Obstacle Detection and Avoidance Through Image Processing-
Review of Autonomous Vehicles Sensing and Classification

Digital Image Processing - EML 5820
Brian White
Dr. Kreidl
5/27/15
Table of Contents

1. Autonomous Vehicles
   1.1 History
   1.2 Overview

2. Autonomous Emergency Braking
   2.1 Overview
   2.2 Applications

3. Sensors
   3.1 Optical (Vision Systems)
   3.2 Radar/Lidar
   3.3 Sonar

4. Algorithms
   4.1 Segmentation
      4.1.1 Edge Detection
      4.1.2 Edge Linking and Boundary Detection
   4.2 Representation and Description
   4.3 Object Recognition

5. References

6. Figures
1. Autonomous Vehicles

1.1 Overview

The dream of autonomous vehicles is here with working prototypes now available. These robotic cars react faster than humans, have 360° perception, and do not get distracted, sleepy, or intoxicated. A staggering 37,423 people died in car accidents in the US in 2008 which could have been drastically lessened if this technology was available to the general public. The test cars have successfully driven 1,000 miles without any human intervention and more than 140,000 miles with occasional human control. Along with the driver who can intervene if necessary, there is a technician in the passenger seat to monitor the navigation systems. The vehicle senses the nearby environment by using a camera to extract data in the form of an image, much like how our own vision works, and processes the image to detect and avoid obstacles in real-time. A microprocessor (i.e. computer) acts much like our own brains and uses algorithms, or code, to interpret the image. If a camera is used for the sensor the image is easily interpreted by a human, on the other hand if the sensor uses electromagnetic energy outside of the visual spectrum or sound the image may only have meaning to a machine processing it. Processing the image is performed by applying the filtering techniques learned in Digital Image Processing to segment the image information. Decisions are then made via algorithms which take the segmented images as input and apply logic allowing the car to react in the optimal manner to stay on course and avoid any obstacles along the way. Such obstacles are essentially limitless; the main ones include other vehicles, curbs, guard rails, cones, pedestrians, bicycles, and animals. How to handle an obstacle which is suddenly encountered is challenging. Most obstacles are discernible with the google self-driving car by using a combination of many sensor types, however, sometimes objects such as harmless trash or light debris cause the vehicle to veer unnecessarily. Some obstacle detection limitations exist, such as temporary traffic lights, but these are projected to be resolved by 2020. Additionally, the car's lidar sensor has difficulties spotting some potholes, discerning when humans such as a police officer are signaling the car to stop, and may spot “invisible” obstacles such as exhaust. Centering the car on the roadway, which is often not flat but undulating, is a good starting point to achieve autonomous driving. However, the algorithms must also have awareness of obstacles on the road for the technology to be safe for the masses. An example of an obstacle is a car ahead which suddenly brakes; sensors on-board the autonomous vehicle need to interpret the rapidly slowing car ahead and react in real-time to avoid an accident.

1.2 History

The visionary and brainchild behind the self-driving car concept is Sebastian Thrun, the now 48-year-old director of the Stanford Artificial Intelligence Laboratory, a Google engineer, and the co-inventor of the Street View mapping surface. In 2005, he led a team of Stanford students and faculty members in the design of the Stanley robot car, winning the second Grand Challenge event of the DARPA competition. This netted a $2M Pentagon prize for driving autonomously over 132 miles in the desert. Less than 2 years later another event proved that autonomous vehicles could drive safely in urban settings. The project team has 15 engineers as well
as over a dozen people hired to sit in the driver's seat. Google's self-driving car concept began in 2008 and the car, without pedals and a steering wheel, was presented in May 2014. A fully functioning prototype was available in December 2014 and was test driven on San Francisco Bay Area roads by early 2015. In 2012, Google's founder Sergey Brin stated that the Google Self-Driving car will be available for the general public in 2017 and in 2014 this schedule was confirmed by project director, Chris Urmson, who indicated a possible release from 2017 to 2020.

Google's autonomous car incorporates ~$150,000 in image acquisition and processing equipment including a ~$70,000 lidar system (Velodyne 64-beam laser range finder) that is mounted on the top. The Google car is equipped with three radar sensors in the front and one in the back, a scanning lidar on the top which creates a detailed and highly accurate 3D digital map of the environment to estimate the car's location, and a position sensor on the left rear wheel to measure movements and position on the map accurately. The merits of using radar, or lidar, versus cameras are important to understand. The cameras provide high resolution, often color images (not just grayscale) that the driver can easily interpret. Radar has the advantage of greater range, and is primarily used for sensing large sized objects, along with the ability to collect data in inclement weather with no distortion in the image. For the foreseeable future both types of sensors will be necessary, and the replacement of one sensor type with another is highly unlikely. The lidar sensors create multiple high-dynamic-range (HDR) images that are combined in a burst mode. Using image recognition software, objects can be viewed up to 600 feet away without blind spots.

