8-2011

Stereoscopic vision in vehicle navigation.

Behnoush Abdollahi 1986-
University of Louisville

STEREOSCOPIC VISION IN VEHICLE NAVIGATION

By

Behnoush Abdollahi

A Thesis
Submitted to the faculty of the
J.B. Speed School of Engineering
University of Louisville
In Partial Fulfillment of the Requirements
for the Degree of

Master of Science

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

August, 2011

Stereoscopic Vision in Vehicle Navigation

By

Behnoush Abdollahi

A Thesis Approved on

August 5, 2011

by the following Thesis Committee:

____

Advisor - Ming Ouyang, Ph.D.

____

Olfa Nasraoui, Ph.D.

____

Ayman S El-Baz, Ph.D.

ACKNOWLEDGMENTS

Researching and developing a system to assist vehicle navigation is an interesting journey, full of struggles, and a challenging task. The goals of this project would not be achieved without the help of many people. I would like to thank Dr. Ming Ouyang, my thesis adviser, who guided me throughout the project. Without his help and numerous inspirational guides, this research work could not be reached this far. Drs. Frigui, Nasraoui, Kantardzic, Desoky provided a lot of knowledge for my work.

I also would like to thank other committee member, Dr. El-Baz for serving in my committee, and for his inputs and feedbacks.

Last but not least, I thank all my family members who have supported me and encouraged me, and their inspirational sayings have been a motivation for me to keep good work. I dedicate this work to all the people I love.

ABSTRACT

STEREOSCOPIC VISION IN VEHICLE NAVIGATION

Behnoush Abdollahi

August 5, 2011

Traffic sign (TS) detection and tracking is one of the main tasks of an autonomous vehicle which is addressed in the field of computer vision. An autonomous vehicle must have vision based recognition of the road to follow the rules like every other vehicle on the road. Besides, TS detection and tracking can be used to give feedbacks to the driver. This can significantly increase safety in making driving decisions. For a successful TS detection and tracking changes in weather and lighting conditions should be considered. Also, the camera is in motion, which results in image distortion and motion blur. In this work a fast and robust method is proposed for tracking the stop signs in videos taken with stereoscopic cameras that are mounted on the car. Using camera parameters and the detected sign, the distance between the stop sign and the vehicle is calculated. This calculated distance can be widely used in building visual driver-assistance systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS AND TABLES

1 INTRODUCTION

1.1 Motivation

Recently, in the area of computer vision, detecting and tracking traffic signs from videos have become the center of interest. In driving environments, signs warn the driver, regulate the traffic and represent for commands or prohibitions; so building a system that automatically detects and tracks these signs has great impact on road safety. However, it is a challenging problem because it is difficult to track signs in videos captured from a vehicle that has non-uniform motion. Also, there is change in lighting or weather that is inherent in outdoor videos. The signs are not always perpendicular to the camera axis; they may be rotated and distorted. The vision system must be robust to illumination and pose challenges. Many algorithms can not handle the challenges such as pose and illumination changes and their performance deteriorates, especially when the camera is not fixed. Recently, to address these problems, the Scale Invariant Feature Transform (SIFT) (1) was used in addition to color and shape characteristics. SIFT features are invariant to scaling, rotation, or translation of the object of interest in the image.

A real-time and robust automatic traffic sign detecting and tracking system can support and help the driver in making driving decisions. For instance, it can remind the driver of the current speed limit, warn him of the upcoming stop sign, and give him feedback about whether to decrease or increase his speed. These systems are built to significantly increase driving safety and comfort.

In this thesis, a driver-assistance system is built that tracks stop signs in videos captured by cameras mounted on a moving vehicle. Videos are taken with two cameras so that stereoscopic characteristics of the detected images are employed to estimate the distance between the vehicle and the stop sign and calculate the current speed of the vehicle to provide feedback to the driver.

For this work, both color and SIFT features are used in detecting and tracking the stop sign. In the tracker part, the mean shift algorithm is implemented, which has shown more promising performance in comparison with other types of real time tracking methods. This algorithm gives a better result in circumstances against occlusions and clutter and can be used in a fast and real time system.

1.2 Related work

Detection and tracking of traffic signs are among the main tasks in building a visual driver-assistance system. Recently, many techniques have been developed in this area. (2) detects the traffic sign using color and shape features. Detection is done based on the developed neural networks to extract features. In their work tracking is done using a Kalman filter. Unfortunately, they can not track the traffic sign without knowing the speed of the vehicle in advance. They need to know the speed to estimate the size of the traffic sign in the next frame. (3) uses joint modeling of color and shape information for detection. (4) uses only color features to detect traffic signs. Gaussian distributions that model each color are used for detecting road and traffic signs.

A more sophisticated method of sign detection based on genetic algorithms is proposed in (5). (6) uses SIFT features in the detection part only and not in the tracking part. (7) detects traffic signs using circular and triangular shapes.

Actually, whether to use color features in developing a traffic sign detection system is an important consideration. In some implementations (8), (9), (10), (11) the morphological characteristics of the signs such as symmetry or shape templates are used. Some systems are mainly based on border detection like in (12), (13), (14). These methods use pyramidal structures.

In color based traffic sign detection, different color spaces are used in different implementations. (15), (16), and (17) use popular RGB color-space. RGB is susceptible to lighting changes. Thus, other color spaces like HIS (18), (19), (20), (21), (22), (23), and (24) or LUV (25) are preferred.

Among the most tracking algorithms used for traffic signs are Kalman filters, Kalman-Bucy, template matching based, various types of neural networks, and Bayesian generative modeling.

2 LITERATURE SURVEY

2.1 Feature extraction

Feature extraction is the first and a crucial step in developing a vision based application. The choice of features has great effects in the performance of the system and the final result achieved by the method. The following issues should be considered when choosing a proper set of features:

The features should carry enough information about the image.

They should not require any domain-specific knowledge for their extraction.

They should be easy to compute in order to be feasible for methods incorporating a large image collection.

Color feature is one of the most commonly used features in image processing algorithms. Texture, shape, and Scale Invariant Feature Transform (SIFT) are among other widely used features. In the following, SIFT will be described in more details since it is used in the development of this project.

2.1.1 SIFT feature

In this project the Scale Invariant Feature Transform (SIFT) algorithm developed by Lowe (2004) is used (1). This algorithm has some significant characteristics that have been helpful in the present object tracking system. SIFT features are reasonably invariant to changes in illumination, scaling, rotation, and to some extent skewness of the object of interest. In videos of outdoors, all these changes are common.

The SIFT features are a set of orientation histograms on 4×4 pixel neighborhoods. A histogram consists of 8 bins each, and each descriptor contains an array of 4 histograms around the key-point. This leads to a SIFT feature vector with $4 \times 4 \times 8 = 128$ elements.

The SIFT algorithm for feature detection is based on the detection of extrema in scale-space, localization of key-points, assignment of orientation, and generation of descriptors.

Detection of extrema in scale space: The image is transformed to different scales and Difference-of-Gaussian (DoG) at these scales is calculated to obtain the extrema.

Key-points: Local maxima or minima of the DoG images calculated across different scales are identified as key-points. A pixel is selected as a candidate key-point if it is local extrema with respect to its 8 neighbors at the same scale and 9 corresponding neighbors at neighboring scales.

Orientation: The gradient orientation histograms are used in determining the orientation of the key points. The gradient magnitude of each neighboring pixel has a weight in calculating the orientation. Additionally a Gaussian window is used to calculate the orientation of these histograms. The feature descriptor computed is shown in Figure 1.

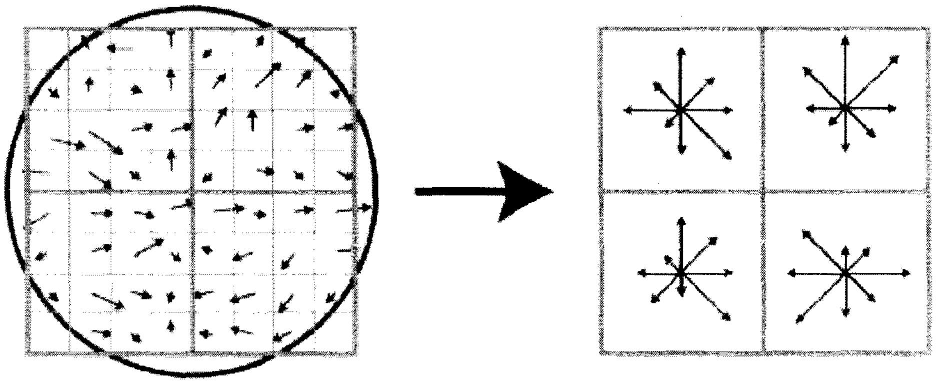


Figure 1 SIFT descriptor representation

To find traffic signs, the calculated SIFT features should be compared with the model features. This is done using the matching process as proposed by Lowe (2004), which calculates the Euclidean distance between features.

To improve the performance of the system, first the areas with some probability of containing a stop sign are filtered out and then SIFT features for candidates are calculated. This results in faster detection and tracking of the stop sign in the videos.

2.2 Object detection

Object detection in the field of computer vision and image processing relates to methods that deal with detecting instances of objects of a certain class in digital images and videos. In this project these methods are employed to detect stop signs in videos taken from a moving vehicle.

There are two main categories of object detection methods in videos; they are feature-based and template-based (33).

2.2.1 Feature-based object detection

In feature-based object detection, one or more features are extracted from the image, and object detection and recognition become the matching between the features extracted and the model of the object of interest in terms of these features. In this type of methods standardization of image features is required, and also transformation to another space may be required to handle changes such as illumination or orientation in the image.

