 1991).

2.1.3.2 Four Wheels

Four-wheel vehicles are perhaps the most common arrangement. Three possible variations of four-wheel vehicles are shown in Figure 2.3. Four wheeled vehicles can provide additional traction and stability over three wheel vehicles, especially four-wheel-drive and all-wheel-drive configurations.

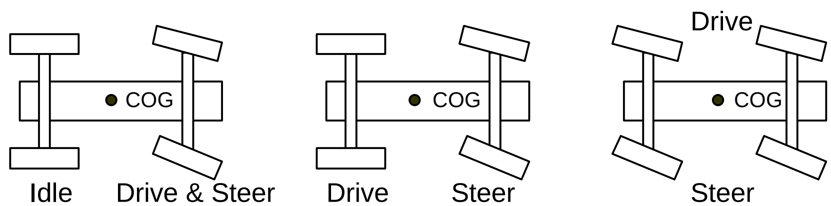

Figure 2.3: Four Wheel Vehicle Configurations (McKerrow, 1991)

Four-wheel vehicles have limited maneuverability, since they must move forward to turn. Wheelbase and length then limit the vehicle's turning radius. Four wheel vehicles are also more prone to "scrubbing" than three wheel vehicles, since both the drive and steering wheel sets must operate at different velocities as the vehicle turns (McKerrow, 1991). Suspension must be added to four wheel vehicles to achieve contact and stability on uneven terrain. Virginia Tech's Herbie vehicle, shown in Figure 2.4, demonstrates these concepts. Herbie's 1.5HP motor provides ample power to the rear wheel drive assembly, but its maneuverability is limited by size and turning radius.

Figure 2.4: Virginia Tech's Herbie

2.1.3.3 Six Wheels

Six wheeled vehicles have been used in a number of applications. The most prevalent in recent literature are Canrnegie-Mellon's Terregator, shown in Figure 2.5, and NASA JPL's Mars Rover series. Terregator uses six powered wheels and skid-steers similar to a tank. This arrangement gives Terregator enough power to climb stairs but makes its steering indeterminate and thus difficult to accurately control (McKerrow, 1991; Thorpe, 1990). This trade-off of accuracy for traction does have certain redeeming values in outdoor applications. The six-wheel design widely distributes the vehicle's weight, helping to avoid sinking on softer terrain. The possibility of six powered wheels also increases potential payload and towing ability.

Figure 2.5: The Carnegie-Mellon Terregator (Thorpe, 1990)

NASA JPL's Mars Rover series of robots have adopted a six-wheel arrangement for maximum terrain handling. The Rocky 7 Rover, shown in Figure 2.6, uses six independently powered, 13 cm diameter wheels and a unique "Rocker-Bogie" suspension system which allows it to climb rocks 1.5 times the diameter of its wheels (Hayati et al., 1997). The rover series changes direction using two or four steerable wheels, depending on the model. Mobility is achieved at the expense of eight to ten separate motors. Six motors are used to individually power each wheel. Depending on the model, two or four more motors are used to handle steering. This design greatly increases mechanical and control complexity.

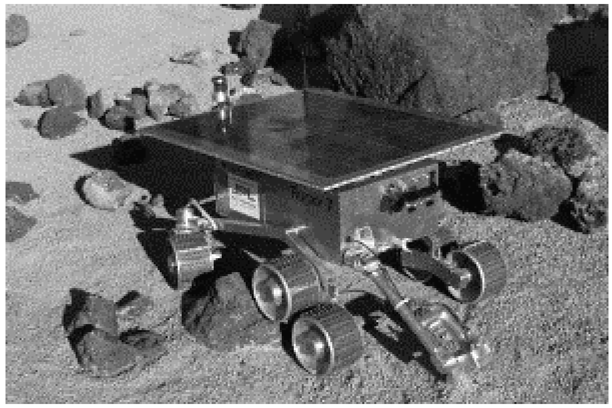

Figure 2.6: Rocky 7 Mars Rover
(http://mpfwww.arc.nasa.gov/tasks/scirover/homepage.html)

The additional complexity of using six wheels is probably only justified in extremely rugged terrain where traction and obstacle traversing are necessary for adequate mobility.

2.1.3.4 Omni-Directional Wheels

The desire for improved maneuverability has led to the development of several omni-directional wheel assemblies such as the Mekanum wheel shown in Figure 2.7. The Mekanum wheel and its counterparts, such as the Standford, Sweedish, and Illanator wheels, are standard wheels with the addition of spherical or cylindrical rollers attached at an angle to the wheel's axis. The combination of wheel and roller movement results in high maneuverability, including zero-turn-radius capabilities (Angeles, 1997; McKerrow, 1991). Precise control of omni-directional wheels is quite complex and they are best suited to only smooth, regular surfaces (Borenstein, 1995). This constraint has largely limited their use to research applications.

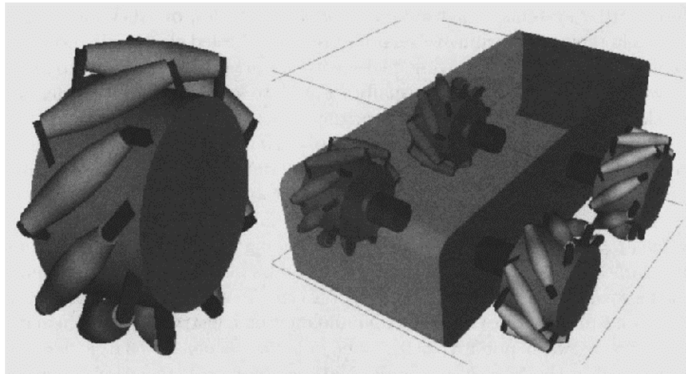

Figure 2.7: The Mekanum Wheel (Angeles, 1997)

2.1.3.5 Multi-Degree-of-Freedom Vehicles

Multi-Degree-of-Freedom vehicles (MDOF) can take on any number of standard wheel arrangements but achieve the maneuverability of omni-directional vehicles. A commercial example, the OmniMate, is shown in Figure 2.8. This MDOF vehicle developed by Johann Borenstein at the University of Michigan uses two differential "trucks" that are controlled independently. The vehicle may move sideways by rotating both trucks 90° and then moving forward. A compliant linkage between the two trucks absorbs momentary control errors and allows the vehicle to maintain an instantaneous center of rotation (ICR) during turns (Borenstein, 1997). The vehicle is capable of tight turns with minimal error and provides a large upper payload area. This vehicle is designed for indoor use, but it could be redesigned for outdoor applications.

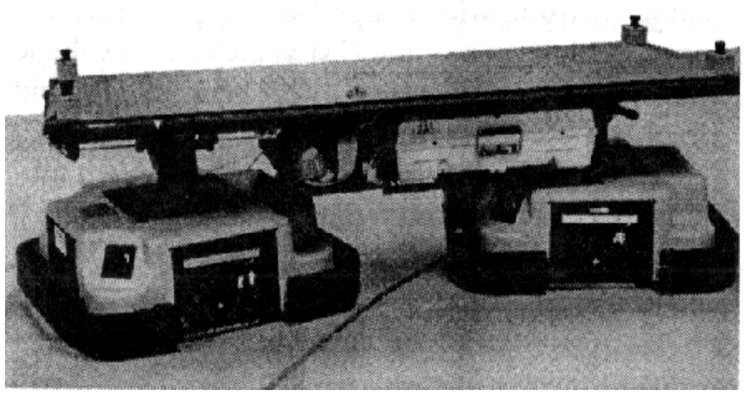

Figure 2.8: The OmniMate MDOF Mobile Robot (Borenstein, 1997)

2.1.4 Tracked Vehicles

Tracked vehicles are well suited to extremely rough terrain. Tracks are typically used in situations requiring large towing or payload abilities. Tracked vehicles are also capable of negotiating trenches, steep inclines, and even stairs. Based on overall weight and dimensions, tracked vehicles also tend to have high ground contact ratios compared to wheeled vehicles. This helps reduce sinking in soft soil and produces higher thrust with a lower slip ratio than comparable wheeled vehicles (Wong, 1978).

The tracked approach has several drawbacks however. Tracked vehicles are maneuvered using “skid-steering,” which is driving the tracks at individual speeds to produce a desired turning radius. Skid steering causes high slippage and makes position tracking (odometry) extremely difficult. Steering is also dependent upon the soil density due to vehicle sinkage. Accurate steering becomes even more difficult at high speeds due to side slippage and uneven track resistance caused by centrifugal forces (Wong, 1978). Fan et al. (1995) proposes a simple solution to tracked vehicle odometry. An “encoder trailer” pulled behind the parent vehicle is used to record odometry and orientation information. This data can be used to find the tracked vehicle’s global position and orientation relative to its trailer. Tracked vehicles also tend to produce environmental damaging caused by skid steering (Fan et al., 1995, McKerrow, 1991). Tracked vehicles are most frequently used in teleoperated applications such as search and rescue, bomb removal, and the nuclear industry.