The car follows a route programmed into the GPS navigation system and drives the speed limit, which for each road is included in its database. A rear-view mirror camera detects traffic lights and moving objects. This technology can be outfitted for existing cars by using cruise control cameras that already exist. Faster image processors have enabled smarter algorithms that can differentiate between pedestrians and vehicles and color cameras can detect when the vehicle ahead brakes. For safety, the car also makes announcements such as “approaching a crosswalk” or “turn ahead” to alert the driver in the off chance they should have to take control of the car. The Tesla model S autopilot car uses radar (in the front), 12 long range UV sensors (similar to lidar-using UV, visible, or near infrared light where a narrow laser-beam maps physical features in the environment with resolutions up to 30 cm/px, up to 75m away), and a forward camera. The hardware costs $4,250 and was added to all new cars in October 2014. Unfortunately, for cars made before this date a retrofit is not available. This tech package also includes software that will engage in auto-steering or “autopilot”, but unlike the Google car, the driver is still advised to pay attention to the road to assist if necessary. Additional capabilities include autonomous lane changing when the turn signal is activated and warning the driver when he is speeding. The car's cameras read the speed limit signs and if the driver is speeding an announcement is made. Additionally, the driver can have the vehicle automatically slow down when the speed limit is exceeded.

2. Autonomous Emergency Braking

Autonomous Emergency Braking (AEB), or Emergency Braking System (EBS), is employed to help stop
the vehicle faster and thus avoid a possible collision. Automatic cruise control (ACC) radar sensors are used as part of the collision mitigation brake system (CMBS) and if the gap with the car in front becomes too small then the car automatically brakes, and accelerates again, to maintain a safe distance. AEB uses radar, lidar, ultrasonic, infrared, or optical sensors to identify potentially hazardous targets.\(^2\)\(^3\) One configuration uses a camera to aid in the classification of an object and then employs radar to detect the relative location of it.\(^2\) The effectiveness of the vehicle to avoid a crash is between 10 -72 \%, with fatal rear-end crashes lessened by 36 – 44 \%.\(^2\) When Mercedes introduced the technology in the late 1990s they claimed that it helped to shorten stopping distance by 45\% (with only 10\% decrease in stopping distance for skilled drivers).\(^2\) In 1996, AEB was introduced in the Mercedes-Benz S-Class and SL-Class models and in 1998 Mercedes made the feature standard on all of its vehicles. Since then several companies, including Acura, Audi, BMW, Infiniti, Land Rover, Rolls Royce, and Volvo, have offered their own versions. The European Commission (EC) has estimated that brake assist could save 1,100 lives per year. According to the US Insurance Institute for Highway Safety (IIHS) the number of 25 year olds that die per year in car accidents is an astounding 400,000 people (>1,000 people per day). Worldwide the total amount of deaths from car accidents is a staggering 1.3M deaths per year (or 3,287 deaths per day).

3. Sensors

In the localization process, sensors along with a global positioning system (GPS), an inertial measurement system (IMS), and a wheel encoder are employed. Sensors are necessary to figure out very accurately where the car is on the road. Sensor fusion is employed to integrate the data from all the different sensors. By combining multiple sensors the quality of the data is improved but it is not always reliable. For instance, believing every obstacle report increases the vehicle's visibility. However, the drawback to reduced “blindness” is that “ghost/phantom” obstacles are potentially introduced (i.e. sense an obstacle that is not actually there). The best approach to limit confusion is to have each sensor be better at a particular problem and trust that sensor over that particular regime of examination.

The main sensors types used for autonomous driving are cameras, radar, and lidar vision systems. Lidar is a variation of radar, which uses light energy as the illumination source. Other sensors that exist are: ladar (laser radar), thermal, long-wave infrared (employs emitted light instead of reflected light), hyperspectral (consists of a camera that works in many color bands), ultraviolet, and sonar (sound waves).

3.1 Optical (Vision Systems)

Sensors collect data using vision based techniques (i.e. video camera), or employ other types of sensors such as ladar (laser range) or sonar (sound range), to compute the position of obstacles.\(^4\) The advantages of using video are its reliability, low cost, and low power consumption.\(^4\) However, computer vision is no where near good enough today to detect all the important features in a camera image with the reliability necessary for safe driving. The main reason cameras alone are not sufficient for computer vision is that they require a great deal of central processing units (CPU) or custom chips to interpret an image. Cameras must deal with lighting variation;
objects are routinely subject to moving shadows and can be lit from any direction or not lit at all. Cameras require illumination at night where headlights do not provide sufficient lighting. A video camera uses optics to focus the incoming light onto an image chip, which then converts the light into a digital signal. A microprocessor takes the digital data as an input, processes it according to the software written in memory using sequential digital logic, and calculates the distance and velocities of other vehicles that are within a close proximity. In order to process the data in real-time the CPU should be capable of transmitting and receiving image data at a fast data rate, however, the faster the processing speed the greater the cost. The image sensors in a camera are generally CMOS, yet, CCD types are also used. There are many advantages to using a CMOS sensor over a CCD sensor; these include smaller size, less power consumption, faster speed, and advanced feature capabilities. Monocular or stereo-based (binocular) vision systems, on-board the autonomous vehicle, estimate in real-time the position and orientation of the vehicle related to the 3D road plane parameters. Binocular systems use stereopsis to determine the 3D motion and parallax. In the field of visual odometry, the ego-motion problem refers to estimating a car’s relative motion from camera images comparing road lines, street signs, etc.