2.2.1.1 Shape-based approaches

In this type of algorithms the image is preprocessed based on the application. For each type of objects, such as persons and flowers, different preprocessing and filtering algorithms may be required. Filter algorithms may also be required for noise removal and transforming the image to handle challenges related to scale and rotation.

After preprocessing, the segmentation of the image and extracting the object of the interest are required. Following this step, the edges and boundaries of the detected object need to be found using edge detection and boundary-following techniques. In the presence of occlusions and illumination in the more complex scenes, object detection using these methods becomes difficult (26), (27).

One of the simplest segmentation methods to extract the object of interest is thresholding (37).

- **Thresholding**

In global thresholding a threshold value is selected, and then the image is transformed into a binary image based on that value. The choice of this value is important, and different values will lead to different results. A threshold value can be chosen by the user or a thresholding algorithm can be used to automatically compute the value.

A simple algorithm is to choose the mean or median as the threshold value. This works well in noiseless images with uniform distribution of colors of the background and the objects.

Using global thresholding the image is segmented into a binary image of object pixels and background pixels.

$G1 = \{f(m,n): f(m,n)>T\}$ (object pixels)

$G2 = \{f(m,n): f(m,n)\leq T\}$ (background pixels; $f(m,n)$ is the value of the pixel located in the m-th column, n-th row)

2.2.1.2 Color-based approaches

Color is among the most commonly used features in image processing techniques. It is invariant to many changes like rotation and scale. It can be easily acquired and algorithms using this feature usually have low computational cost. Although color-based approaches may not work properly in images with changes in illumination, color is still a desirable feature when appropriate.

In applications based on color, the model for the color histogram of the objects in the image is estimated. This is one of the simple tracking methods in which the model obtained is tracked from one frame to the next. For this type of method to be applicable, the image needs to be segmented into background and objects. It is based on tracking the regions of similar normalized colors. These regions are relative in positions and have a fixed size in consecutive frames. The collection of pixels from these regions are sampled and used as a color vector in tracking. These methods are robust enough to handle occlusion to some extent.

2.2.2 Matching-based object detection

Matching is a classifying technique used in digital image processing for comparing portions of images against a template. A sample image may be used as a template for the recognition of similar objects.

There are two main groups of matching-based object detection methods: feature-based and template-based matching.

2.2.2.1 Feature-based matching

The feature-based approach is usually considered if the image has strong features, such as color, edges, and corners. In this approach features from both the template and the reference image are used in comparing and finding the best match. This approach is useful when the template image may be transformed in the reference image. This approach also has good performance because in comparing the images, only features are considered, which are usually significantly fewer than the actual number of points.

2.2.2.2 Template-based matching

In the template-based approach, the entire template is considered. The matching method is to compare the template image with all or some test places in the reference image, and the best location that matches the template is obtained. In this approach, a whole portion of points in the template image is considered as a search window, which is slid over the reference image. Thus this method has high computational cost, which can usually be improved by reducing the resolution of the images.

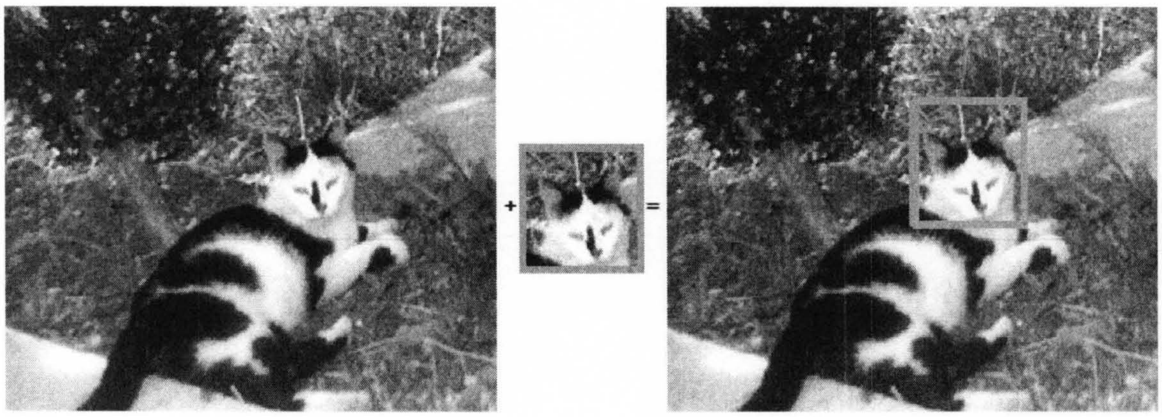


Figure 2 Template matching

The template-based matching can be further classified as fixed or deformable.

1) Fixed template matching

Fixed templates can be used when shape, size and the scale of the object do not change in consecutive frames. Techniques used in fixed template matching are image subtraction and correlation (20).

- **Image subtraction**

In this technique, the distance function between the template and different positions in the image is minimized to find the position of the object in the image. This subtracting technique performs well in environments where there is no deformation in the object, and the imaging condition such as image intensity is fixed. In these restricted environments, this method is preferable because it requires less computation time.

- **Correlation**

In this method, the peak of the normalized cross-correlation between the template and the image is obtained to locate the object of interest in the image. Although this method has high computational cost, it is immune to noise and illumination changes. By using a small set of carefully chosen points, the amount of computation can be reduced.

2) Deformable template matching

In many cases, the object of interest may be deformed, and its shape and color may vary. Since in most videos, the position of the object relative to the cameras changes a lot, and objects become deformed in the images, deformable matching approaches are more appealing in most tracking problems.

In this technique, a prototype contour of the object is created that represents the shape of the object. To fit the template to salient edges in the image, a probabilistic transformation on this prototype is applied. Then, the cost of such transformation that changes the shape and size of the template is formulated in an objective function with transformation parameters. By updating the transformation parameters iteratively, this objective function is minimized to match the object.

2.2.3 Learning-based object detection

Object detection can also be performed using a supervised learning mechanism like in (38). In this type of methods, the object is learned automatically from a set of examples. There is no need to use templates anymore, because supervised learning methods generate a model from the set of training images. This model acts like a function and maps inputs to outputs. The problem of object detection can be viewed as a classification problem. In classification, the learner approximates the behavior of a function that generates an output in the form of a continuous or discrete value, which is treated as a class label. In the context of object detection, the process of learning is performed based on the object features, and the output of the classification function would be the associated object class that is defined manually for each object in the learning phase.

The selection of features becomes important when using supervised learning. Features should be selected in such a way that they properly discriminate classes from each other. Features used can be color, edges, object area, object orientation, and object appearance in the form of a density function like a histogram.

The next step is choosing a learning approach to learn the object from the features. Some commonly used learning methods are neural networks, adaptive boosting, decision trees, and support vector machines.

One of the main difficulties in using supervised learning methods is that a large collection of samples from each object class is required. Additionally, all the samples in the collection must be manually labeled. An approach that can be used to reduce the amount of manually labeled data is to use the co-training method: First, two classifiers are built using a small set of labeled data. Next, each classifier is used to assign unlabeled data to the training set of the other classifier. This co-training method has shown to provide accurate results using a small amount of labeled data.

2.3 Object tracking

Object tracking is an important task within the field of computer vision. One of the main steps in video analysis is object tracking. After detecting the interesting object, tracking is performed from frame to frame, and it is used in recognizing the behavior of the object in the video.

Vehicle related tracking helps a lot in obstacle avoidance, path planning, and traffic sign tracking. These functionalities are used in autonomous vehicles, and they can help the driver in making driving decisions and navigation.

Tracking of an object in a video is to continuously determine the position of the object in images. Many algorithms have been established to solve this problem. (28)

presented a Kalman filter-based method for tracking. Some other major tracking methods are kernel based tracking and optical flow-based tracking (29), (30), (31).

Recently, the mean shift algorithm is proposed as a tracking method, and it has more promising performance than the other mentioned method (32). This algorithm gives a better result in circumstances involving occlusions and clutter. It is very fast and can be used in a real time system.

A taxonomy of tracking methods is shown in Figure 3. The details of these methods are described in the following sections.

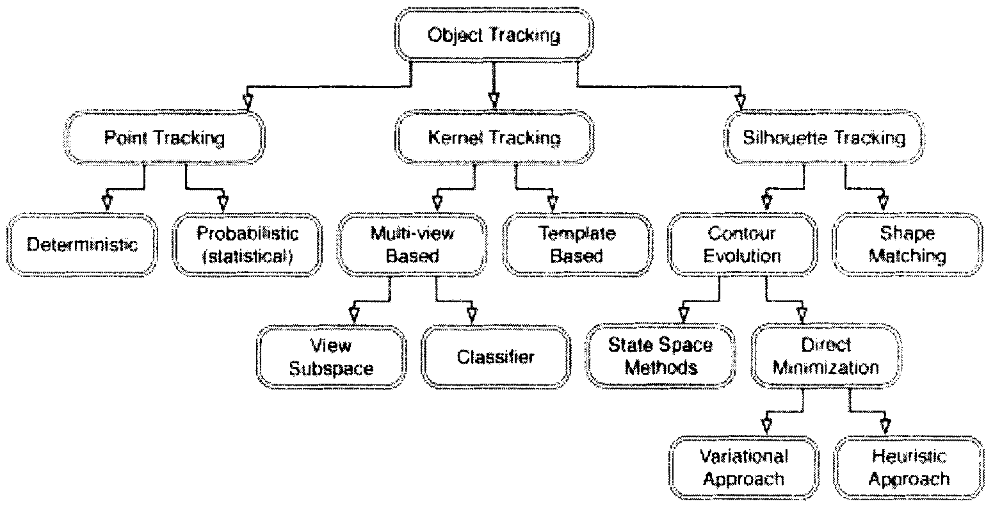


Figure 3 Taxonomy of tracking methods (33)

2.3.1 Point tracking

Tracking can be stated as a point correspondence problem across frames. Point correspondence methods can be divided into two main categories: deterministic and statistical.