2.1.5 Legged Vehicles

In depth studies of “legged” walking vehicles began during the mid-fifties. Legged vehicles represent designers’ best attempts at reproducing the motion of biological systems. Legged vehicles provide several advantages over wheeled vehicles. Stulce et al. (1990) point out that “about 50% of the earth’s surface and rugged terrain such as the surface of Mars are inaccessible to wheeled or tracked vehicles.” Legged vehicles are capable of stepping over obstacles and trenches, ascending and descending stairs and other steep inclines, and can provide smooth travel by separating body and leg motion. These characteristics make legged vehicles ideally suited to rough and unpredictable terrain (Venkataraman, 1996; McKerrow, 1991; Raibert, 1986).

Legged vehicles may be classified by static and dynamic stability. Static stability requires a walking system to have a minimum of three feet on the ground with the center of gravity located the convex polygon formed by the legs. Dynamic stability is dependent upon adaptive movement of the vehicle's legs or body to maintain the center of gravity within the contact area of the feet and ground (McKerrow, 1991). Legged vehicles must also adapt to reactions and inertial forces to maintain dynamic stability. Thus, legged vehicles are typically designed with a minimum of three legs for stable operation. A "hexapod", or six legged, vehicle is often preferred for stability and terrain adaptability (Meystel, 1991). A typical example is the TU Munich Hexapod shown in Figure 2.9.

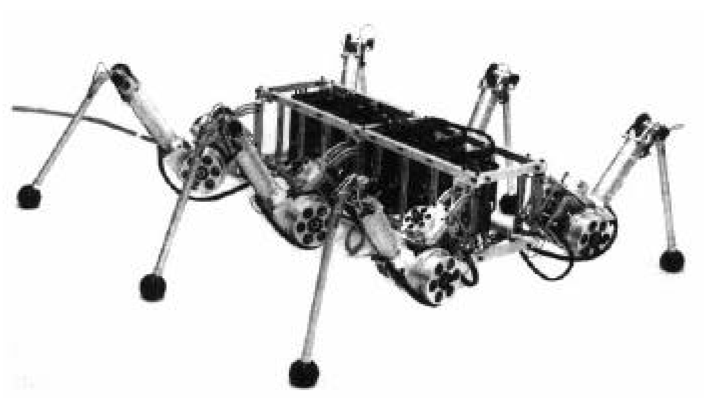

Figure 2.9: TU Munich Hexapod (Angeles, 1997)

An exception to this "six-leg" rule is the series of hopping vehicles designed by Marc Raibert at Carnegie-Mellon University. The robot shown in Figure 2.10 is capable of hoping motion in three dimensions. Dynamic stability is accomplished using active balance principles and complex control algorithms. Raibert's hopping machines have achieved speeds up to almost 5 mph (Raibert, 1986). The principles learned through these hopping machines have been extended to running biped and quadruped machines.

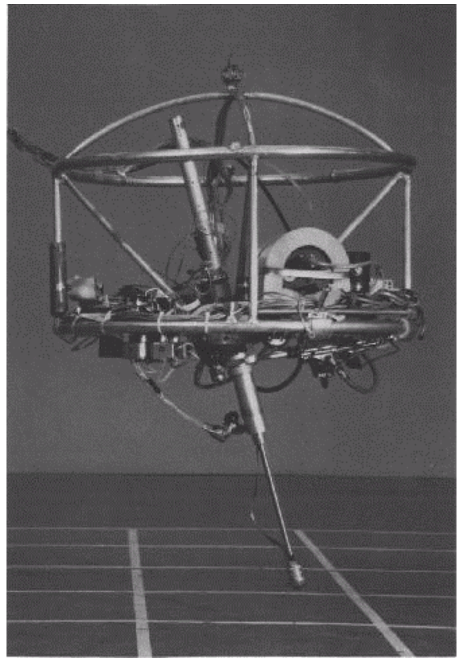

Figure 2.10: Hopping Robot (Raibert, 1986)

Control of any legged vehicle requires the ability to accurately repeat leg motion. The cycle of leg motion frequency, phase, and amplitude is controlled by a gait system, and complex control algorithms and feedback systems are necessary for stable operation over long time periods (Venkataraman, 1991). To be worthwhile, legged vehicles must also be able to adapt leg length and suspension to deal with extremely rough terrain. To negotiate rough terrain, legged vehicles must be able to sense terrain and detect and plan foothold sequences to maintain stability (Raibert, 1986). Adaptive control algorithms needed for these systems may limit real time vehicle operation (Angeles, 1997).

Legged vehicles have several physical shortcomings. Legged vehicles are more likely to sink and become immobile than wheeled vehicles of comparable size. Legged vehicles may also suffer from weak friction contact on hard surfaces (Bekker, 1969). Sinking and slippage during motion leads to poor position tracking. Legged vehicles are especially limited in autonomous applications. Since all components must be carried on board, space in autonomous systems is at a premium. Legged vehicles require constant power to be supplied to joint motors to maintain static and dynamic stability (McKerrow, 1991). Exceptions to this rule are passive-legged robots. The use of actuated, multi-body designs allows motion to occur between segments of the vehicle body. This is akin to the movement of a caterpillar (Stulce et al., 1990). Since the vehicle legs are passive, static stability can be achieved using a minimum of energy. Typically though achieving a balance between onboard power, run time, and weight in legged designs is difficult.

2.1.6 Mechanical Architecture Summary

The selection of a mechanical architecture for a mobile robot is highly dependent upon the application. Indoor applications often require a highly mobile robot to work in cluttered environments. Indoor surfaces tend to be smooth and regular without steep inclines. Outdoor environments, however; call for a more robust platform capable of handling rough terrain. Many outdoor applications, such as smart highway systems, do not call for high degrees of maneuverability. Applications such as planetary exploration require robots capable of obstacle avoidance and climbing ability. Multi-degree-of-freedom and differential drive configurations can provide this extreme maneuverability.

2.2 Sensors

Sensors act as a mobile robot's window to the world. Sensors provide information about the surrounding environment as well as the state of the vehicle. A wide variety of sensors of varying purpose, resolution, accuracy, reliability, and compatibility are available to autonomous vehicle developers. The ability to combine multiple sensors of varying characteristics into viable system falls into the realm of sensor fusion. Issues associated with sensor fusion will be covered in Section 2.2.4. The goal of the current section is to provide a brief overview of some of the most common sensors and their characteristics. In-depth discussions of these and numerous other sensors may be found throughout current literature (Larkin, 1996; Denes et al., 1995). Sensors that rely on predetermined landmarks or beacons will not be discussed.

2.2.1 Sensor Classification

Sensors can be grouped according to several specific criteria such as the type of data measured, accuracy, and resolution. A more general classification system labels sensors as either active or passive (Larkin, 1996). Active sensors determine the state of the surrounding environment by emitting and receiving a signal. Passive sensors do not emit signals but collect data through observation. Passive sensors may be further classified as internal or external (Fu, 1987). Internal sensors collect data concerning a vehicle's state such as motor velocity or steering angle. External sensors are used to determine the state of the vehicle's environment such as range or proximity.

2.2.2 Active Sensors

2.2.2.1 Laser Range Finders

Laser range finders are rapidly gaining favor as sensors for obstacle detection and terrain mapping in autonomous applications. A typical scanning laser range finder is shown in Figure 2.11. Laser range finders operate under two basic principles, triangulation and time-of-flight or phase shift. Triangulation utilizes a laser source and detector. The emission angle of the laser source and its distance to the detector allows the range to a detected object to be determined through simple geometry (Fu et al., 1987). The time-of-flight method may be achieved in two ways. First, the laser source may be used to create a pulse of light. The time required for the pulse to return coaxially may be used to calculate range. A second alternative is to use a continuous beam. A beam splitter is used to redirect the coaxially returning beam into a detector. The phase shift between the outgoing and incoming beams is used to determine range. The high frequency of laser light makes phase shift determination difficult. A common solution to this problem is modulation of the laser light signal. Modulation can extend a sine wave frequency of 10 MHz to a wavelength of 30 meters, thereby allowing more accurate phase shift determination (Fu et al., 1987).

Figure 2.11: Scanning Laser Range Finder

Laser range finders provide fast, high-resolution readings over a long measurement range. Lasers' insensitivity to illumination conditions also eliminates a majority of noise and false readings associated with other active sensors. Laser range finders also have the capability to produce 2-D and 3-D contours of surrounding terrain (Liu and Lewis, 1994; Kelly et al., 1992; Fu et al., 1987).