The initial tasks to consider in solving the complex problem of autonomous driving is to ensure the car stays centered in its lane and travels at the correct speed. Traveling at a constant speed using cruise control is not a new technology, in fact, it has been available in cars since 1900 relieving the driver from some of the monotony of highway driving. The next logical technological step is to include steering cruise control. During long drives both of these technologies allow the driver to expend less effort, by freeing up the driver’s attention to anticipate changes in the steady flow of traffic and obstacles that may arise, as well as, driving the correct route. The monotonous nature of typical highway driving is mitigated and the journey is safer since the car can hold its speed and position if the driver becomes distracted or falls asleep. One configuration consists of a button which the driver hits once the car is driven onto the highway, arrives in the correct lane, and is traveling at its desired cruising speed. The button activates the advanced cruise control allowing the car to hold its lane while also maintaining a constant speed. The lane keeping assist system (LKAS) employs a camera beside the rear-view mirror to monitor the road markings, feed the data to a computer, and apply the correct steering torque (i.e. slight nudges) to the wheel to keep the car centered in its lane. One way to improve the lane holding provided by the LKAS is to incorporate lane changing assist. Lane changing assist uses cameras to detect cars in lanes nearby once the driver activates the turn signal. If no obstacle was detected, the car will proceed to change lanes by applying slight steering adjustments which can be easily overpowered by the driver if necessary.

Lateral control vehicle systems allow a vehicle to follow a lead vehicle by sensing the lead vehicle using either infrared, ultrasound, magnetic markers, or some other sensing method. The vehicle's lateral movement is controlled by a lateral control algorithm. The image processing algorithms are either based on a pixel intensity profile or on vanishing points in the image plane. The control algorithm is based on a simple classical P-control and a control scheme based on $H\infty$. The automatic lateral control is a two-step process: the first step is to obtain the position of the vehicle with respect to the road from the video image and the second step uses this information as the input to a control
Algorithm. Classical controllers (i.e. P, PI, PID), fuzzy logic, neural network, etc., in conjunction with the algorithms generate the steering angle of the vehicle that is necessary to maintain its desired position on the road (see the function block diagram shown in figure 1). For the lateral vehicle control many of the existing mathematical models are adapted to the specific hardware in use. Parameter estimation is employed for hard-to-measure parameters in the system. Attempts can be made to design a modern robust controller based on the model gained by parameter estimation.

3.2 Radar/Lidar

Radar, an acronym for radio detection and ranging, uses radio waves as the energy source, which are larger in wavelength compared with visible light waves. Due to radar's larger wavelength, it is better suited to detect large obstacles, such as vehicles or pedestrians. Radar has the advantage of being able to see through fog, but, the downside of using light based radar (lidar) is that it may detect exhaust smoke- an “invisible” obstacle. Lidar is often in scanning form- typically rotating at 10 rev/sec- where just one revolution scans the environment creating a 3D image. There are radar sensors behind the emblem on the front of the car that constantly monitor the distance of the car in front. As mentioned earlier, radar is used for AEB allowing the car to recognize if a collision is imminent. The system first warns the driver by an audio message and visual pictogram on the dashboard and if no action is taken the seat belt pre-tensioners will lightly tug the driver giving a physical warning. If there is still no action taken the system will apply a strong braking.

3.3 Sonar

Sonar, which was around before radar, is an acronym for sound navigation and ranging. This technique uses sound propagation to locate obstacles. The autonomous cars employ active sonar, which sends out a pulse of sound (ping) and listens for reflections (echoes) of the pulse. The pulse is generally created electronically using a sonar projector consisting of a signal generator, power amplifier, and electro-acoustic transducer/array. A beamformer is usually employed to concentrate the acoustic power into a beam which can then be swept to cover the search angles of interest. Another option is to use multiple beams. Sonar and radar sensors both have a limited field of view, but the range of sonar is only 6 feet compared with radar's 200 foot range. Sonar's acoustic frequencies range from low (infrasonic: 0.001 - 20 Hz) to high (ultrasonic: 20 kHz – several GHz).

4. Algorithms

Along with sensors that have improved performance and are less expensive, the study and construction of algorithms that learn and make predictions from the sensor data is very important to the on-going progress in autonomous vehicle driving. The car's ability to adapt to new circumstances is a necessary condition. An important facet in this endeavor is to detect and extrapolate patterns: an area of research known as machine learning. Pattern recognition in algorithms is analogous to primate vision and how neurons in the inferior temporal cortex allow for object recognition. A way to judge the efficacy of machine learning is to give the
algorithms more experience and measure how much the performance on tasks improves (the assumption here is that the performance improves with more experience). In machine learning predictions are made based on known data (generalizations are made based on a limited data set) and the performance is evaluated on the ability to reproduce the known knowledge. Conversely, knowledge discovery and data mining (KDD) is the discovery of unknown properties in the data. KDD is the next step, after solving the task of autonomous driving using machine learning, to make the car's algorithms smarter and thus give it the ability to learn without having to provide extensive amounts of evidence. Computational learning theory as well as computational statistics are two underlying areas of machine learning. Machine vision allows some detection of other vehicles, pedestrians, edges of the road, and road markings. It is also able to detect objects independent of scale or rotation. Nonetheless, when a more complex, non-localized analysis of an image takes place the term computer vision is more appropriate. When speaking of computer vision, as opposed to machine vision, it is assumed there is improved object recognition.