1) Deterministic methods

In these methods, a cost function is defined based on associating each object in one frame to a single object in the next frame. The problem is solved by minimizing this cost function using combinatorial optimization methods. The correspondence cost is usually defined using some constraints. For example, it is assumed that the location of the object does not change notably from one frame to the next. The direction and speed of the object does not change much in a smooth motion, and the velocity of objects in a small neighborhood is similar.

One approach to solve for the correspondence problem is using a greedy approach that considers a combination of the constraints.

2) Statistical Methods

When features are extracted from video frames, they may contain noise. Moreover, the object may undergo random perturbations due to the movement of the camera. Statistical methods take these uncertainties into consideration when modeling and solving for tracking problems. The statistical correspondence methods model the object properties such as position, velocity, and acceleration. These properties are measured in each frame using object detection mechanisms.

The objective of tracking is to estimate the state of the object in the next frame given all the measurements up to that moment. This estimation can also be performed by constructing a probability density function (PDF) of the possible states of the object. There is a prediction step and a correction step. In the correction step, the posterior PDF is computed by employing the likelihood function. These two steps are sufficient if there is a single object being tracked in the scene. However, if there are multiple objects in the scene, measurements need to be associated with the corresponding object states.

2.3.1.1 Kalman filter based tracking

This tracking method is based on the Kalman filter. There is an interesting relationship between the Kalman filter and the hidden Markov model (HMM) because the sequential data on which the Kalman filter operates can be represented with an HMM.

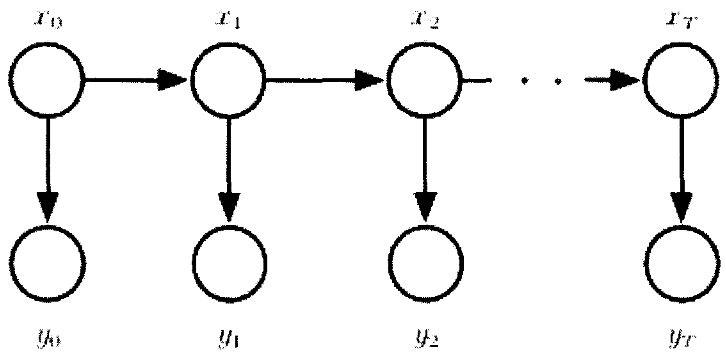


Figure 4 Hidden Markov Model (HMM)

Figure 4 shows a model of HMM. The y_i nodes represent the measurements and the output data that are observed, and x_i nodes are named hidden nodes or states.

The physics concept behind object motion relates the state nodes to each other. There are different alternatives to explain the transition from one state to the next. All these ways can be separated into two groups of linear and non-linear functions that describe the state transition.

The standard Kalman filter employs a linear transition function. Let A be the state transition matrix, and w_t be a noise term. In a Kalman filter the state transition from t to $t + 1$ is represented with:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \qquad \textbf{(Equation 1)}$$

The noise term w_t is independent of the state x_t and is considered to be a Gaussian random variable with zero mean and a covariance matrix Q, which accounts for possible changes between the states t and $t+1$ that are failed to be considered in the transition matrix.

Also the relationship between the state and the measurement needs to be modeled. Let C relate the state to the measurement, and υ_t be the noise of the measurement. Based on the assumption that the relationship is linear it is expressed as follows:

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\upsilon}_t \qquad \textbf{(Equation 2)}$$

The noise υ_t is also considered to have a normal distribution with zero mean and a covariance matrix R.

The process of tracking is modeled by making a prediction of the location of the object at each step based on the measurements. Initially y_t is observed and then y_{t+1} is predicted. After the prediction is made and the measurement is taken, the process is repeated for the next state $(t+2)$.

2.3.2 Kernel tracking

In kernel tracking, a histogram of a template shape is tracked. This refers to the shape and appearance of the object. For example, the template can be chosen to be a rectangle or an elliptical shape. Tracking is based on computing the motion of the kernel from frame to frame. This kernel can be the color histogram of the object of interest or any other feature. One of the main algorithms in this category of tracking methods is mean shift tracking.

Based on the appearance representation, there are two main categories of these tracking methods: template and density-based models and multi-view appearance models.

2.3.2.1 Template and density-based models

Templates and density-based models are relatively simple and have low computational cost. These trackers can be divided into two subcategories based on whether the objects are tracked individually or jointly.

1) Tracking single objects

A common approach in this category is template matching. Generally in template matching, the image is searched for a region similar to the object template. In tracking by template matching, the position of the template in the current image is estimated via a similarity measure. This similarity measure usually considers image intensity and can have high computational cost.

To reduce the computational cost, the search in the image can be limited to the vicinity of the object's position in the previous frame. In a simpler form of template matching, a circular or a rectangular region in the image is considered, and a color histogram or other object representations are used for tracking. The similarity between the object model and the hypothesized position is computed and the position with the highest similarity is chosen as the current location for the object. The mean-shift tracker is one of the methods that use this technique.

- **Mean shift tracking**

Among the various object tracking algorithms, mean shift tracking has recently achieved considerable popularity due to its simplicity and robustness (34).

In this tracking method, a target region is described using a color histogram calculated from the region. A similarity measure is employed to measure the similarity between the target region and a template or model region. Using the mean shift algorithm, tracking is accomplished by iteratively finding the local minima of the distance measure functions.

The most commonly used similarity measures are the Kullback-Leibler divergence, Bhattacharyya coefficient, and other information-theoretic similarity measures.

Given the sample points and the kernel function k(x), the color distribution of the object in the current image is calculated using the following kernel density estimation,

$$p_x(x,u) = \frac{1}{N}\sum_{i=1}^{N}\omega\left(\left|\frac{x-x_i}{\sigma}\right|^2\right)k\left(\left|\frac{u-u_i}{h}\right|^2\right) \qquad \textbf{(Equation 3)}$$

where x is the spatial feature and u is the color feature of the pixels. Also, σ and h are the bandwidths in the spatial and feature spaces. N is the number of sample points. The sample points are defined by x_i and u_i, where x_i is the 2D coordinates and u_i is the corresponding feature vector like color.

Similarly, we can estimate the probability density function (PDF) of the target image.

These two color distributions are compared using the Bhattacharya coefficient in order to define the distance between two histograms.

$$\rho(\mathrm{p},\mathrm{q}) = \sum_{u=1}^{m} \sqrt{p_u q_u} \qquad \textbf{(Equation 4)}$$

A large Bhattacharya value means that there is a good color match between the two regions.

In this project, a modified mean shift algorithm is implemented. In the simple mean shift algorithm, the scale of the object being tracked is fixed. The size of the tracking window is selected by choosing a proper value for h. In a video with moving objects or a moving camera, the size of the objects and the scale for tracking changes all the time so an efficient tracking algorithm should be scale adaptive. In (35), a few different scales are tried every time and the one providing the most similarities is selected as the new scale of the object in the new frame. This kernel scale is a crucial parameter to the performance of the mean shift algorithm. If it is too large, the tracking window will contain pixels not belonging to the object and the computed histogram from these collected points will be describing the background as well. This histogram will not be a good similarity measure to be used in the tracker and the tracker will be distracted.

2) Tracking multiple objects

In many tracking environments, there is interaction among multiple objects and between objects and the background that need to be considered in the tracking algorithm. For example, the whole image can be considered as having different layers, one layer for each object. Based on the object's motion model and its shape and color characteristics, the probability that a pixel belongs to a layer is computed.

Another multiple object tracking technique is joint modeling of the background and objects. The objects are modeled as cylinders, and it is assumed that the ground plane is known. Thus, the 3D positions can be computed and used in tracking. This method can be used in handling occlusion between the objects. However, the maximum number of objects is predefined.

2.3.2.2 Tracking using multi-view appearance models

Generally, tracking uses online data gathered from the most recent observations. This data is gathered in the form of histograms, templates, and any types of models. The object's appearance may change in sequences of images, and a new modeling may be needed to match the object view. Thus, if the object view changes dramatically during tracking, the current model may not work, and the tracker may lose track of the object. Tracking using a multi-view appearance model is a solution to this problem. In this method, different views of the object are learned and used in tracking.

One method in this category is learning the object using Support Vector Machine (SVM) classifier for tracking. A slight difference in this approach is that in addition to positive set of objects, a negative set of background regions and all the things that are not to be tracked is provided. This is an advantage of this approach because the knowledge about background objects is explicitly incorporated in the tracker.

2.3.3 Contour tracking

In this category of methods an initial contour is considered in the current frame. Through iterations the contour is evolved to its new position in the next frame. The process of evolving is usually performed by modeling the shape and motion of the object, or using optimization techniques in minimizing the contour energy.

2.3.4 Stereoscopic tracking

Most of the trackers currently available are designed for a single stream of data and not for stereoscopic pairs. They do not utilize the intrinsic stereoscopic constraints, which results in low efficiency. As in (36), our tracker implementation for stereoscopic road videos tracks two images simultaneously. In each iteration of tracking, the features are detected in the left image and then features are tracked from the left image to the right image as well as next left image in the video. The only possible displacement from left image to the right image is horizontal and along the scan-line, so using left image to track in the right image is quite efficient.

2.4 Camera calibration

The objective of camera calibration is to determine a set of camera parameters that describe the mapping between 3-D reference coordinates and 2-D image coordinates. These parameters are used in finding the distance of the object from the camera. In camera calibration the camera extrinsic and intrinsic parameters are computed.