Laser range finders do have several limitations. Lasers are among the most expensive sensor system available. Continuous beam systems may also require high power requirements and can create hazardous conditions for human operators (Salem and El-Khamy, 1995). Uncertainties in measurements require averaging of several readings. Stentz et al. (1992) list several other limitations of laser range finders in outdoor settings. Sensor limitations typically cause lasers to yield little or no return signal when detecting specular surfaces. Angular resolution of particular systems may also cause the edges of objects to become ambiguous at long distances. Rough terrain can cause uneven sampling and create “shadows”, areas block by objects or grades. Vehicle speed can also lead to data misalignment and under sampling depending on a system’s scan time.

2.2.2.2 Radar

Radar is a popular sensor that is being implemented in smart highway and driver assistance research. Radar systems emit a microwave or millimeter-wave signal that is reflected off objects in the sensor's field of view. The time of flight for the returning signal may then be used to determine object distance as well as speed (Hennessey, et al., 1995). Radar is more effective than most time-of-flight sensors, since it is effective over both short and long ranges, depending on the signal power level, and is capable of penetrating fog and rain. Radar provides better accuracy, resolution, and noise performance than ultrasonic sensors and is not susceptible to specular reflection from metallic surfaces like laser range finders. Scanning radar systems are also capable of returning 3-D terrain and object information (Salem and El-Khamy, 1995; Denes et al., 1995).

The main disadvantages of radar are spatial resolution and cost. Single, non-scanning radar systems can have trouble detecting multiple objects, especially as range increases. Detection of multiple objects may produce "cluttered" or "blurred" data caused by multiple return-signal averaging (Denes et al., 1995). The cost of radar also makes its use somewhat prohibitive, although recent developments in microwave integrated circuitry has brought the cost of systems down to the $300 - $500 price range (Ganci et al., 1995).

Radar is effective for obstacle avoidance, path planning, mapping, and object identification (Salem and El-Khamy, 1995). Ganci et al. (1995) have developed a forward looking automotive radar system to detect distance to other vehicle and their closing velocity. Hennessey et al. (1995) developed a "virtual bumper" system using a series of smaller radar units. The system creates a "personal space" about the vehicle that, when invaded by other objects, causes the vehicle to alter its course away from the invading object. This approach could supplement or replace current tactile sensor arrays.

2.2.2.3 Structured (Contrived) Lighting

Structured, or contrived, lighting is another method for range determination. A specific, regular light pattern is projected onto the area of interest. A typical setup is shown in Figure 2.12. A camera is used to record the image and distortion in the light pattern is used to calculate range and depth information (Critchlow, 1985; Fu et al., 1987). Lighting patterns can consist of single dots and lines or a regular grid depending on the application. Various light sources may be uses to create the structured pattern depending on the desired accuracy and lighting conditions present. Scanning lasers are commonly used to produce intense, high accuracy patterns. Less costly solutions involve common light sources, such as spotlights, filtered through a pattern template (Critchlow, 1985).

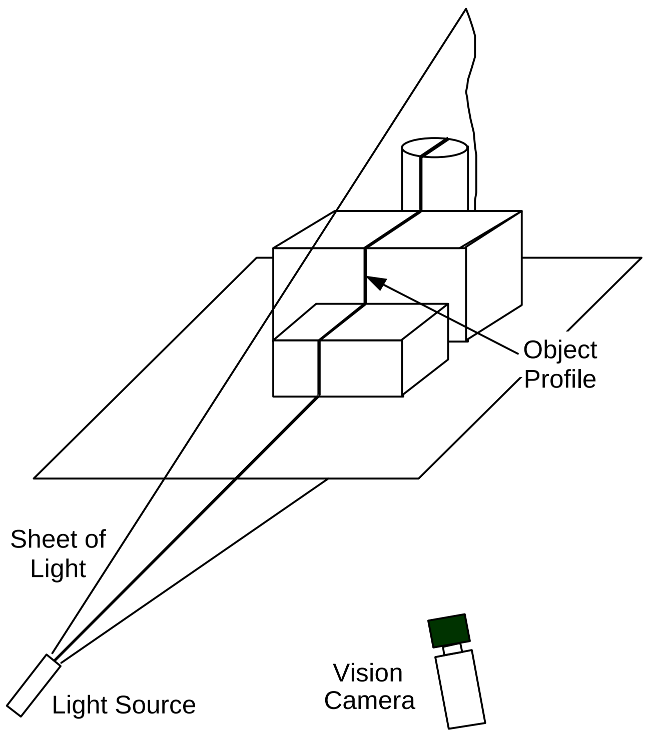

Figure 2.12: Basic Structured Light Setup

Structured light systems vary widely in price and accuracy. The combination of light source and secondary camera also requires accurate system calibration, thus adding another possible source of error. Structured lighting systems are dependent on environmental illumination and tend to lose accuracy as distance increases (Fu et al., 1987; Critchlow, 1985).

2.2.2.4 Ultrasonics

Ultrasonics, also known as sonar, operate on the same principles that allow bats and dolphins to navigate and locate prey. A common ultrasonic assembly is shown in Figure 2.13. A piezo-electric transducer is used to periodically produce an acoustic pulse, or “chirp”. The pulse is reflected off of an object within the surrounding environment and received by the same transducer. The speed of sound in the surrounding medium, and wavelength are then used to determine the pulse’s time-of-flight and the associated object’s distance (Fu, 1987). The acoustic pulse is typically sent at several frequencies to avoid errors produced by objects at a distance that is an exact multiple a certain pulse’s wavelength (Critchlow, 1985). Effective detection ranges from 0.9 to 35 feet. Ultrasonics are an efficient method of obstacle detection due to a “comparatively large beam aperture” (Lim and Cho, 1993). Ultrasonics, like most active sensors, are able to deliver range data directly to the sensor fusion system. This decreases subsequent processing time. The most attractive feature of ultrasonics is their low cost (Liu and Lewis, 1994).

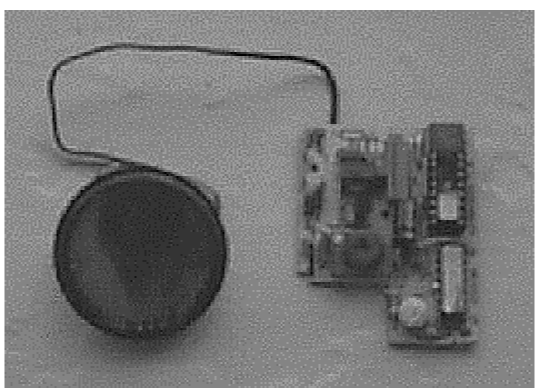

Figure 2.13: Ultrasonic Range Finder

Ultrasonics have numerous problems associated with their operation. Low resolution, precision, and signal noise are common to ultrasonics. These problems may be attributed to variations in the signal speed propagation, inaccuracies in the timing circuit, and uncertainties in the time of arrival of the returning signal (Liu and Lewis, 1994; Kam et al., 1997). Sonar’s large signal beam makes object location uncertain within the beam. Kam et al. (1997) presents a “Tri-Aural Sensor Array” to solve this problem that uses three ultrasonics placed in-line and evenly spaced. The center sensor transmits and receives while the outer sensors receive only. Object location is achieved using signal triangulation.

Two other major drawbacks of ultrasonics are “specular reflection” and “crosstalk”. At angles approaching 60°, smooth, mirror-like surfaces cause the signal pulse to be reflected away from the transducer and object detection fails (Lim and Cho, 1993). To avoid this “specular reflection” situation and gain a full 360° view of the environment, many systems will employ 20 to 30 transducers evenly placed about a vehicle. This can quickly increase the cost of a previously low-cost sonar system (Liu and Lewis, 1994; Borenstein and Koren, 1995). Crosstalk occurs when an ultrasonic detects the signal produced by another transducer in the same array or on another vehicle. Borenstein and Koren (1995) has devised a Error Eliminating Rapid Ultrasonic Firing system (EERUF) that employs pulse timing and delay to characterize the signal produced by each transducer. This system eliminates the problems caused by both crosstalk and environmental noise.

2.2.3 Passive Sensors

2.2.3.1 Potentiometers

Potentiometers may be the most basic of all passive sensors. Potentiometers are mechanically variable resistors. Linear and rotary models are available and are commonly used in robotics to determine linear and angular displacement. Resolution is highly dependent upon cost, manufacturer, and, in the case of rotary models, the number of turns between end stops. Various models range in price from a few cents to about $60 and can have errors less than 0.1%. Potentiometers still suffer from limited life due to physical wiper contact (Crtichlow, 1985; Gerry, 1988).

Potentiometers are commonly used to determine motor shaft position. Angular position can be tracked as a function of potentiometer resistance or voltage output. When applied to a steering actuator this provides a simple way of sensing a vehicle's steering angle.