After surveying the stereo algorithms from the ARPA Unmanned Ground Vehicle program the SRI algorithm (developed by INRIA) was discovered. The SRI algorithm attempts to produce a set of high quality matches using hierarchal techniques, however, with existing hardware this method falls short of real time speed requirements. Attempts to establish the correspondence of selected points at the frame rate were developed by TELEOS. Selection of these points is critical for the still open problem of obstacle detection. A variation of the typical binocular stereo techniques for detecting obstacles is trinocular stereo, which uses three cameras instead of two in order to simplify matching. Regretfully, speed is an issue using trinocular stereo sensors and algorithms with a speed that is 5-10 times too slow.

Two road approximation techniques generally used for stereo based driver assistance applications are v-disparity space and Euclidean space. The v-disparity space approach uses prior knowledge of the approximated road scene and 3D reconstruction software to create a disparity image. The disparity image and values are used to compute the v-disparity representation and in the process avoids having to compute 3D data points, whereas, the Euclidean space computes the 3D data. A plane in the Euclidean space (assuming the road geometry fits to a plane) becomes a straight line in the v-disparity space, and subsequently, plane fitting becomes segment fitting. Instead of segment fitting via the least-squares method, the Hough transform (HT) with a voting scheme is proposed to extract the largest sub-set of points that define a straight line. The results showed that working in the Euclidean space, by using a filtering technique, gives better results over all road surface contours than those obtained in v-disparity space using the HT. Nevertheless, the extraction of planar representations with the HT in v-disparity space is faster and is better-suited for highway driving where the roadway is flat. The Euclidean space approach is two staged consisting of structuring 3D data points so that real-time processing can be performed, followed by, performing a fitting technique to find the best surface to approximate those data points. Least squares is the most common fitting technique used due to its simplicity. The least squares fitting method minimizes the sum of the squares of the offsets instead of taking the absolute values of the offsets. This allows the residuals to be treated as continuous differentiable quantities. A 2D (i.e. YZ plane, lateral view)
representation of the 3D data points is generated, and then a RANSAC based least squares approach is used to fit a plane to the road candidate points. The sensor used for data acquisition was a commercial stereo vision system called Bumblebee (www.ptgrey.com).

4.1 Image Segmentation

Image segmentation, also referred to as classification, takes an input image and groups it into its constituent regions or objects. The output image is representative of the attributes extracted from the input image. In 1925, Max Wertheimer, a renowned psychologist who was one of the three founders of Gestalt psychology, listed several key factors which led to perceptually organizing an image including similarity, proximity, and good continuation. There are many ways to partition an image into subsets, and there may not be a single correct answer, rather a Bayesian approach is preferred. A Bayesian approach means the level of detail to which the subdivision is carried out depends on the problem being solved.

Image segmentation should be accomplished from the big picture downward, much like a painter who marks out the major areas first. Segmenting the image of a car's environment is a non-trivial task. The partitioning is inherently hierarchical and thus more like a tree structure than a “flat” one. Low-level cues, such as brightness, color, texture, and motion attributes, are combined with mid- and high-level cues to either confirm the low-level group or select one for further attention. Improving the probability of accurate segmentation is critical with the eventual success or failure of computerized analysis procedures hinging on this accuracy. Selecting the types of sensors to most likely enhance the objects of interest while diminishing the contribution of irrelevant image detail is important. The result of segmentation is a simplified or changed representation of an image into something more meaningful and easier to analyze. Segmentation algorithms are typically based on two properties of intensity values: discontinuity (i.e. abrupt changes in intensity) and similarity, both of which are defined by a set of predefined criteria. Segmenting in terms of discontinuities locates objects or boundaries, such as points, lines, and edges, using either derivatives if the signal is viewed as continuous or difference methods if it is viewed as discrete.

4.1.1 Edge Detection

Discontinuities, such as lines or edges, where two regions meet, are detected by either applying 1st or 2nd order derivatives (i.e. spatial or high-pass frequency filters). The magnitude of the 1st derivative can be used to detect the presence of an edge at a point in an image and the 2nd derivative's sign can be used to determine whether an edge pixel lies on the light or dark side of the image. Additionally, the zero-crossings can be used for locating the centers of thick edges. Detecting the location and orientation of roadway painted line segments relative to the car is one method to keep the car centered in its lane. The roadway lines in the digital images have edges that are blurred and noisy, a result of focusing mechanisms (i.e. lenses in the case of optical images) and noise from electronic components. Even though the image noise may be hardly noticeable to the eye, derivatives are very sensitive to noise and for this reason image smoothing is useful for noise reduction. If the edge
transition is more than one pixel across then modeling the line as a ramp or roof edge is more appropriate than the ideal step edge. The ramp or roof edge models, used to represent the actual edge, the edge points become any point contained in the edge profile and the set of such points that are connected are referred to as an edge segment. The models allow for mathematical expressions for edges in the development of image processing algorithms. The algorithms performance is the difference between actual edges and the models used in developing the algorithms. The brightness of the line will be a function of the line thickness and the filter size chosen (as well as the thresholding value selected). When line detection is discussed the assumption is that lines are thin with respect to the size of the detector (laplacian filter). The line width that one is trying to detect should be similar in size to the filter used, such as using a 3X3 filter to detect a 1 pixel line width. The filter types applied to an image are typically either horizontal, +45°, vertical, or -45° based on the line orientation in question. The four different filters vary in the direction of the larger coefficients (i.e. the preferred direction has values of 2) within the array of filter values (i.e. -1) that make up the filter. Applying the filter performs a sum of products of the mask coefficients with the intensity values in the region encompassed by the mask, resulting in the response of the mask at the center point of the region. To find if a given region in an image is more associated with a certain line orientation apply each of the four filters (horizontal, vertical and the two diagonals designated as k= 1, 2, 3, and 4) individually to a given point in the image to yield $|R_k|$ and if $|R_k| > |R_j|$ for all $j \neq k$ than the line is in the direction of mask k. Alternatively, if we want to find all the lines in the image with a certain orientation then run the mask through the image and threshold its output to yield an easily discernible binary image.