For application of camera calibration in stereoscopic camera calibration, parameters need to be experimentally obtained to be used in estimating three dimensional (3D) coordinates of a point. The extrinsic parameters of a camera determine the position and the orientation of the camera with respect to the coordinate system, and the intrinsic parameters indicate the intrinsic properties of the camera such as the focal length.

Using the triangulation formula below, the 3D coordinates of a point can be calculated using stereo-pair images obtained from two cameras.

If the world and image points are represented by homogeneous vectors, the central projection is a linear transformation:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

$$x_i = f\frac{x_s}{z_s}$$

$$y_i = f\frac{y_s}{z_s}$$

(Equation 5)

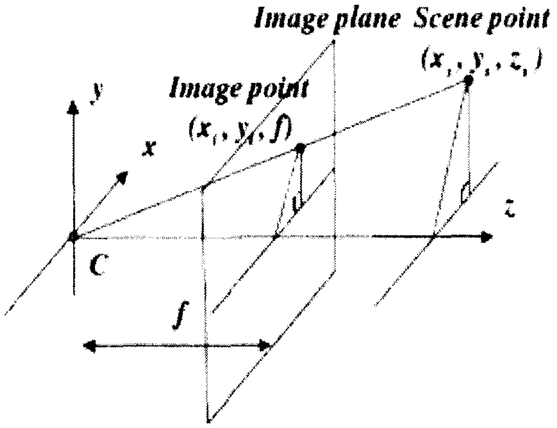


Figure 5 projection of point

3 METHODS

In this project a driver assistance system is built that gives feedback to the driver about the car's distance from the stop sign and its speed. The inputs are stereoscopic videos which require considering the characteristics of multi streams of video, in the process of object detection and tracking. This experiment is divided into two main tasks:

- Stop sign detection and tracking
- Distance computing and speed estimation

3.1 Stop sign detection and tracking

3.1.1 Object detection

The first step is object detection in stereoscopic videos of the roads and highways. The object of interest is the stop sign which is red and has octagonal shape. There are three steps in the standard technique for detecting and recognizing road signs. First, to restrict the area in the image for searching for the object, color segmentation or color thresholding is applied. This is a preprocessing phase to find possible signs in the image. Second, we need to match the object with the appropriate class for shape detection. This is done using template matching. Third, specific signs need to be detected using again template matching or using some other methods like neural networks. In this project the third step is integrated with the second step because we only consider detection of stop signs in this project rather than other types of traffic signs.

The raw data we have are stereoscopic videos in which we want to detect and track the stop signs. In our approach the detection of the stop sign is the first task that is performed on the left frame of the stereoscopic input. The video is separated into left and right streams. The detecting part of the system iteratively checks the left frames for the object of interest. Whenever a stop sign is detected, the image is given to the tracker part for the next task to be performed, which is object tracking.

Working on images requires a step of preprocessing and feature extraction. Images are usually obtained in the RGB (Red, Green, Blue) color space. In this approach images are transferred from RGB space into HSV (Hue, Saturation, Value) color space.

- The Hue component can be thought of as the actual color of the object. That is, if you looked at some object what color would it seem to you and this is the reason that HSV color space is close to human visual perception of colors.
- Saturation is a measure of purity.
- Intensity tells us how light the color is.

Processing the images in the HSV color space results in easier segmentation because we can then apply segmentation on hue channel rather than the three RGB channels. This also helps in better detection in the presence of illumination because the hue value is invariant to illumination which happens a lot in videos of outdoor scenes. Thus image regions that have color characteristics similar to the stop sign are filtered out.

Next is shape analysis to evaluate candidate regions for a stop sign. This step is performed by applying template matching. Using template matching, a stop sign is located in the image by searching for red octagonal shapes surrounding white characters of "stop." Six templates of sizes 42x42, 52x52, 62x62, 72x72, 92x92 and102x102 pixels are used to detect the position of the stop signs in the frames of the video. The stop sign in the left image is detected using the detecting part of the system. To locate the stop sign in the associated right image, the tracker part is used rather than detecting part. This results in improvement in the performance of the system. The tracker gets the image in which the stop sign is located as input and tracks the object in both the current right frame and the next left frame.

3.1.1.1 Thresholding

Stop signs can be detected by their red color. Thresholding the images based on the color in the preprocessing step expedites the process of stop sign recognition. Thresholding reduces the number of candidates later used in the SIFT feature extraction. SIFT feature extraction is performed only on the stop sign candidates instead of the whole image. This results in much less computational cost in feature extraction and later in feature matching. SIFT features of all the stop sign candidates are extracted from both the left and the right images. At this point a feature vector of a few candidates is obtained that is used in template matching. For a candidate to be selected as a stop sign, both left and right features need to be matched with the template.

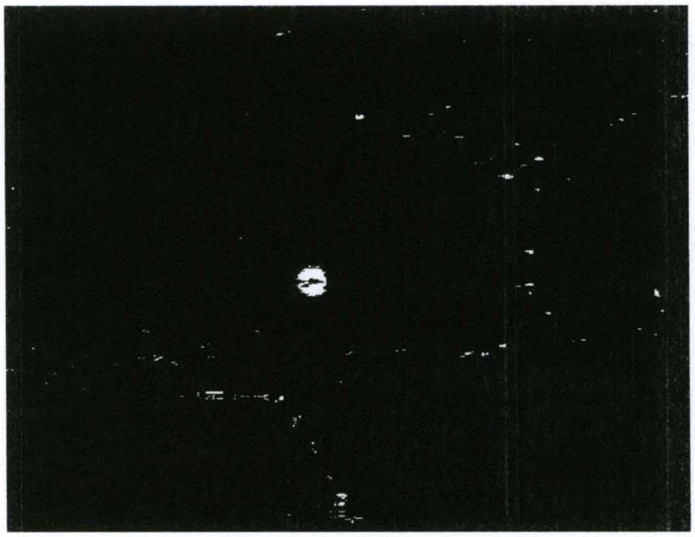


Figure 6 Thresholding to detect stop sign

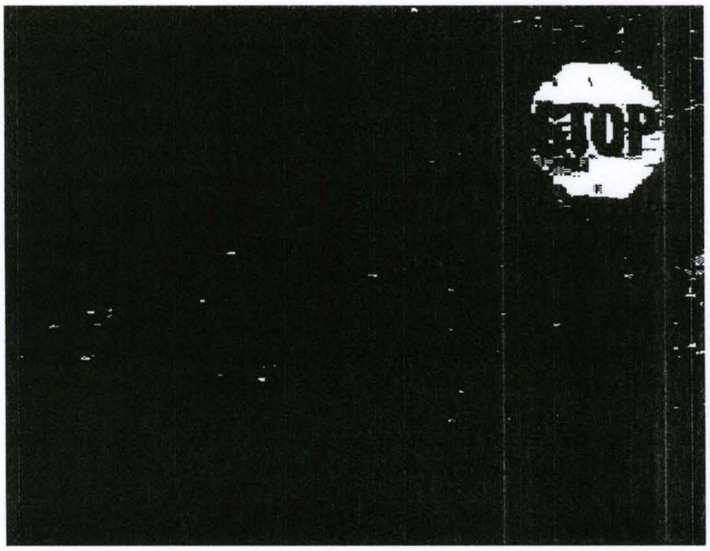


Figure 7 Thresholding result

3.1.1.2 Template matching

In template matching, six templates of sizes 42x42, 52x52, 62x62, 72x72, 92x92, and102x102 pixels are used (Figure 8). Each candidate in the left image is matched with templates, and if it successfully passes the test, the image is given as input to the tracker.

The approach used in this phase is feature based template matching. Only the color and SIFT features are compared rather than considering all the points. This improves the performance of the method. Also using feature based approach helps in detecting transformed objects in images, because a disoriented picture of stop sign in an image has the same SIFT features as a symmetric, octagonal one.

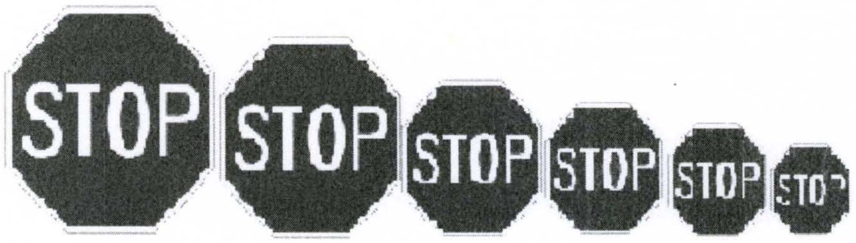


Figure 8 Templates of stop sign used

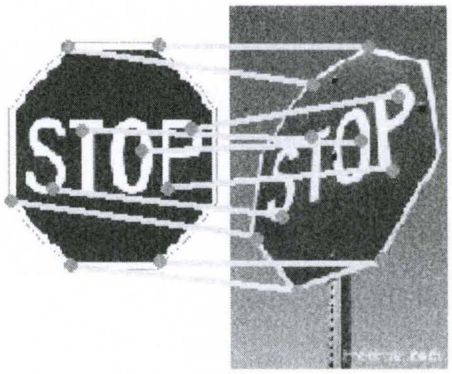


Figure 9 Template matching

3.1.2 Object tracking

Our proposed algorithm effectively integrates mean shift and SIFT feature tracking. Mean shift uses an iterative approach to locate the maxima of a similarity function between the object model and an object candidate by shifting the region of interest in the image toward the mean of its points. We define the similarity measure such that it contains SIFT features in addition to the classical term color.

$$S = \omega_p \times \rho[\vec{\mathrm{p}}(\vec{x}, u), \vec{\mathrm{q}}] \quad , \quad \omega_p = \frac{N(p,q)}{N(q,q)} \qquad \textbf{(Equation 6)}$$

Where $\rho[\vec{\mathrm{p}}(\vec{x}, u), \vec{\mathrm{q}}]$ is the Bhattacharyya coefficient between the candidate model p and the target model q. p and q are defined using the described density kernel function. $\vec{x}, u$ are the spatial and scale parameters we are solving for. S is the new similarity which is defined by multiplying ρ by a weight factor ω_p, which is the proportion of matching SIFT key points of p and q in all the key features of the target model. $N(p,q)$ is the number of matching SIFT features between p and q.