2.2.3.2 Tachometers

Tachometers are used to measure rotational shaft speed. Their operation is similar to a small DC generator or DC motor driven in reverse (i.e., power supplied to the motor shaft). A multiple wound armature is rotated inside a permanent magnet stator to produce an output voltage. The output voltage is proportional to the speed of the shaft that the tachometer is coupled to. While tachometers are inexpensive, their output is noisy and they have limited life due to commutator brush wear (Gerry, 1988). Still, tachometers provide a relatively simple means of determining vehicle velocity and acceleration based upon motor shaft speed.

2.2.3.3 Resolvers and Synchros

Resolvers and synchros are also similar to motors in construction, containing a single rotor winding and 3 or 4 stator windings. AC power is passed through the rotor as it is revolved and the sine wave produced at the output can be used to determine shaft position and velocity (Critchlow, 1985). Resolvers and synchros can produce extremely accurate output for a reasonable cost. A 16-bit synchro can have a resolution of 0.00549 degrees (McKerrow, 1991). The disadvantage of this type of resolvers and synchros is that they require an analog-to-digital converter. Resolvers have the ability to determine rotational position and velocity. Thus, they may be used for both vehicle odometry and velocity tracking.

2.2.3.4 Encoders

Encoders are used to determine the angular or linear position. Optical encoders, the most common type, use an etched-substrate radial disc with concentric rings as represented in Figure 2.14. A photo-emitter and sensor placed on either side of the disc produce a binary output as the etched areas of the disc revolve between them. Differentiation of the binary output may be used to determine disc speed and acceleration.

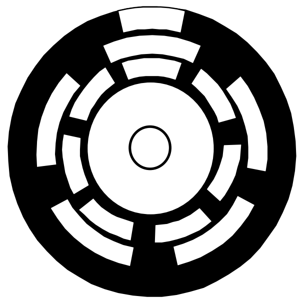

Figure 2.14: Two-Track Incremental Encoder Disc

Encoders may be either absolute or incremental. Absolute encoders are typically of higher quality and require only one ring to determine position. Multiple rings may be used to increase resolution however. With each concentric ring representing a single bit, a 20-ring encoder can measure angular position up to one part in 2^{20}, or 1,048,576. Incremental encoders require multiple rings to determine position and direction. In Figure 2.14 the outer ring is used to count disc rotations. The two inner rings are used to determine rotation direction based on the staggered binary pulses from the two inner rings' separate photosensors (Critchlow, 1985).

Numerous types of encoders exist (Gerry, 1988). Optical encoders, described above, have a long life but are sensitive to shock. Brush encoders replace the etched-substrate disc with a segmented metal disc and brushes. This type of encoder is relatively cheap but tends to have poor resolution and short life due to physical brush contact. Magnetic encoders utilize a segmented metal disc as well. The disc is passed between a set of permanent magnets to form a Hall-Effect sensor. The change in magnetic field is used to produce the necessary binary output. Magnetic encoders have a reasonable life and are not as susceptible to shock as optical encoders. Resolution of magnetic encoders is approaching that of optical encoders.

Encoders are very similar to resolvers and synchros in application. The binary data stream may be used to determine position and velocity. Encoder resolution allows for precise vehicle control.

2.2.3.5 Tactile Sensors

Tactile sensors provide robots with a sense of “touch”. The most common tactile sensors consist of “whiskers” attached to various types of contact switches. Robot “skins” are also available which are constructed of alternating layers of conducting strips and insulating material. Incorporating strain gauges into such arrays can even given the system the ability to sense applied forces. Tactile sensors are cheap and easy to implement. However, tactile sensors are prone to noise and have a limited range, making them impractical for mobile robots operating at high speeds (Critchlow, 1985; McKerrow, 1991;Gerry, 1998). Experience has also shown that improperly designed tactile sensors can give false signals due to triggering by inertial forces. Tactile sensors are still practical as a “last line of defense” when obstacles have passed the envelope of computer vision and range finding detection.

2.2.3.6 Cameras

Cameras provide perhaps the highest potential for rich data return. Cameras are typically one component of a larger computer vision system. A typical vision system, shown in Figure 2.15, consists of a camera, a vision interface processor, or “frame-grabber”, a computer system to process and interpret data, and possibly a supplemental video monitor (Critchlow, 1985). Vision systems can be used for boundary detection, object identification and tracking, ranging, and navigation. In autonomous vehicle applications, the derived data is used to produce actuator commands.

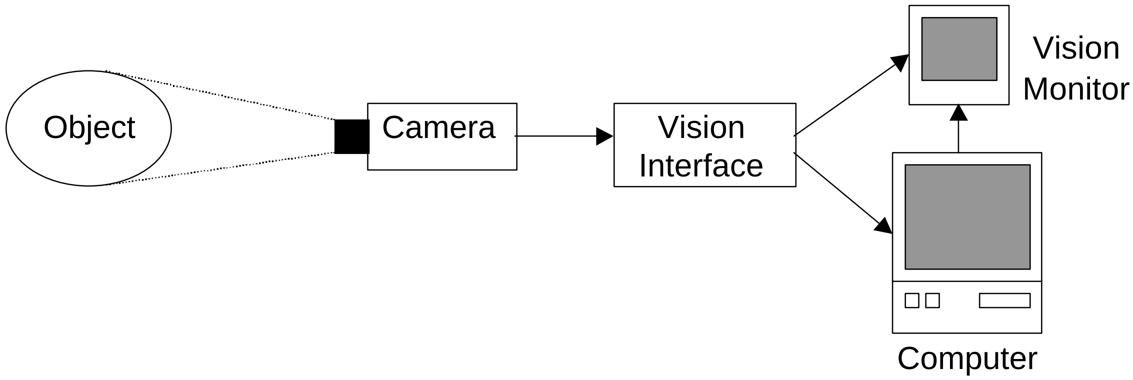

Figure 2.15: Vision System Components

Cameras operate by collecting reflected light from objects and focusing it onto a light sensitive electrical component. Charge-Coupled Device (CCD) cameras are common in computer vision systems. A typical CCD camera is shown in Figure 2.16. CCDs are popular due to their high sensitivity to a wide spectral range, low power requirements, high resolution, fast data throughput, and low noise (Gerry,1998). Typical camera resolution is 512x512 pixels, although some newer CCD units are capable of producing up to 4 million pixels per frame (Denes et al., 1995). CCD units are available in Gray Scale and Color versions. Once a camera collects a visual image it must be processed to be useful to the system. The remainder of the vision system accomplishes image processing and interpretation.

Figure 2.16: CCD Camera

2.2.3.6.1 Computer Vision

Computer vision may be broken into numerous sub-processes shown in Figure 2.17 (Fu et al., 1987). Sensing is the image capturing process performed by the camera. The acquired image is then preprocessed to remove noise and enhance certain image qualities. Typical processes include blurring, color extraction, thresholding, and image plane adjustment. When preprocessing is complete, the image is segmented to isolate specific objects or regions such as lane boundaries and obstacles. The segmented objects are then analyzed to determine features such as color, size, and range. The extracted features are used to recognize specific objects and assign them an identity. This is useful in advanced highway applications to classify different road signs and vehicle types (Tsinas, 1995). The most challenging task for computer vision is interpretation. After all the objects of interest have been identified, it is the task of the interpretation algorithm to determine the association between objects and how this should affect the operation of the vehicle. Interpretation is often considered to be separate from computer vision, since it deals with how the data is used and not the actual image processing.

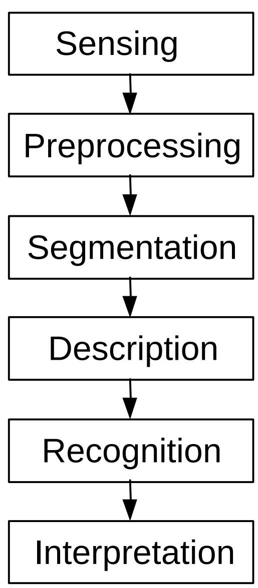

Figure 2.17 – Computer Vision Process

Zhang et al. (1992) states that computer vision's "primary task is to provide a description of the world rich enough to facilitate behaviors such as road following and obstacle avoidance." To achieve these tasks, computer vision is carried out on several levels of complexity. The most basic of computer vision systems are line scanners. Line scanners are quite useful for tracking the movement of objects within a plane. They are extremely fast due to the minimum of data to be processed (McKerrow, 1991). Line scanners produce a very limited view of the world.

The next step in computer vision beyond line-scan systems is gray scale vision. Grey scale systems are capable of edge detection, object identification, target tracking, and numerous other tasks. The larger amount of information comes at the cost of longer processing times and memory requirements (McKerrow, 1991). Grey scale systems are not capable of producing range data and many are quite sensitive to illumination conditions.

Color vision systems add to the capabilities of gray scale systems. Color provides another source of data for object recognition. Tsinas (1995) uses a color system for traffic sign and vehicle signal light recognition. Color systems are typically high-resolution but are not capable of ranging and contain even more data than gray scale systems (Yang and Wu, 1995). Zhang et al. (1992) suggests compressing the number of colors used depending on the application of the system. This approach helps to alleviate some of the processor burden.