In general, finding the edge strength and direction at a certain location in the image means finding the gradient vector and from this determining the magnitude and angle respectively; the edge is perpendicular to the direction of the gradient vector. To detect edges one applies either a 1D-mask (vector) or a 2D-mask. 2D-masks include the 2X2 Roberts cross-gradient operator, or the more accurate 3X3 operators- Prewitt (mask coefficients: -1, 0, 1) and improved noise-suppression (smoothing) Sobel (mask coefficients: -2, -1, 0, 1, 2). The mask coefficients of the Prewitt and Sobel operators differ for diagonal edge detection compared with horizontal and vertical edge detection. After finding the gradient in the x and y directions the magnitude is calculated through the method of squares and square roots or the less computationally burdensome method of absolute values. The downside of using absolute values for the gradient magnitude image (i.e. edge map) is that it is not invariant to rotation in general. Improving the gradient images is achieved by first smoothing the image so only the principle edges are highlighted. After smoothing, thresholding is used to sharpen the lines and improve connectivity. The Marr-Hildreth edge detector incorporates more sophisticated analysis which requires the use of operators of different sizes, since they believed that intensity changes were not independent of image scale. The Marr-Hildreth algorithm consists of convolving the Laplacian of a Gaussian (LoG) with an input image and finding the zero crossings to determine the locations of edges. An important consequence of using zero crossings is that the resulting edges are 1 pixel thick; this property simplifies subsequent stages of processing.
such as edge linking. The LoG is a Laplacian operator (2nd derivative, which is isotropic) applied to a blurring (smoothing) 2D Gaussian function. If the accuracy of the zero crossings is inadequate a technique proposed by Huertas and Medioni (1986) allows subpixel accuracy. The more complex Canny edge detection algorithm is used when increased performance is necessary, such as requiring thin edges with no breaks in it.

4.1.2 Edge Linking and Boundary Detection

Edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries. The three processing approaches commonly used for linking edges are local, regional, and global. Local processing establishes similarity of edge pixels in a small neighborhood using the properties of strength (magnitude) and direction of the gradient vector. If two pixels within a neighborhood satisfy both magnitude and direction criteria then they are linked, and as the neighborhood is moved throughout the image a record must be kept of linked points (often a different intensity value is assigned to each set of linked edge pixels to keep track of them). Regional processing often uses a functional approximation, where a 2D curve is fit to the known points. Given a set of points which represents an open curve, the end points will represent the vertices of the polygon and are connected with a straight line. Then, we compute the perpendicular distance from all other points in the curve to this line and select the line that yielded the maximum distance. If this distance exceeds a specified threshold, the corresponding point is declared a vertex. In local and regional processing it was at least partially known if pixels belonged to individual objects. On the other hand, in global processing this knowledge is not known and all pixels are candidates for linking and thus have to be accepted or eliminated based on predefined global properties. The most simplistic approach, which is computationally prohibitive, is to find all lines determined by every pair of points and then find all subsets of points that are close to particular lines. Using the alternative HT approach, a parameter space is composed of many intersecting lines which are connecting two points. The principle lines in that plane could be found by identifying all points in parameter space where large numbers of parameter-space lines intersect.

The methods used to segment an image are edge detection, thresholding, region growing, region splitting, morphology, and motion cues. Using more than one method is generally preferred, such as combining edge detection with thresholding. Edge detection requires post processing, such as edge linking, whereas global thresholding offers speed. Region growing is a procedure that groups pixels or subregions into larger regions using seed points and the growth is based on a predefined criteria. Region splitting and merging is much different than growing in that it subdivides an image into a set of arbitrary, disjoint regions and then either merges and/or splits the regions in an attempt to satisfy the conditions of segmentation. Starting with the entire image (or a region within the image) we subdivide it based on a selected predicate being either true or false. If the predicate is false for that region of examination we divide it into four disjoint quadrants, giving a quadtree representation. The images corresponding to the nodes of a quadtree sometimes are called quadregions or quadimages. Finally, the adjacent regions are merged if the predicate of the union of these two images is true. It is customary to specify a a minimum quadregion size beyond which no further splitting is carried out.
Morphological watersheds embody many of the properties already discussed often producing more stable results, including connected segmentation boundaries. This concept is based on visualizing an image in three dimensions: two spatial coordinates versus intensity. This topographical interpretation then has three types of points: (a) points belonging to a regional minimum, (b) points where if a drop of water was placed, it would fall with certainty to a single minimum (point in the region referred to as the catchment basin or watershed), and (c) points where the water would be equally likely to fall to more than one such minimum (these points form crest lines on the topographic surface and are termed divide lines or watershed lines). The next step is dam construction performed by applying watershed segmentation algorithms. Imagine that holes are punched into the bottom of the catchment basins and water is allowed to flood in. The water will eventually spill from one basin to another and a dam must be built between these basins to prevent this from happening.