To solve for the parameters of this equation, the EM algorithm is used to estimate the local maxima and also the local scale. EM is an optimization technique that solves for the parameters in two stages of Estimation and Maximization of the unknown parameters.

The EM algorithm is initialized and solved using the method described in (34). From the Jensen's inequality (34) we get:

$$\log(p_x(x,u)) \geq \sum_{i=1}^{N} \log \left(\frac{\omega\left(\left|\frac{x - x_i}{\sigma}\right|^2\right) k\left(\left|\frac{u - u_i}{h}\right|^2\right)}{\mathrm{q}_i} \right)^{\mathrm{q}_i} \qquad \textbf{(Equation 7)}$$

where *qi*-s meet the following requirements:

$$\sum_{i=1}^{N} q_i = 1 \text{ and } q_i \geq 0 \qquad \textbf{(Equation 8)}$$

We can assume that the current estimate values of the parameters are denoted by $\bar{x}^{(k)}$ and $u^{(k)}$.

The E and M steps are as follows:

1)E step: *x* and *u* are kept fixed and *qi*-s are found to maximize the right side of the inequality.

$$\mathrm{q_i} = \frac{\omega\left(\left|\frac{x-x_i}{\sigma}\right|^2\right)k\left(\left|\frac{u-u_i}{h}\right|^2\right)}{\sum_{i=1}^{N}\omega\left(\left|\frac{x-x_i}{\sigma}\right|^2\right)k\left(\left|\frac{u-u_i}{h}\right|^2\right)} \qquad \textbf{(Equation 9)}$$

2)M step: *qi*-s are kept fixed and the left side of the inequality is maximized with respect to *x* and *u*.

By solving the roots of the gradient the update equations will be obtained:

$$\bar{\mathrm{x}}^{(\mathrm{k+1})} = \frac{\sum_{i=1}^{N} y_i\omega\left(\left|\frac{x-x_i}{\sigma}\right|^2\right)k\left(\left|\frac{u-u_i}{h}\right|^2\right)}{\sum_{i=1}^{N}\omega\left(\left|\frac{x-x_i}{\sigma}\right|^2\right)k\left(\left|\frac{u-u_i}{h}\right|^2\right)} \qquad \textbf{(Equation 10)}$$

$$\vec{u}^{(k+1)} = 2\sum_{i=1}^{N} q_i \left(\vec{y}_i - \vec{x}^{(k)}\right)\left(\vec{y}_i - \vec{x}^{(k)}\right)^T \qquad \textbf{(Equation 11)}$$

where y_i-s are the sample points used in defining the distribution p.

3.1.2.1 Stereoscopic tracking

The implemented single tracker is used to track both the left and right streams at the same time. The object of interest in both the left and right images are located at the same horizontal level. When a stop sign is detected in a left frame, it will be tracked in both the current right image and in the next left frame. The stop sign in the right frame is located in the vicinity of the object detected in the left image. The tracker searches for the object in the right image only on the same horizontal level as in the left image.

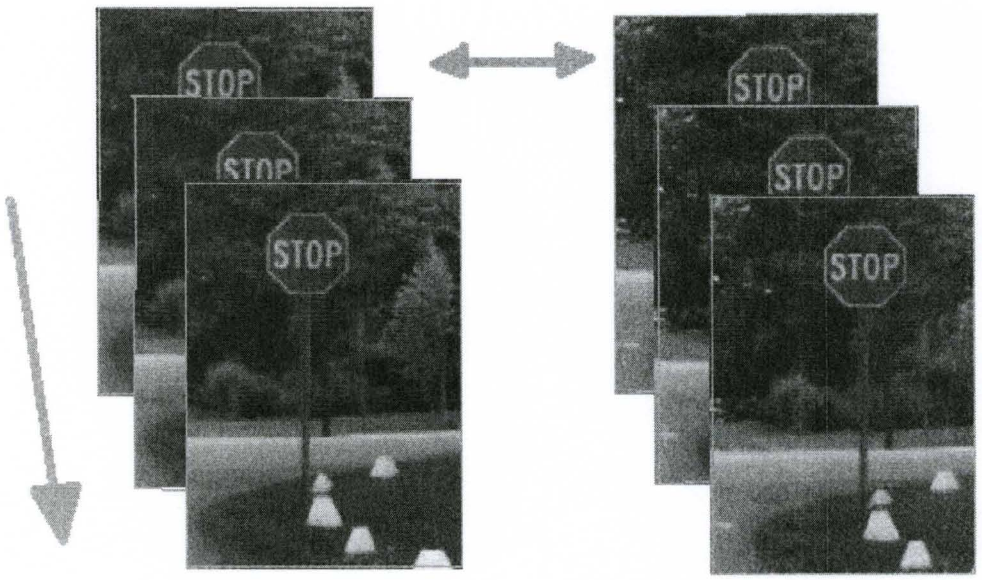


Figure 10 Stereo tracking

3.1.3 Results

The tracking algorithm proposed is scale invariant and can track the object of interest at any shape and scale. The detailed algorithm is as follows:

1) In each frame, using the color measure, the candidates for traffic signs are located. This step is followed by shape analysis to recognize the sign.

2) SIFT features of the detected object is extracted and together with color features are used to initialize the tracking module.

3) The similarity measure is computed based on the proposed formula.

4) The EM algorithm is launched to calculate the new position estimate $x(i+1)$ and the new variance estimate $u(i+1)$. EM algorithm is initialized by the values provided with the detection part. The location and scale of the detection stop sign are the initialization values.

5) The above steps are repeated until no new pixel is included in the object region. For each frame, the above steps are iterated.

The results in the figures below show the comparison between the simple mean shift tracker (Figure 11) and the proposed method (Figure 12).

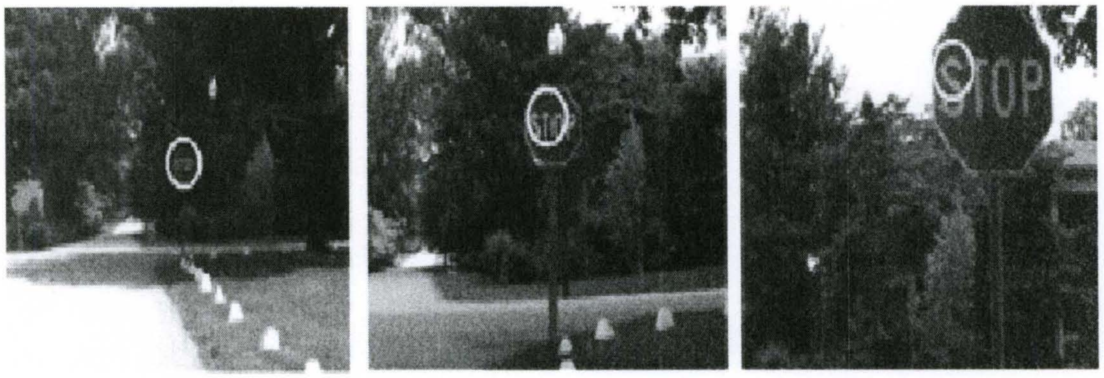


Figure 11 Tracking results using simple mean shift tracker

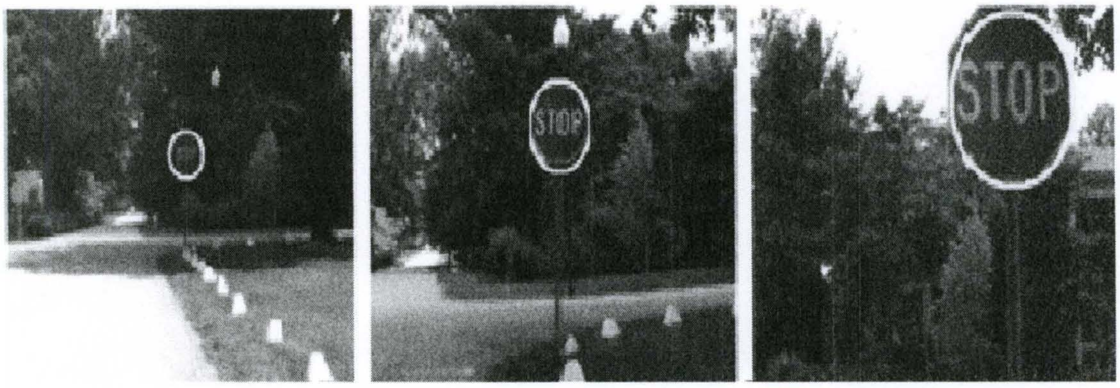


Figure 12 Tracking results using proposed method

For 60 consecutive frames the number of iterations per frame is calculated. The mean number of iterations is 6.32 per frame. Using Matlab, the average time obtained for running each iteration is 131 miliseconds.

3.2 Distance computation and speed estimation

The second task performed in this project involves camera calibration and distance finding. The cameras used in capturing videos need to be calibrated to obtain the parameters such as the focal point. These parameters are later used in stereoscopic matching of the left and right frames, distance finding, and later speed estimation of the vehicle.