Binocular, or stereo, vision is the most complex of all vision methodologies. Stereo vision utilizes two cameras (color or gray scale) at offset positions to mimic human vision. The spatial image coordinates of associated pixels in each image are compared and triangulation is used to determine the range and depth of objects (Fu et al., 1987). Stereo vision is the most computationally intensive vision approach. Precise camera calibration is also essential for accurate system operation.

The ability of computer vision to collect a large quantity of information from a single sensor makes it an excellent choice for autonomous vehicle systems. The relatively low cost, hundreds to thousands of dollars, make vision systems quite economical as well. A major disadvantage is that vision systems suffer from excessive computational requirements, which limits their real-time implementation. The difficulty in extracting range information is another drawback. Nevertheless, the low cost and wide breadth of long-range sensing abilities of computer vision systems far outweighs their shortcomings (Huber et al., 1992).

2.2.4 Sensor Fusion

2.2.4.1 What is Sensor Fusion?

Sensor fusion, at its highest level, is the attempt to emulate the ability of humans and animals to combine multiple sensory input in real time (Hall, 1997; Dasarathy, 1997). In the context of autonomous robotics, Kam (1997) states that sensor fusion is “the process of integrating data from distinctly different sensors for detecting objects, and for estimating parameters and states.” The use of multiple sensors within a “sensor suite” helps to produce a more defined “worldview” by providing a diverse set of input data. The use of complimentary and redundant sensors within a sensor suite can also help to reduce or correct errors produced by a single sensor system (Hall, 1997; Kam et al., 1997; McKerrow, 1991). Ever increasing processing speeds of computing systems as well as advances in sensor technology make attempts to mimic the human sensory system increasingly possible.

The remainder of this section will describe issues inherent in the design of sensor fusion systems and typical fusion architectures. The intent is not to present an in-depth discussion of sensor fusion, but to provide a comprehensive overview of the major concepts involved.

2.2.4.2 Sensor Fusion Considerations

The choice to use a multi-sensor system is ultimately based on the application and on the developer’s preference and experience. When thoughtfully applied, a multi-sensor system can produce an extremely robust description of an autonomous vehicle’s environment. Hall (1997) states that when developing a sensor fusion system for a given application, the following issues should be considered:

1. what algorithms and techniques are appropriate;
2. what architecture (data fusion point) should be used;
3. how is each sensor’s data processed to gain the maximum information possible;
4. what accuracy can realistically be achieved;
5. how can fusion be dynamically optimized;
6. how does the environment affect processing;
7. under what conditions does multi-sensor fusion improve the system?

While sensor fusion can provide obvious benefits to autonomous vehicles, caution should be used when developing any sensor suite. Careless combination of sensors may actually produce greater errors than would be realized using a single, appropriately chosen sensor for a given task. Hall (1997) states that this is often the result of “an attempt to combine accurate with inaccurate or biased data, especially if the uncertainties or variances of the data are unknown.” Experience has shown the overzealous attempts to develop a complex system for a simple application can quickly lead to system failure. Development of a sensor suite should keep in mind the four main problems of multi-sensor systems as set forth by McKerrow (1991):

1. Sensors measure different features such as distance or light reflection. Can the data be effectively combined and benefit the system?
2. Sensor location and field of view will vary. What is the level of difficulty to transform all data to a common reference plane?

3. Sensor time bases vary. Does the addition of a sensor slow the system down or produce communication problems?
4. Sensors may produce noise or incomplete data. How easily is data of different resolutions combined?

The true test of any multi-sensor system is during physical implementation. Performance can be judged by a system's ability to accomplish its goals in a minimum of time with a minimum of control effort. Autonomous applications should also consider the constraints of obstacle avoidance and maximum vehicle velocity (Kam et al., 1997). System developers should question the benefits and drawbacks of adding any sensor to a given system.

2.2.4.3 Sensor Fusion Architectures

Sensor fusion architecture refers to the specific point at which data is combined within a control algorithm. A debate is evident in sensor fusion literature over the definition of "sensor data." Two schools of thought differentiate between completely "raw" sensor data and sensor data with small amounts of preprocessing as is present in many computer vision systems. No attempt will be made here to resolve this debate. The term sensor data will be used to refer to all data originating from physical sensors, which is then used to produce a final control decision.

The three most common fusion architectures are low, high, and hybrid level fusion. Low level fusion (also called centralized fusion) provides "parameters and estimates" to high level control algorithms (Kam et al., 1997). Individual sensor data is first combined into a common data set. This common data set is used to produce an environmental representation, such as a navigation map, which is then used to produce vehicle control (Hall, 1997; Kam et al., 1997). High level fusion (also called state vector or autonomous fusion) uses each individual set of sensor data to produce a state vector or control estimate for the system. This group of decisions is then processed using a subsumption, voting, or neural network based decision architecture (Kam, 1997).

Hybrid fusion takes advantage of both of these approaches. Sensor data is combined at a low or high level depending on the data accuracy of importance (Hall, 1997).

Dasarathy (1997) presents an alternative approach to sensor fusion classification. He expands on the ideas of low and high level fusion by putting forth an I/O-based characterization. The type of input and output of the fusion module is used to describe sensor fusion. The five I/O modes are shown in Figure 2.18.

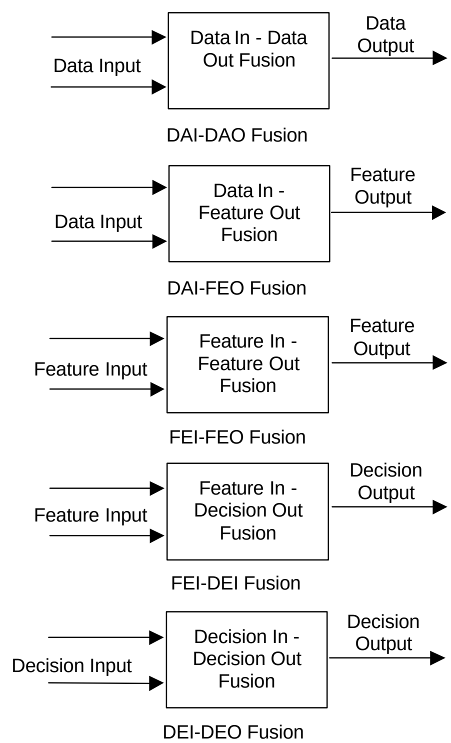

Figure 2.18: Sensor Fusion I/O Modes

The most basic form is *Data In-Data Out* fusion. This fusion type is comparable to low level fusion. Data from multiple sensors is compiled into a common data set that is then used to extract features or produce maps used for navigation. *Data In-Feature Out* is the next highest level of fusion. Data from multiple sensors is used to derive some feature from the environment. Features may include object ranges, profiles, or color and texture. This approach is particularly useful for sensors with similar data structures. Systems using sensors with unique data structures are better suited to *Feature In-Feature Out* fusion. Features are determined by individual sensors and then compared and combined to increase sensor accuracy. The final two levels of Dasarathy's hierarchy are akin to the high-level fusion concept presented earlier. *Feature in-Decision Out* fusion, one of the most common in fusion literature, uses features derived from individual sensors to produce a decision for the vehicle control system. The determination of a specific obstacle type based on *a priori* knowledge could allow a system to take specific actions linked to that obstacle. This fusion technique has wide application in target recognition systems in the military sector (Dasarathy, 1997). The final level is *Decision In-Decision Out* fusion. Control decisions are made for individual sensor data. The decisions are then fused to produce a comprehensive decision for the control system.

Although specific sensor fusion approaches may often be the solution to a given task, developers should not rule out the concept of combining several of the approaches presented. To truly mimic human sensor fusion, advanced systems must be able to adapt to environmental changes. Changes in light level, temperature, terrain texture, or even simple sensor failure could render a static system useless. Flexible fusion systems and self-improving fusion systems that can adapt to and learn from changes in operating conditions point the way to the future of sensor fusion (Dasarathy, 1997).

2.2.5 Sensor-Fusion Summary

The wide selection of possible sensors and fusion architectures present the designer with a rather daunting task. It is often the first inclination to include as many sensors as possible within a system in hopes of "covering all the bases." Engineering principles show us that the simplest systems are often the most effective and robust. Before selecting sensors for a mobile robot, the design should consider several questions:

- What information does the operating environment provide?
- What environmental information is *really* needed to complete the given task?
- How accurate does the information need to be?
- What sensors can provide that information?
- Will multiple sensors aid or hinder that data collection?
- How readily is different sensor data within a sensor suite combined?
- What specific sensor fusion methods and algorithms exist?
- What is the most effective sensor fusion approach for the task?

In the end, sensors and fusion architecture selection is a combination of experience and trial-and-error. The best policy is to start with a simple system then add more sensors as the need arises.