Motion is a powerful cue used by humans and many other animals to extract objects or regions of interest from a background of irrelevant detail. In particular, for autonomous driving, motion is occurring all the time from the movement of the car, the surroundings, or both. One of the simplest approaches for detecting changes between two image frames is to compare the two images (pixel by pixel). By subtracting the two images, the stationary elements will cancel leaving only the moving elements. Accumulative difference images are achieved by comparing a sequence of image frames and incrementing a counter every time a pixel at a certain location changes intensity from each subsequent image. Frequency domain techniques are also an option for detecting motion by using Fourier transforms.

There are countless approaches to creating specific types of algorithms but a few of the main ones are: decision tree, association learning rule, clustering, and bayesian networks. The decision tree maps an observation about an item to a conclusion about the item's targeted value. The association learning rule is for discovering interesting relationships between variables in a large database. Clustering is the assignment of a set of observations into subsets or clusters based on similarity. Bayesian networks is a statistical approach that updates the probability of a hypothesis as more evidence is acquired.

Two basic questions arise: what is the criterion one wants to optimize and is there an efficient algorithm for carrying out the optimization? After finding an attractive criterion an effective algorithm must be sought to find the criterion's minimum. Greedy or gradient descent type approaches fail to find global optima for high-dimensional, non-linear problems. In the 1980s, the high-dimensional problems became solvable with the introduction of Markov random fields (MRF) and variational formulations. MRF, similar to Bayesian networks, allowed future events to be predicted from present events in which chains of states are formed across two or more dimensions.

4.2 Representation and Description

After segmenting the image into regions, or super-pixels, representing a region involves two choices: in terms of external characteristics (i.e. boundary) or internal characteristics (i.e. the pixels comprising the region). Next, describing the region follows from the chosen representation. Some features, or descriptors, that can be
used to describe a boundary are length, orientation of the straight line joining its extreme points, and number of concavities. Often a connected sequence of straight-line segments, or (Freeman) chain codes, are used to represent a boundary as a simplified version of itself. This approach resamples the boundary, as it is traversed, by assigning the region boundary to the closest grid node. The set of node points are then connected with straight line segments which correspond to the numbers 0 to 3, for 4-code, or 0 to 7, for 8-code, depending on the orientation of the segment (90° or 45° increments respectively). A digital boundary can also be approximated by a minimum-perimeter polygon (MPP), an approach that encloses the boundary by a set of concatenated cells, known as a cellular complex. Thinking of the boundary as a shrinking rubber band within the cells, the final boundary shape is straight edged and constrained by the cells inner and outer walls. Recognizing that only convex vertices of the inner wall and concave vertices of the outer wall can be vertices of the MPP, our algorithm needs to focus attention on only these vertices. Another technique used to simplify a boundary is segment splitting. This method subdivides a segment, or region, successively into two parts until a specified criterion is satisfied. This method seeks inflection points by first drawing a line connecting the shape's two extreme points and then finding the longest perpendicular line extending from this centroid line and the boundary edge. This perpendicular line is obtained on both the top and bottom of the centroid line to find the vertices of the polygon. If the perpendicular line exceeds some threshold value further splitting occurs. Another variation of this method is to plot the distance from the centroid to the boundary as a function of angle (0-360°) yielding a signature, or 1D functional representation. Scaling of the image will then result in a change in the amplitude values of the corresponding signature. One way to normalize the signature is to scale all functions so that they span the range of 0 to 1. However, this simple method of normalization, which depends on only two values (minimum and maximum), can be a significant source of error if the shape is noisy. Yet, distance versus angle is not the only way to generate a signature. Another method, called the slope density function, creates a histogram of tangent-angle values. The way to interpret the histogram is deep valleys represent points of rapidly varying angles (i.e. corners or other sharp inflections), whereas, the peaks are sections of the boundary with constant tangent angles (i.e. straight or nearly straight segments).

Decomposing a boundary into segments is often useful to simplify the description process for irregularly shaped objects that contain one or more significant concavities. Following the contour of an object (set S) and marking the points at which a transition is made into or out of the convex deficiency (D) yields points which delineate the ends of the boundary segments that comprise the region boundary. Another method of representing the structural shape of a region is to obtain a skeleton of the region via a thinning (i.e. skeletonizing) algorithm. The medial axis transformation (MAT) is a line(s) drawn through the region which has more than one border point (i.e. border points on opposing side of the MAT line). This method creates a representation of the region that is simplified and visually pleasing. However, the drawback of the skeleton representation is that it is expensive computationally, since implementation potentially involves calculating the distance from every region point (i.e. interior point) to every point on the boundary.
4.3 Object Recognition

The result of segmenting, representing, and describing an image is an image composed of regions or artifacts which are more meaningful, and these individual regions are easier to recognize as objects or patterns. The approaches to recognizing patterns are centered around learning techniques and are classified as either decision-theoretic (i.e. quantitative descriptors such as length, area, or texture) or structural (i.e. relational descriptors for recursive relationships involving primitive elements). The key concept to keep in mind for the decision-theoretic (i.e. vector approach) is that selecting the descriptors on which to base each component of a pattern vector has a profound influence on the eventual performance of object recognition. For quantitative descriptions vectors are used to arrange patterns, whereas, for structural descriptions strings and trees are used. Pattern recognition refers to assigning patterns to their respective classes automatically and with as little human intervention as possible. Class membership is often a function of not only quantitative measures about each feature but also the spatial relationships between the features. Primitive elements that repeat can be sampled and expressed in terms of a pattern vector; however, if primitive elements are based on relatively simple connectivity than string descriptions are a better approach to adequately generating patterns of objects. Lastly, the most powerful descriptions- tree descriptions, are used for most hierarchical ordering schemes.