3.2.1 Camera calibration

This project consists of some phases that are done independently. In the first phase the camera used for taking videos is calibrated so that parameters can be obtained. The procedure of calibration for a stereoscopic camera is as follows. A checkerboard pattern as in Figure 13 is generated. A number of images (around 20) are taken of this checkerboard pattern and then the left and right images are separated. A similar image taken and decomposed is shown in Figures 14 and 15.

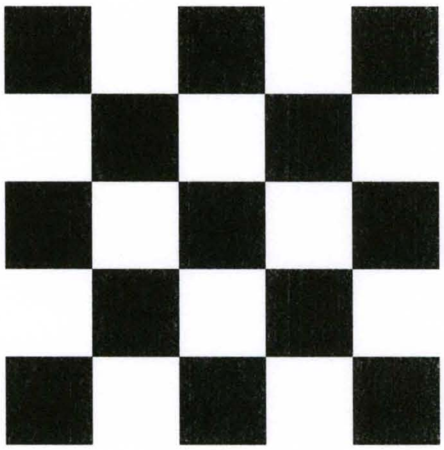


Figure 13 Checkerboard used in camera calibration

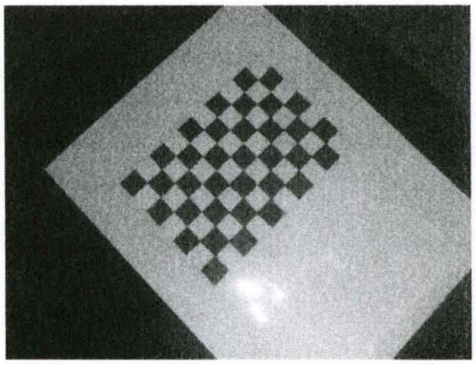


Figure 14 Left frame of checkerboard image

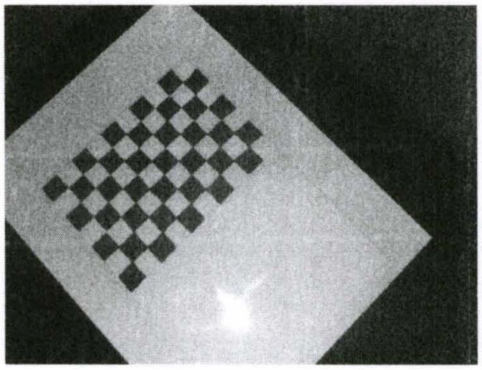


Figure 15 Right frame of checkerboard image

The Matlab toolbox for camera calibration is used to calibrate our stereo camera. Calibration is done by extracting the grid corners from the images. Sample output using Matlab functions is shown in Figure 16.

The average running time for extracting corners from each image is 72.4 milliseconds.

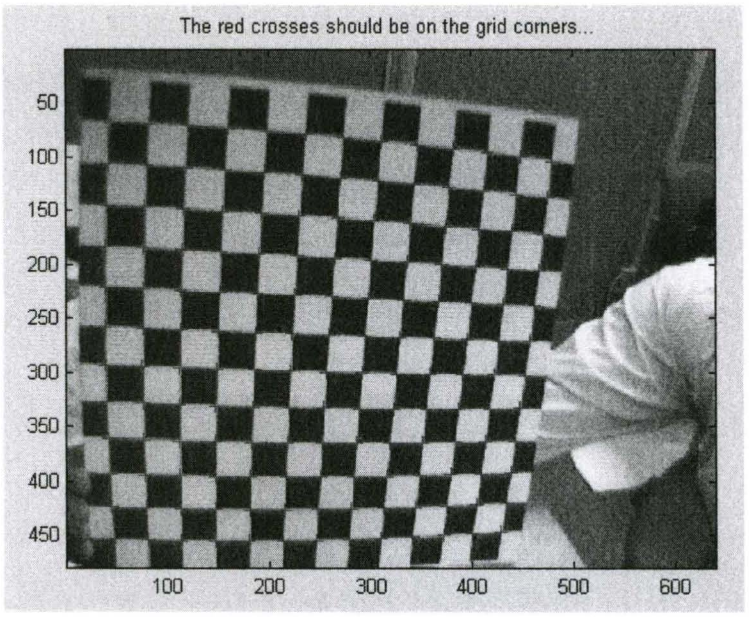


Figure 16 Extracted grid corners (from the Matlab camera calibration toolbox)

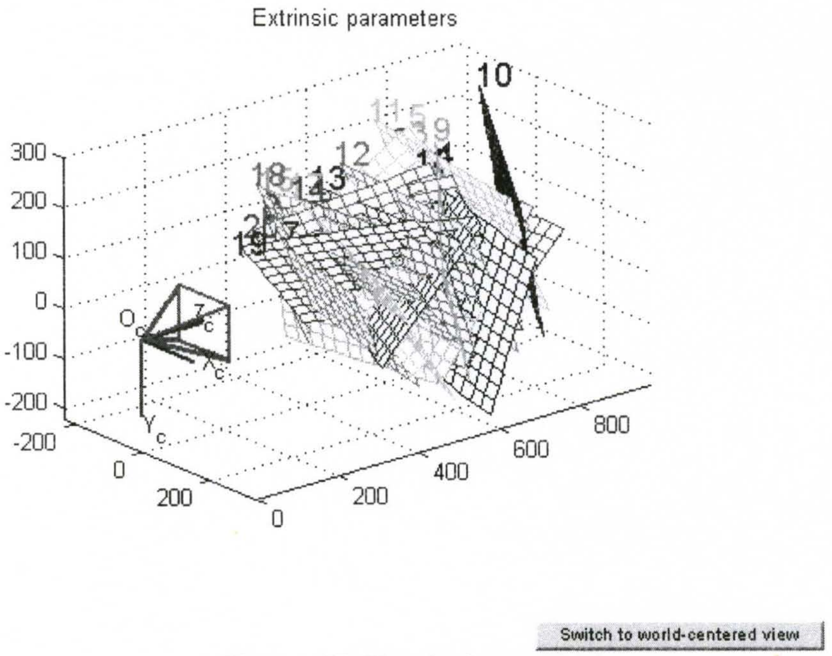
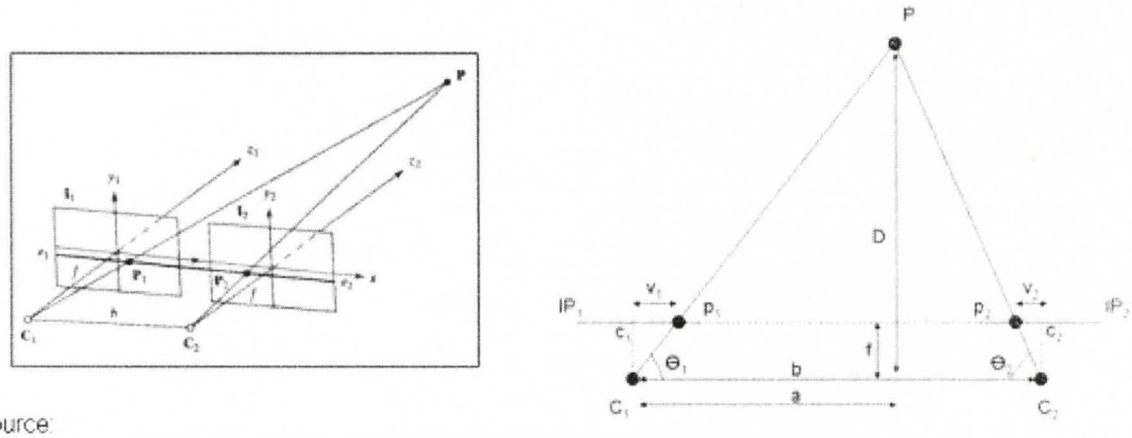


Figure 17 Extrinsic parameters

3.2.2 Stereoscopic matching and distance finding

A common technique for gauging depth information given two offset images is triangulation. The parameters found in the camera calibration phase are used in this phase to estimate the distance of the object from the cameras. The variables used in triangulation are the center points of the cameras (c1, c2), the cameras' focal length (F), the angles (O1, O2), the image planes (IP1, IP2), and the image points (P1, P2).



Source:
http://egweb.mines.edu/faculty/tvincent/Welding/fundamentals_of_stereo_computer.htm

Figure 18 Triangulation method

Figure 18 describes how the triangulation works. The images taken with cameras C1 and C2 are projected in planes IP1 and IP2. So any real world P is represented by P1 and P2 on the planes. F, the focal length of the cameras, is already obtained in camera calibration. The focal length is the distance between the lens and the image plane.

The offset distance between cameras C1 and C2 are shown with B. The displacement of the projected points from the cameras is V1 and V2. The difference between these two values is the horizontal shift of points from one image plane to the other image plane. This is called the disparity of the points P1 and P2. The calculated disparity is used to calculate the actual distance of the points from the cameras. The triangulation formula used is as follows:

$$D = \frac{bf}{d} \qquad \textbf{(Equation 12)}$$

where D is the distance of the point in real world, b is the base offset, f is the focal length of cameras, and d is the disparity.

Employing this algorithm is associated with having the exact projections of P in both image planes. In the tracker, in each step the corresponding object in the right image is obtained from the left image. These two sets of points are the projection of the object of interest in both left and the right planes. From the disparity between these two sets, the actual distance are calculated.

3.2.3 Results

The experiment of distance finding and speed estimation of the car is designed to be performed on two datasets. The first dataset is a video taken with an average speed of 15 mph, and another video is taken with 20 mph. For each video, the algorithm of distance finding is performed on 60 consecutive frames after the stop sign is detected in the first frame. At each frame, the disparity, which is the pixel difference between the left and the right detected stop sign, is calculated. To get more consistent results, for each frame the average disparity in the vicinity of five frames is considered. Based on the triangulation formula, the distance between the car and the stop sign is calculated. The results are shown in the following Figures.