2.3 Navigation

The final portion of mobile robot design is navigation. Navigation is the process of using sensor data to produce a representation of a robot's world that can be used to safely direct it to a given destination. Navigation has three main components: mapping, planning, and action. Mapping uses sensor data to create the environmental model. The map is then used by the planning stage to compute a path to the destination that safely avoids obstacles. Once a path is found, action is taken and the robot is moved forward. A high-level task planner is used to regulate each of these sub-tasks (McKerrow, 1991). The task planner contains the goal or destination as well as constraints such as speed or elapsed time. Human operators control task planning, while the sub-tasks are left to the robot.

This section will explore some of the common approaches to mapping and planning. The concepts as well as their advantages and disadvantages will be discussed. While this discussion is fairly comprehensive, the reader is directed to the recent literature for specific details and nuances.

2.3.1 Reactive Vs. Planning Navigation

Two general control structures can be used to describe navigation approaches: *reactive* and *planning*. Reactive systems map currently perceived sensory data straight into actuator control commands such as speed and steering angle. Reactive behaviors are hierarchically classified as *emergent* or *primitive*. Emergent behaviors are akin to the task or goal set forth by the task planner. Completion of an outdoor obstacle course is an example of an emergent behavior. Primitive behaviors are system actions executed simultaneously or temporally to achieve the emergent behavior (Moreno et al., 1995). Lane following and obstacle detection and avoidance are examples of primitive behaviors.

A common approach to reactive navigation is subsumption. The subsumption architecture approach was used on two prior Virginia Tech autonomous vehicles (Johnson et al., 1996). The primitive navigation behaviors were arranged into parallel layers to control the final actuator output (Figure 2.19). The lowest level uses computer vision to compute an initial navigation direction. The second layer uses an ultrasonic array to modify the initial navigation direction based on long range obstacle detection. The highest level, tactile sensors, is used in the event that the lower levels fail and the vehicle collides with an obstacle. This parallel layer arrangement allows higher levels to alter or completely override lower level behaviors.

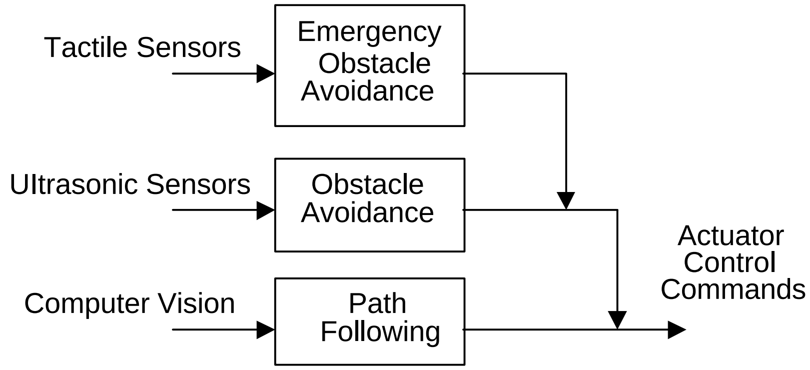

Figure 2.19: Subsumption Architecture Reactive Navigation (Johnson et al., 1996)

Reactive navigation has the advantage of extremely quick processing, since no environmental model is built. This approach is good for low level applications such as wall following and obstacle avoidance (Castellano et al., 1996). Reactive navigation typically relies upon a linear or fixed hierarchy of behaviors, thus limiting its ability to deal with complex situations. The failure of a single behavior module within a linear arrangement can cause a cascade of failures throughout the remainder of the system. Hierarchical architectures present difficulty in selecting the precedence level of each behavior (Moreno et al., 1995). Reactive approaches are best suited to well structured environments and as supplements to industrial beacon following AGV's.

Planning navigation relies upon a much more complete map of the surrounding environment. The map information may be provided *a priori* or produced from sensor data. Planning approaches are much more computationally intensive and may rely upon incomplete sensor data. Planning has the advantage of memorizing the local and global environment via various map forms. Mapping allows planning systems to produce more intelligent decisions based on a larger set of sensor data.

The remainder of this section will focus on mapping and path planning approaches. These topics presented are applicable to both reactive and planning navigation, however; an advantage is placed to planning approaches based on the possibility of data memorization.

2.3.2 Mapping

Maps are extremely important in the navigation process. Maps are necessary for autonomous vehicles since they "require the ability to recover robust spatial models of surrounding world from sensory information" (Elfes, 1991). The type of information representing the environment can be used to classify maps as path, free space, object oriented, or composite. The different map types vary in applicable data structures and resolution. Certain maps are better suited to specific path planning approaches, although most are interchangeable.

Maps may be either local or global in nature. Local maps define a finite area around the vehicle. The boundary of a local map may be the walls of a room or a set radius about the vehicle. Global maps are much broader in nature. Numerous floors of an office building or the entire length of an obstacle course may set the extent of global maps. Global maps are used to set a general path to the final goal or destination. Local maps include detailed information, such as unexpected obstacles, to continually update the global map (Chung et al., 1995, McKerrow, 1991). Reactive navigation can rely solely on local maps. Planning navigation requires global mapping, at least to the extent of memorized local maps, in order to continually update the navigation path.

A major concern for all map types is sensor noise and uncertainty. The detected position of an obstacle will fall within a range of the actual position due to sensor resolution. This is especially significant when using low-resolution sensors, such as ultrasonic range finders. Internal vehicle sensors, such as encoders, are also prone to error. Mapping often relies upon sensor data that is relative to the location of the vehicle. The vehicle's location is, in turn, relative to a global coordinate system. Other maps rely upon multiple sensor data sets that must be aligned based upon relative vehicle movements over a period of time. Error in vehicle odometry can quickly cause misalignment during coordinate transformation. This factor should be carefully considered during map selection. Maps such as certainty grids inherently average sensor data. On the other hand, obstacle maps utilizing vector representations rely upon accurate sensor data.

The following sections will outline various mapping approaches. To keep the discussion general, the maps are presented without regard for any specific navigation (path finding) strategy.

2.3.2.1 Path Maps

Path maps contain lists of paths or movements based upon human *programming* or *teach* methods. Program maps are created offline and often contain room layouts and landmark locations. Teaching involves physically moving the robot and periodically recording motions and stop points. These maps are stored internally and used to guide the vehicle with the aid of external beacons or guide wires. Path maps are used extensively in industrial applications where there is little variation in vehicle path or application (McKerrow, 1991). Path maps are typically of little use in fully autonomous vehicles.

2.3.2.2 Free Space Maps

Free space maps are concerned with the space between obstacles rather then the obstacles themselves. As the vehicle proceeds through an unknown environment, data concerning the extent of free space is collected. The locations of occupied areas, or obstacles, are recorded in a spatial graph (McKerrow, 1991). Once sensor data has been collected, the free space may be represented in several ways. Voronoi diagrams are one popular method. Voronoi diagrams partition the environment into polygons whose edges are the set of points equidistant to neighboring points (Figure 2.20-a). The polygons are then labeled as being empty, containing an object, or unknown. Voronoi diagrams may also be constructed using Delaunay triangulation. Delaunay triangulation computes the shortest lines between the center points of neighboring objects (Figure 2.20-b). The Voronoi diagram is found by computing the perpendicular bisector of each connecting line (McKerrow, 1991). The Voronoi diagram boundaries represent free paths for vehicle motion.

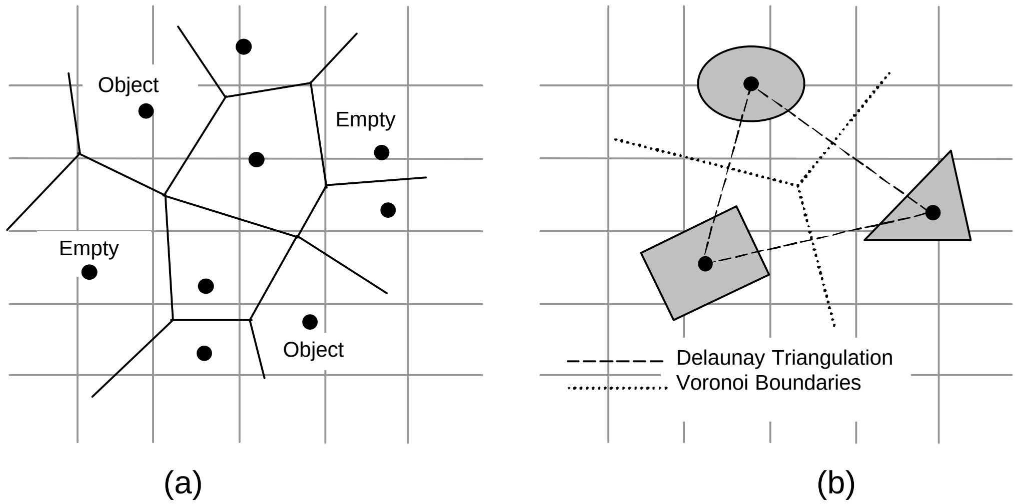

Figure 2.20: Voronoi Diagrams

Brooks (1983) presents a free space method using generalized cones. A generalized cone is formed by sweeping a specified cross-section along a spline. In this case, the splines are straight lines, equidistant between two given objects. The swept cross-section is a line perpendicular to the spline and whose end points are defined by the outer boundaries of the objects (Figure 2.21). Each of the generalized cones forms a “freeway” such that the center spline represents the safest path in each freeway. Overlapping cones form possible vehicle routes and intersections.