Decision-theoretic approaches to recognition are based on the use of decision (or discrimination) functions and matching an unknown pattern to the closest class (i.e. prototype pattern vector). The minimum distance classifier computes the distance between the unknown pattern vector and each of the prototype vectors. The prototype vector that is the smallest (Euclidean) distance away from the unknown vector in question is chosen for the decision. Additionally, spatial correlation performed using a template can be used for matching. The places with the maximum normalized correlation coefficient highlight where the best match occurs. High classification speeds can be achieved with analog circuits composed of resistor banks.

Taking a probabilistic approach to pattern recognition is often beneficial due to the randomness in pattern classes that are normally generated. Classification using the optimum statistical classifier approach is optimal in the sense that, on average, its use results in the lowest probability of committing a classification error. The probability that a particular pattern comes from a certain class is both a function of the probability density function of the patterns from that class and the probability of occurrence of the class a priori (note: all classes may be equally likely to occur). The conditional average risk (or conditional average loss in decision theory terminology) is used to determine the average loss in assigning a pattern to a certain class when in fact it is not from that class, and there are many other classes from which it could come from. The classifier that minimizes the total average loss is called the Bayes classifier. The loss for a correct decision is generally assigned a value of 0 and the loss for any incorrect decision usually is assigned the same nonzero value (of say 1).

Sample patterns are used to estimate statistical parameters of each pattern class. These sample patterns are called training programs and a set of such patterns from each class is called a training set. The process by which a training set is used to obtain decision functions is called learning or training. The minimum distance classifier is specified by the mean vector of each class. The Bayes classifier for Gaussian populations is specified by the
mean vector and covariance matrix of each class.\textsuperscript{9} The following models that use a multitude of elemental nonlinear computing elements (called neurons) which are organized as networks (much like how our own brain works) are: neural networks (aka neural nets), neurocomputers, parallel distributed processing (PDP) models, neuromorphic systems, layered self-adaptive networks, and connectionist models.\textsuperscript{9} 

Interest in neural networks began in the early 1940s by McCulloch and Pitts (1943).\textsuperscript{9} Stochastic algorithms are used on training sets of patterns, nonetheless, improved statistic properties are achieved via learning through the use of neural systems that adaptively develop the coefficients of decision functions.\textsuperscript{9} The mathematical models are generated via this concept of learning by reinforcement or association (Hebb 1949).\textsuperscript{9} The neurons suddenly change binary states upon learning.\textsuperscript{9} Learning machines, called perceptrons, which appeared in the mid-1950s and early 1960s (by Rosenblatt 1959, 1962), used mathematical proofs to show that linearly separable training sets (separable by a hyperplane) would converge to a solution in a finite number of iterative steps.\textsuperscript{9} The solution took the form of coefficients of hyperplanes capable of correctly separating the classes represented by patterns of the training set.\textsuperscript{9} Although perceptrons proved to be a well-founded theoretic model, unfortunately, the training algorithms were inadequate for most pattern recognition tasks of practical significance.\textsuperscript{9} More recent research, by Rumelhart, Hinton, and Williams (1986), dealing with the development of new training algorithms for multilayer perceptrons have changed the outlook for perceptrons considerably.\textsuperscript{9} Their method, called the generalized delta rule for learning by back-propagation, has established multilayer perceptron-like machines as one of the principal models of neural networks currently in use.\textsuperscript{9} 

The perceptron learns a linear decision function that dichotomizes two linearly separable training sets.\textsuperscript{9} The linear decision function uses weights to modify the components of the pattern vectors; the weights are analogous to synapses in the human neural system.\textsuperscript{9} The function is the sum of all the weighted pattern components and is referred to as the activation function.\textsuperscript{9} If the function is more than 0 the perceptron output value is thresholded to 1 indicating that the pattern was recognized as belonging to that class.\textsuperscript{9} The orientation of the hyperplane is established by the the weights of the function, with the exception of the last weight which is proportional to the perpendicular distance from the origin to the hyperplane.\textsuperscript{9} 

A simple iterative algorithm for obtaining a solution weight vector for two linearly separable training sets is the fixed increment correction rule.\textsuperscript{9} This method employs a correction increment if the pattern is misclassified.\textsuperscript{9} Convergence of the algorithm occurs in a finite number of steps (assuming the two or more classes are separable) when the entire training set for the classes is cycled through the machine without errors.\textsuperscript{9} In practice, linearly separable pattern classes are the exception and for this reason a significant amount of research effort in the 1960s and 1970s went into developing techniques designed to handle non-separable pattern classes.\textsuperscript{9} However, many of these advances in handling non-separable pattern classes have become obsolete due to the advances in training of neural networks.\textsuperscript{9} 

One of the early methods used for training perceptrons, that is still relevant, is the original delta rule, also known as the Widrow-Hoff or least-mean square (LMS) delta rule.\textsuperscript{9} This method minimizes the mean square error between the patterns (i.e. actual and desired response) at any training step.\textsuperscript{9} When the pattern classes are
separable, the solution given by the delta correction algorithm may or may not produce a separating hyperplane, nonetheless, this method does converge under both separable and non-separable cases. The methods discussed so far can be extended to more than two classes and to nonlinear decisions.