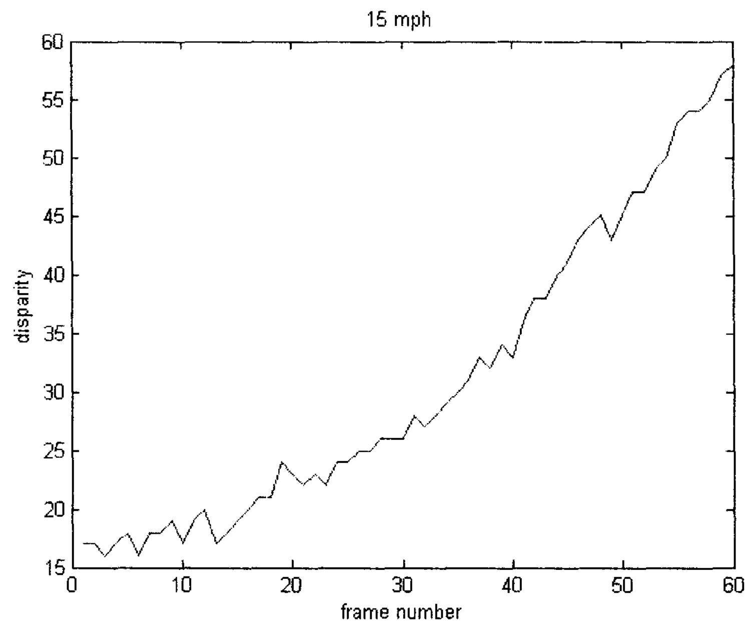


Figure 19 Disparity vs. frames (15 mph)

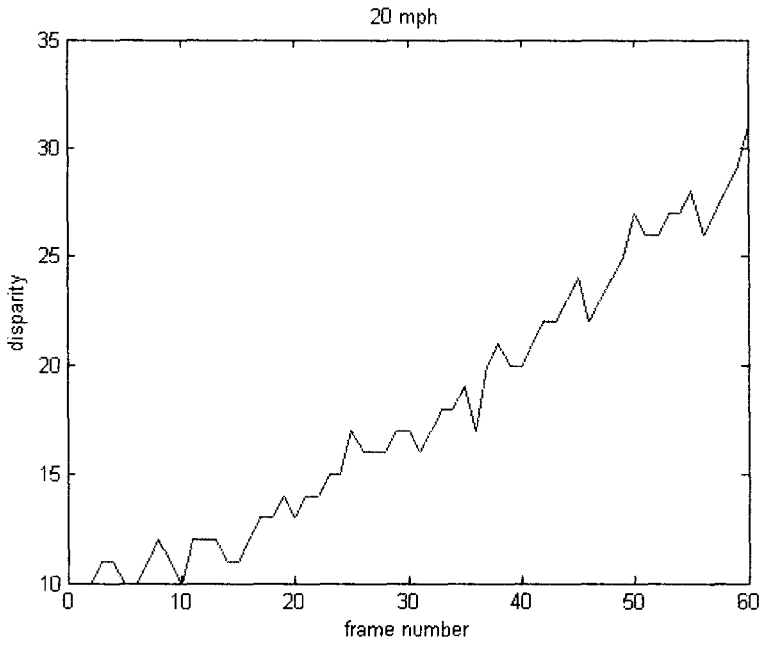


Figure 20 Disparity vs. frames (20 mph)

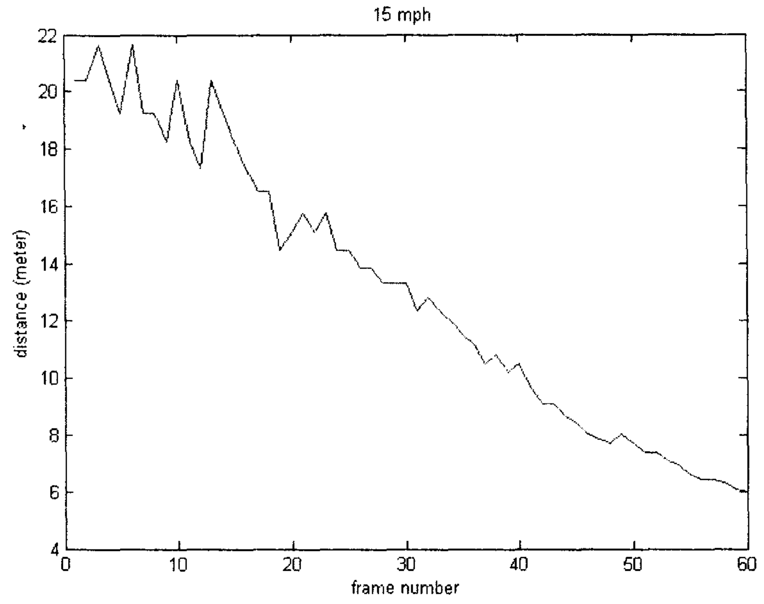


Figure 21 Distance vs. frames (15 mph)

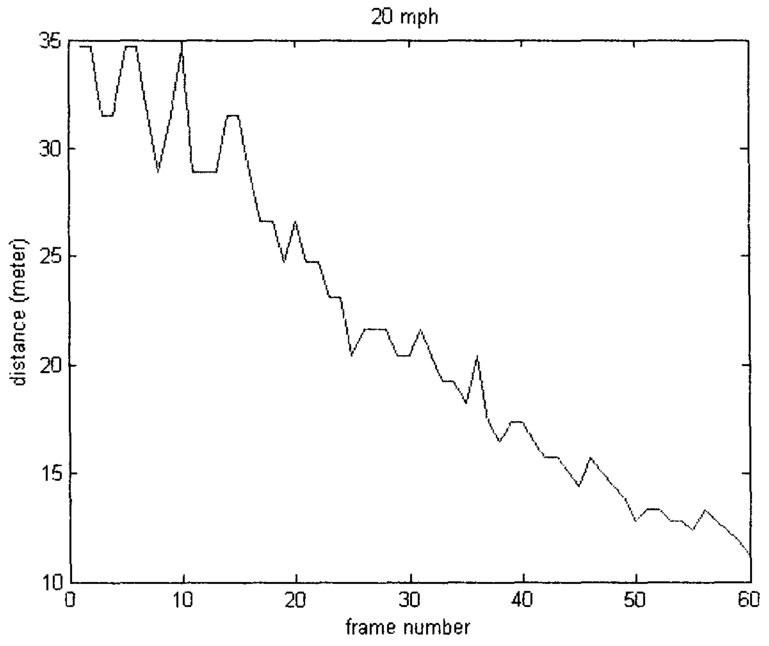


Figure 22 Distance vs. frames (mph)

Figures 21 and 22 shows that the distance calculated decreases more smoothly as the cameras get closer to the stop sign and the distance becomes less. This is due to disparity computing. For farther distances the disparity and also the object region are small and disparity is more susceptible to noises when calculated. As plots shown in Figures 21 and 22, the speed of the car can be calculated and compared to the actual speed of the car for both videos. The camera takes 30 frame per second so each frame is 1/30th of a second.

Table 1 Speed estimation

videos	average speed estimated
15 mph	6.2 meter per second = 13.84 mph
20 mph	10.1 meter per second = 22.59 mph

4 CONCLUSION AND FUTURE WORK

Autonomous traffic sign detection and tracking is interesting and one of the main areas of research in computer vision. Traffic signs provide drivers with valuable information about the road, in order to make driving safer and easier. They also play the same role for autonomous vehicles. For example, Google Driverless Car is a recent project by Google that involves developing an autonomous vehicle. With increasing interests in 3D visual related technology, building a system that works with 3D inputs and gives the results back to the users in the form of 3D is very challenging and interesting.

In the present work, a system is developed that takes stereoscopic input videos of roads and highways to detect and track stop signs. Furthermore, the distance of the stop sign to the vehicle is computed, and the speed of the vehicle is estimated as feedback to the driver.

Detecting and tracking traffic signs are among the challenging tasks in computer vision. Videos of outdoor environments are usually susceptible to illumination changes and pixel densities of the object of interest vary a lot throughout the videos. Due to the motion of the camera the traffic signs may be deformed; for example, stop signs do not always appear as the symmetric, octagonal shapes in the video with uniform shades of red color. In addition, the camera is located on the moving vehicle and the method used should be able to detect and track the traffic sign with any size and scale at any distance from the vehicle.

In the developed system, the changes in the color and shape of the stop sign are considered in both detection and tracking parts. A novel idea is proposed in the tracking method to track the stop sign at any scale and in real time. It is important to highlight that the algorithm tracks the changes in shape and scale of stop signs from frame to frame in a video and estimates the local scale of the tracked traffic sign in the image in addition to its new location.

Although there is a long way in building a complete driverless car, building a driver assistance system helps a lot in providing feedback to the driver and increasing driving safety.

Future work will be improving the detection part of the system to include other kinds of traffic signs in addition to the stop sign.

This system can be used in augmented reality environments like games when the user wears 3D glasses and the view of the road he is driving on is shown to him in the form of 3D with highlighted traffic signs.