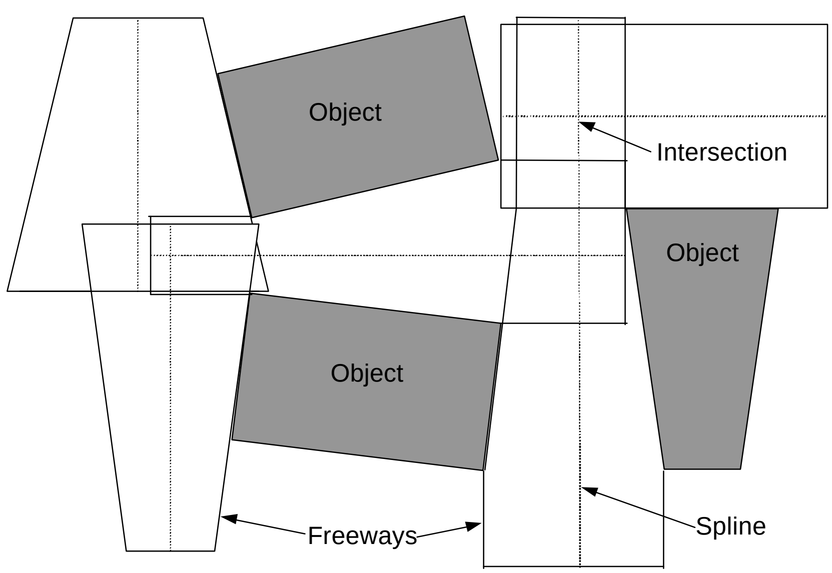

Figure 2.21: Generalized Cone Free Space

2.3.2.3 Object Oriented Maps

Object oriented maps are concerned with the location of obstacles in the environment. The free space is determined by implication (e.g., areas without obstacles). Objects are typically recorded as a set of x,y vertices relative to a global reference frame. Alternatively, the position and orientation of each object can be specified in a linked list of x, y, and θ coordinates (Lozano-Perez, 1983; McKerrow, 1991). Both approaches are shown in Figure 2.22. Object oriented maps can produce a very small data set for a given environment. However, accurate object models are highly dependent upon precise, detailed sensor data, thus limiting practical applications (Zelinsky, 1992). Object maps are most effective in situations where the environment is well known.

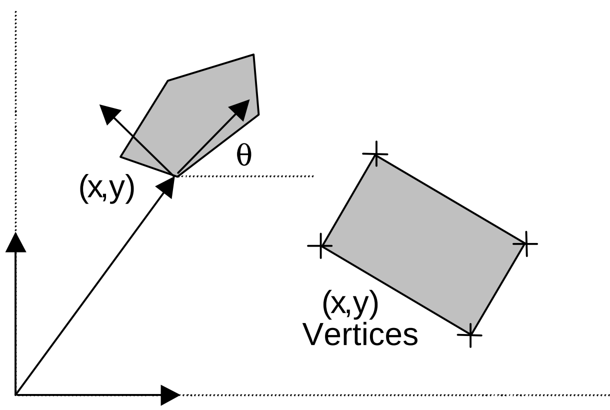

Figure 2.22: Object Oriented Map

2.3.2.4 Composite Maps

Neither of the previous mapping approaches completely represents the environment. Free space maps ignore objects, while object oriented maps are only concerned with objects and free space is implied. Composite maps include data concerning both objects and free space.

Brooks and Lozano-Perez (1985) present a composite approach based on configuration-space where "the configuration of a rigid object is a set of independent parameters that characterize the position of every point of an object.” The set of parameters are typically x,y coordinates relative to the equations of motion for the vehicle. Thus, the configuration space specifies the boundary between free space and object space during vehicle movement. Lozano-Perez (1983) has developed a method in which the vehicle is reduced to a single point. The object size is concurrently increased in proportion to all possible vehicle orientations that could cause collision. This facilitates quick path planning, since computing a point-object intersection is easier than that of an object-object intersection. However, this approach suffers from a computationally intensive front end and the undesirable elimination of certain potential paths.

A more common approach to composite mapping is the grid-based map. Grid based maps superimpose a geometric structure over the environment. Grids may be tessellated into finite areas or points. Each individual grid “cell” or point represents a finite amount of space within the environment. (Zelinsky, 1992; McKerrow, 1991). Grids may contain data concerning occupancy, danger, reachability, observability, reflectance, etc. The most common grids, however, are *occupancy* and *certainty* grids.

Occupancy grids combine sensor data for each finite grid cell. A threshold is then applied to each grid cell based on heuristic data. The cell is then determined to be empty, occupied, or unknown (Figure 2.23). This binary approach locates both free space and objects. The disadvantage to occupancy grids stems from the finite area representation of the world. An object may only partially occupy a cell but the cell will be listed as fully occupied (Elfes, 1991). This problem can be overcome by using *certainty* or *quadtree* grids.

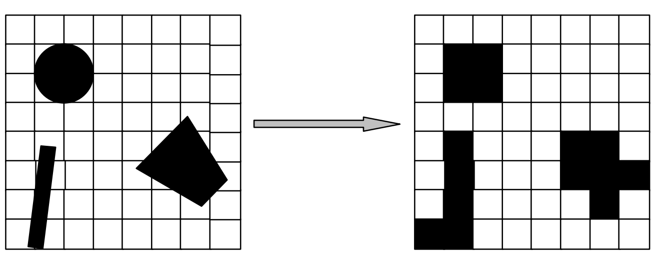

Figure 2.23: Occupancy Grid

Certainty grids are similar to occupancy grids with the exception of data thresholding. Thus, the grid contains the probability that a block is occupied. Certainty can be represented by a number range or grayscale color (Figure 2.24). The lower probability areas indicate possible unobstructed paths. Certainty grids still suffer from a finite resolution of the world.

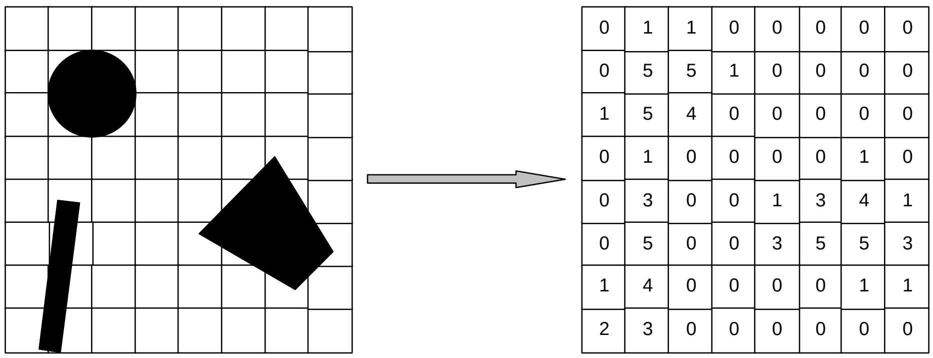

Figure 2.24: Certainty Grid

Quadtree grids can contain occupancy or certainty data. A quadtree grid is initially set to a coarse resolution. As data is collected, areas containing obstacles can be divided into increasingly smaller sections to better represent object profiles (Figure 2.25). This approach minimizes data storage since large unoccupied areas can be represented using a single grid cell (Zelinsky, 1992).

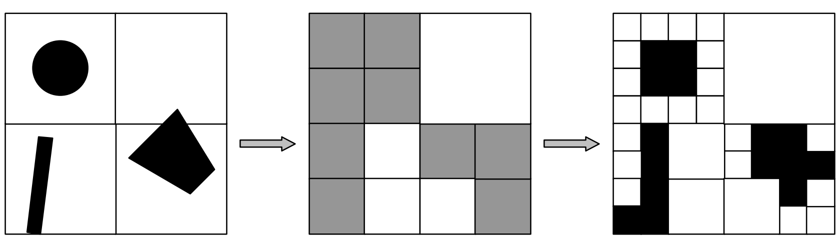

Figure 2.25: Quadtree Occupancy Grid

Grid based mapping has several advantages. Grids are good for unknown environments since several sensor types can be easily combined. Probabilistic models are often used to compensate for accuracy and noise variations among sensors. Grids can be continually updated to build a more robust world model. Certainty and quadtree grids specifically have the advantage of averaging or blurring together multiple data sets during integration. This helps to eliminate errors caused by vehicle odometry. The data structure also lends to the easy integration of local maps into larger global maps (Elfes, 1991).