Multilayer feedforward neural networks consist of layers of perceptron computing elements in which the output of every neuron in one layer feeds into the input of every neuron in the next layer. The number of neurons in the input layer is often the dimensionality of the pattern vectors and the number of neurons in the output layer is the number of of pattern classes that the neural network has been trained to recognize. The number of neurons in an intermediate layer can be chosen heuristically, an average of the first and last layers, or arbitrary and then refined by testing. If a pattern vector belongs to a class than the output of the network for that class is “high” while all other outputs are “low”. There is the option to replace the hard-limiting activation function with a soft-limiting “sigmoid” function, additionally, different types of functions could be used for different layers or even for different nodes within the same layer of a neural network. Training by backpropagation develops a rule, similar to the delta rule, that allows adjustment of the weights in each of the layers in a way that seeks a minimum to an error function. Starting with the error at the output layer, which yields an error term and weights, this method propagates the error back into the network. The process starts with an arbitrary (but not all equal) set of weights throughout the network. Then by applying the generalized delta rule at any iterative step involves two basic phases. In the first phase, a training vector is presented to the network and is allowed to propagate through the layers to compute the output for each node. Then, the outputs of the nodes in the output layer are then compared against their desired responses, to generate the error terms. In the second phase, a backward pass through the network occurs during which the appropriate error signal is passed to each node and and the corresponding weight changes are made (taking into account any bias weights that may have also been present). Common practice is to track the network error, as well as errors associated with individual patterns. In a successful training session, the network error decreases with the number of iterations and there is convergence to a stable set of weights that exhibit only small fluctuations with additional training. If a pattern has been classified correctly during training the response of the node in the output layer associated with the pattern class from which the pattern was obtained should be high (≥ 0.95), while all other nodes should have outputs that are low (≤ 0.05). After the system has been trained the feedback paths are disconnected and the patterns are classified based on the parameters established during training. The input pattern is allowed to propagate through the various layers and is classified as belonging to the class of the output node that was highest. If more than one output node is high (or if all the nodes are low) either chose the highest valued node or declare a misclassification. When the system is said to have learned the shapes from all the classes tested, due to each training pattern yielding a output layer node which is high and all other nodes low, the next stage of training occurs. The second part of training is a similar process except the patterns are now noisy samples. By running progressively noisier test sample images for each class being tested the weight factors are adjusted and the improvement in performance is seen (which is measured by the decrease in the probability of misclassification). The larger the number of samples there is the better learning from each (noisier) test set.
When the nature of the noise is known, small incremental additions of noise to the test set is possible, and this is ideal for improving the convergence and stability properties of a neural network during learning. A single-layer perceptron implements a hyperplane decision surface, whereas, a multilayer network is capable of implementing complex decision surfaces composed of intersecting hyperplanes. Let's first consider a two-input, two-layer network. The outputs of the two nodes, in the first layer, are either high or low (which is denoted by a 1 or 0 respectively). The two nodes in the first layer feed a single node in the second layer and therefore the possible combinations feeding each of the second layer nodes is either (1,1), (1,0), (0,1), and (0,1). The output nodes from the first layer are able to classify any input pattern as belonging to one of the two regions (or classes) by performing a logical AND operation between the two nodes in the first layer. The output node in the second layer responds with a 1 only when both outputs from the first layer feeding it are 1 as a function of the AND logic performed. Finally, lines delineating the two separate regions (which may break up the classes) are drawn by having the node lie on the positive side of the line if it is high or on the negative side if it is low (note: a single linear surface may not be sufficient, rather, a line with a vertex (i.e. two lines connected) or two separate lines may be necessary). If the number of nodes in the first layer were increased to three, the network would implement a decision boundary consisting of the intersection of three lines. The next logical step is to increase the number of layers to three. In this case the nodes of the first layer implement lines, the nodes of the second layer then perform AND operations in order to form regions from the various lines, and finally the nodes of the third layer assign class membership to the various regions.

Patterns so far have been dealt with quantitatively, largely ignoring any structural relationships inherent in a pattern's shape. Strings are the most practical approach in structural pattern recognition. A procedure analogous to the minimum distance concept for pattern vectors can be formulated for the comparison of region boundaries that are described in terms of shape numbers. String matching is another option if the region boundaries being compared are coded into strings. Based on the number of matches between the strings one can quantify their similarity.

The next step beyond extracting attributes from images and employing object recognition is image analysis methods whose proper development requires concepts from machine intelligence. Machine intelligence and areas that depend on it, such as scene analysis and computer vision, still are in their early stages of practical development. Today, image analysis problems are characterized by heuristic approaches, that is, learning to arrive at an approximate solution, as opposed to, an exact closed form solution.
5. References


5) Sappa, A., Herrero, R., Dornaika, F., et. al.. "Road Approximation in Euclidean and v-Disparity Space: A Comparative Study." *Computer Vision Center, Edifici O Campus UAB.*


6. Figures

![Image of Automatic Lateral Control System]

Figure 1: The Two Steps of Automatic Lateral Control, Ref. 6