REFERENCES

1. *Object recognition from local scale-invariant features.* **Lowe, R.**,Proc. Internation Conference on Computer Vision ICCV, 1999, pp. 1150-1157.
2. *Road sign detection and tracking.* **Chiung-Ya Fang, Sei-Wang Chen, Chiou-Shann Fuh.** 2003, IEEE Transactions on vehicle technology, pp. 1329-1341.
3. *A system for traffic sign detection, tracking, and recognition using color, shape, and motion information.* **Bahlmann, Y. Zhu, Y. Ramesh, M. Pellkofer, T. Koehler.** IEEE Intelligent Vehicles Symposium, 2005. pp. 255-260.
4. *color-based road sign detection and tracking.* **Luis David lopez, Olac Fuentes.** montreal, CA : international conference on image analysis and recognition (ICIAR), 2007.
5. *A study on traffic sign recognition in scene image using genetic algorithms and neural networks.* **Y. Aoyahi, T. Asakura.** 22nd International Conference on Industrial Electronics, Control, and Instrumentation, IEE, 1996.
6. *Automated traffic sign detection for modern driver assistance systems.* **Alexander Reitere, Taher Hassan, Naser El-Sheimy.** Sydney, Australia : FIG Congress, 2010.
7. *Robust method for road sign detection and recognition.* **G. Piccioli, E. de Micheli, P. Parodia, M. Campani.** Image and Vision Computing 14 (3), 1996, pp. 209-223.
8. *Fast greyscale road sign model matching and recognition.* **Sergio Escalera, Petia Radeva.** Recent Advances in Artificial Intelligence Reseach and Development, 2004, pp. 69-76.
9. *Fast shape-based road sign detection for a driver assistance system.* **G.Loy, N. Barnes.** Proceedings of International Conference on Intelligent Robots and Systems (IROS), 2004.
10. *Fast traffic sign detection on greyscale images.* **Xavier Baró, Jordi Vitrià.** Congrés Català d'Intelligència Artificial (CCIA), 2004.
11. *Real-time object detection for smart vehicles.* **D.M. Gavrila, V. Philomin.** International Conference on Computer Vision, IEEE, 1999.
12. *Analysis of traffic scenes using the hierarchical structure code.* **H. Austerirmeier, U. Buker, B. Merstching, S. Zimmermann.** International Workshop on Structural and Syntactic Pattern Recognition, 1992.
13. *Shape classification for traffic sign recognition.* **B. Besserer, S. Estable, B. Ulmer, D. Reichqardt.** First International Workshop on Intelligent Autonomous Vehicles (IFAC), 1993.
14. *Parallel evaluation of hierarchical image databases.* **U. Buker, B. Mertsching.** Journal of Parallel and Distributed Computing 31 (2), 1995, pp. 141-152.

15. *Unsupervised statistical detection of changing objects in camera-in-motion videos.* **Rozenn Dahyot, Pierre Charbonnier, Fabrice Heitz.** International Conference on Image Processing (ICIP), 2001.
16. *A system for traffic sign detection, tracking, and recognition using color, shape, and motion information.* **Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, Martin Pellkofer, Thorsten Koehler.** Las Vegas, NV : IEEE Intelligent Vehicles Symposium (IV), 2005.
17. *Road traffic sign detection and classification.* **A. de la Escalera, L. Moreno, M.A. Salichs, J.Ma. Armingol.** IEEE Transactions on Industrial Electronics 44 (6), 1997, pp. 848-859.
18. *Robust method for road sign detection and recognition.* **G. Piccioli, E. de Micheli, P. Parodia, M. Campani.** Image and Vision Computing 14 (3), 1996, pp. 209-223.
19. *An automatic road sign recognition system based on a computational model of human recognition processing.* **C.Y. Fang, C.S. Fuh, P.S. Yen, S. Cherng, S.W. Chen.** Elsevier Computer Vision and Image Understanding 96, 2004, pp. 237-268.
20. *Road sign detection and recognition using matching pursuit method.* **S.H. Hsu, C.L. Huang.** Elsevier Image and Vision Computing 19, 2001, pp. 119-129.
21. *Traffic sign detection for driver support systems.* **A. de la Escalera, J. M. Armingol, M.A. Salichs**. 3rd International Conference on Field and Service Robotics , 2001.
22. *Traffic sign recognition and analysis for intelligent vehicles.* **A. de la Escalera, J.M. Armingol, M. Mata**. Image and Vision Computing, 2003, Vol. 11, pp. 247-258.
23. *Visual sign information extraction and identification by deformable models for intelligent vehicles.* **A. de la Escalera, J.M. Armingol, F.Rodriguez, J.M. Pastor**. IEEE Transactions on Intelligent Transportation Systems, 2004, Vol. 5, pp. 57-68.
24. *Road-sign detection and tracking.* **Chiung-Yao Fang, Sei-Wang Chen, Chiou-Shann Fuh.** IEEE Transactions on Vehicular Technology, 2003, Vol. 52, pp. 1329-1341.
25. *An invariant traffic sign recognition system based on sequential color processing and geometrical transformation.* **D.S. Kang, N.C. Griswold, N. Kehtarnavaz.** Dallas, TX : Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretaion, 1994, pp. 88-93.
26. *Efficient and effective querying by image content.* **C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz** . Journal of Intelligent Systems, pp. 231-262.
27. *Photobook: Tools for content-based manipulation of image databases.* **A. Pentland, R. Picard, and S. Sclaroff.** San Jose, Calif. : SPIE, 1994, pp. 34-47.
28. *3D model based vision for innercity driving scenes.* **K. Fleischer, H. Nagel, T.M Rath.** Versailles, France : IEEE Intelligent Vehicles Symposium, 2002, pp. 477-482.
29. *Region tracking via level set pdes without motion computation.* **Mansouri, A.** , PAMI, 2002, pp. 947-961.
30. *Statistical shape knowledge in variational motion segmentation.* **D. Cremers, C. Schnorr.** Image and Vision Computing, 2003, pp. 77-86.
31. *Contour based object tracking with occlusion handling in video acquired using mobile cameras.* **A. Yilmaz, X. Li, M. Shah**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, pp. 1531-1536.
32. *Kernel-based object tracking.* **D. Comaniciu, V. Ramesh, P. Meer.**Transaction on Pattern Analysis and Machine Intelligence, 2003, Vol. 25.

33. *Object tracking: a survey.* **A. Yilmaz, O Javed, M Shah.** , Acm Computing Surveys (CSUR), 2006.
34. *An EM-like algorithm for color-histogram-based object tracking.* **Z. Zivkovic, B. Krose.** Computer Vision and Pattern Recognition, 2004, pp. 798-803.
35. *Mean shift blob tracking through scale space.* **Collins, R.** Computer Vision and Pattern Recognition, 2003.
36. *Stereo tracking and three-point/one-point algorithms - a robust approach in visual odometry.* **K. Ni, F. Dellaert.** Atlanta, GA : 2006 IEEE International Conference on Image Processing, 2006, pp. 2777 - 2780.
37. *Survey over image thresholding techniques and quantitative performance evaluation.* **Mehmet Sezgin and Bulent Sankur.** Journal of Electronic Imaging 13(1), 146–165, 2004.
38. *A general framework for object detection.* **Papageorgiou, C., Oren, M., Poggio, T.** 1998. In IEEE International Conference on Computer Vision (ICCV), pp. 555–562.

CURRICULUM VITAE

Behnoush Abdollahi

Personal Information

Date of Birth: May 10, 1986
Citizenship: Iran
E-mail: b0abdo03@louisville.edu
Phone Number: 502-599-4725

ACADEMIC TRAINING

Jan 2010- Present	M.S. Computer Science and Engineering University of Louisville, KY, U.S.A
Sep. 2004- Feb 2009	B.S, Computer Engineering, University of Tehran, Tehran, Iran.

TEACHING EXPERIENCE

CECS Department, University of Louisville
- Introduction to Java, Summer 2011.
- Introduction to Compilers, Spring 2010.

ECE Department, University of Tehran
- Operating Systems, Fall 2007.

PROFESSIONAL SOCIETY MEMBERSHIP

ACM student member
IEEE student member
Kentucky Academy of Science

PROFESSIONAL EXPERIENCE

Jan 2010- Aug. 2011	Graduate Teaching Assistant, Department of Computer Engineering and Computer Science, University of Louisville
July 2007- Jan 2008	Developer, Router Lab, ECE Department, University of Tehran

HONORS AND AWARDS

Provost **Fellowship Recipient** from University of Louisville (The most prestigious university-wide scholarship at UofL - two years), Fall 2011.

Anita Borg **Scholarship Recipient** to attend Grad Cohort, 2011.

Graduated as **Exceptional Talent** from University of Tehran, Iran.

Ranked **Top 0.1 percent** among over 1,000,000 participants in National University Entrance Exam (Konkoor-e-Sarasari), June 2004.

Ranked **Top 0.01 percent** among all participants for Electrical Engineering in National Azad University Entrance Exam , June 2004.

Qualified for **Semifinal Round** of national Mathematics and Physics Olympiad, 2003.

PUBLICATIONS

B. Abdollahi, *"A Fast Scale Invariant Method for Traffic Sign Detection and Tracking"*, Grace Hopper Conference (GHC), Portland, OR, 2011.

B.Abdollahi, *"Road sign tracking in vehicle driving"*, Poster Presentation, CRA-W Grad Cohort for Women Program, Boston, April 2011.

B.Abdolahi, *"Stereoscopic vision in augmented reality and vehicle driving"*, Poster Presentation, E-Expo, Louisville, Spring 2011.

B.Abdollahi, *"Stereoscopic vision in augmented reality and vehicle driving"*, Oral Presentation, Kentucky Academy of Science,November 2011.

Participated in the IPM-UNU Winter School on Foundations and Trends in Computer Science, (IPM/UNU - FTCS), Jan 31 - Feb 10 of 2008, School of Computer Science, IPM, Tehran, Iran.