Grids do have several disadvantages. As mention previously, grids generate a finite representation of the world. The representation may also be discrete, full or empty, in the case of occupancy grids. Thus, round objects may be represented as squares or occupy more grid space then they do in reality. This can be partially overcome by using quadtree grids or different grid cell geometries, such as hexagons or triangles. Grids are typically computationally intensive. The data structure is highly dependent upon grid resolution and field of view. Again, this may be partially overcome by using a quadtree approach (Zelinsky, 1992).

2.3.3 Path Planning

Once a map of the environment has been generated, a safe, obstacle free course to the goal or destination must be found. Path planning may occur on the global and local level (Hwang, 1996). Global paths give general direction to a destination. Local path planning modifies the global path based on unexpected obstacles. The objective of both path planners is to find the optimal trajectory to the goal. Distance, time, power requirements, etc may be used to define an optimal path. Path planners may be broadly classified as graph searching or potential fields (Warren, 1989).

2.3.3.1 Graph Searching

Graph searching operates by producing a graph or “tree” of all collision-free spaces between the vehicle and the goal. The free spaces, or “leaves”, are then linked together to produce a path to the goal (Warren, 1989). Graph searching may be applied to grids, free space maps, and object oriented vector maps (Figure 2.26). Grid map paths (a) are generally specified as a combination of block center point coordinates. Free space map paths (b) may be specified as Voronoi diagram edges or generalized cone splines (McKerrow, 1991; Brooks, 1983). Object oriented maps (c) rely upon vectors which are tangential to objects or connect object vertices (Lozano-Perez, 1983).

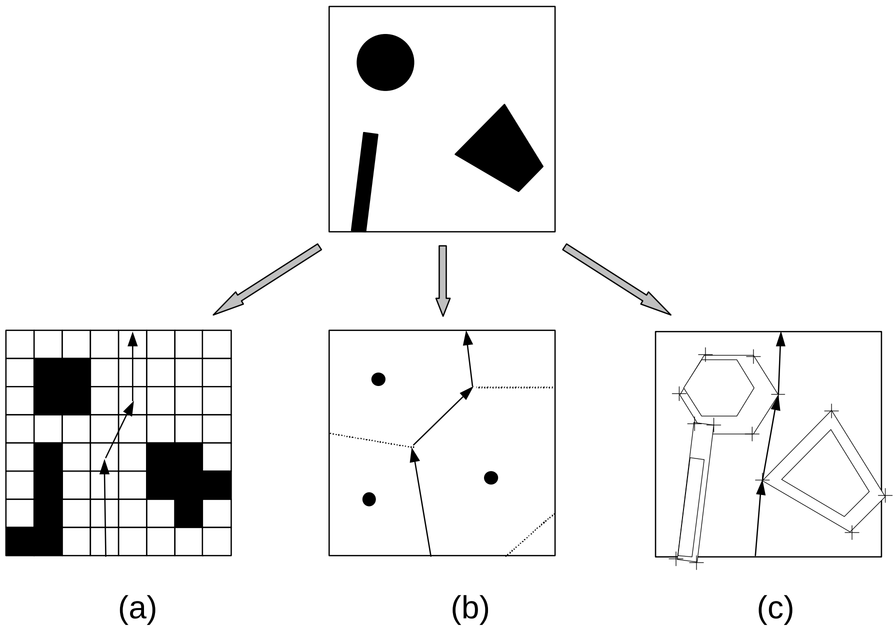

Figure 2.26: Map Paths

Regardless of the map type, a graph of all possible routes is computed. The paths are then searched and the lowest *cost* path is selected. Cost may be specified in several ways. The overall length of the path is one typical method. Zelinsky (1992) outlines a method in which the free blocks of an occupancy grid are assigned weights based on the distance from the goal. Weights increase away from the goal along a wavefront similar to the ripples caused by dropping a pebble in a pond. This approach tends to drive the searching algorithm towards the goal since the cost decreases as the vehicle proceeds “downhill” towards the goal. Certainty grids produce paths that vary in cost depending on the certainty of passing through occupied cells. Paths containing a majority of obstacle-free cells will inherently produce lower certainty paths.

A typical graph of possible paths is shown in Figure 2.27. The lettered blocks indicate free spaces within a grid or object oriented map. The interconnecting lines are possible paths between adjacent open cells from an occupancy grid or vectors connecting the nodes of an object oriented map. The numbers along interconnecting graph lines are representative of various costs such as distance and certainty. A common graph searching procedure is known as the A* algorithm. The A* algorithm is a branch search method which computes optimal paths by eliminating multiple paths to subnodes and the goal based on cost comparison. The resulting path should have the lowest possible cost (McKerrow, 1991; Hwang, 1996). Graph searching is guaranteed to find a path if one exists. However, graph searching can be computationally intensive, especially in the presence of numerous paths produced by large maps (Zelinksy, 1992).

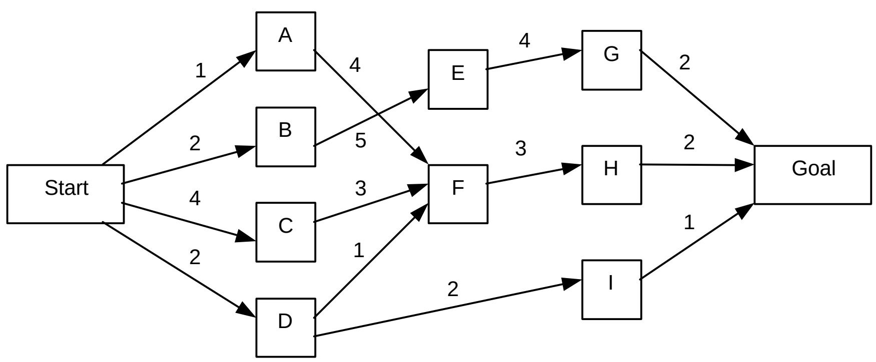

Figure 2.27: Graph Path Search

2.3.3.2 Potential Fields

Potential fields may also be applied to numerous map types. The potential field assumes that the vehicle, goal, and obstacles carry virtual electrical charges (Figure 2.28). The goal is assumed to have a negative (attractive) field that draws the vehicle towards it. Obstacles are given positive (repulsive) fields that tend to push the vehicle away. A continuous potential field will create minimum potential valleys that drive the vehicle to the goal while avoiding obstacles (Warren, 1989, Hwang and Ahuja, 1992). Potential field masks act as exponential functions such that repulsive forces increase as the vehicle gets closer to an obstacle. Potential fields are most commonly used locally but they may also be used globally (Hwang and Ahuja, 1992).

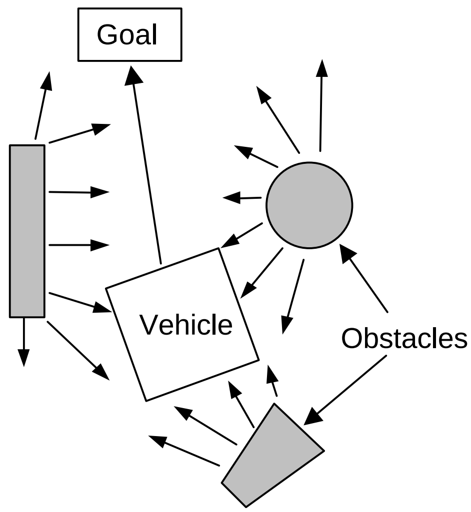

Figure 2.28: Potential Field Model

One advantage of potential fields is that they are relatively fast computationally. A single path is calculated as a function of the field. The need to compute numerous paths and then search for the best one is eliminated (Warren, 1989). The resulting path is often similar to the path computed using Voronoi techniques (Hwang and Ahuja, 1992). Another advantage of potential field models is that they provide a good estimate of distance and shape of objects. This can be used to produce smooth, continuous paths (Hwang and Ahuja, 1992).

The main disadvantage of potential fields is the presence of local minima. Multiple obstacles may result in “gravity” wells that force the vehicle into an inescapable trap. This effect is most common during local mapping, especially if the final attractive goal is located at a great distance from the vehicle. To avoid this, Warren (1989) suggests considering the entire path of the vehicle to help push it through local minima.

2.3.4 Navigation Summary

The selection of a navigation strategy can be quite daunting. The chosen navigation strategy should address several points:

- Should a reactive or planning approach be employed?
- Are existing maps available?
- What sensors are available?
- What accuracy is needed?
- How much computational power is available?
- What refresh rate is desired?

Several navigation algorithms are readily available in literature and on the Internet. Again, the designer is encouraged to use the simplest approach applicable to a given task. Any navigation strategy used must find a balance between model accuracy, speed, and reliability.