Proceedings of the Twelfth IASTED International Conference
Intelligent Systems and Control (ISC 2009)
November 2 - 4, 2009 Cambridge, MA, USA

LANDMARK EXTRACTION USING CORNER DETECTION AND K-MEANS CLUSTERING FOR AUTONOMOUS LEADER-FOLLOWER CARAVAN

Andrew B. Nevin
Department of Electrical and Computer Engineering
Auburn University
Auburn, Al 36849, United States
email: nevinab@auburn.edu

David M. Bevly
Department of Mechanical Engineering
Auburn University
Auburn, Al 36849, United States
email: bevlydm@auburn.edu

ABSTRACT
Extracting a landmark in a digital video sequence is vital for visual navigation. For the application of a leader-follower caravan, a database of man-made or natural landmarks is created. This research focuses on the challenge of the follower-vehicle losing line-of-sight with the leader vehicle. An experiment is conducted using data in an outdoor environment to test an algorithm based on corner detection and classifying them into subsets using k-means clustering techniques. Assuming all corners from a landmark will be grouped into the same cluster, the centroid of each cluster is established as a region of interest and is stored in an output file. From the results, it is concluded that the optimal parameters for the algorithm are to extract the ten strongest features and cluster them into three subsets. These parameters yield a success rate of 98%. All regions of interest are used to build a database of visual landmarks or objects for the follower vehicle to use for visual navigation.

KEY WORDS
computer vision, landmark extraction, k-means clustering, corner detection

1 Introduction

Landmark extraction is an important research topic for using computer vision in an autonomous leader-follower application. Leader-follower caravans are currently being implemented to replace humans on jobs that are dangerous, time-consuming, or difficult. A key step in creating an autonomous leader-follower caravan is self-localization. Typically, GPS is used to estimate a vehicle's current position. However, due to indoor environments, dense trees, or urban canyons, the line of sight between the satellites and vehicle can be blocked. Therefore it is necessary to continue research in visual navigation.

1.1 Previous Work

Kannan et al [1] designed an algorithm using visual sensors for a leader-follower caravan. The follower vehicle is equipped with a camera. While the follower vehicle is at the desired pose relative to the leader, a snapshot is taken of the leader vehicle. The follower vehicle uses this image as a reference image. The control algorithm attempts to match the follower's pose with that of the reference image at all times.
However, this research is focused on investigating the problem of losing line-of-sight with the leader vehicle. The goal is to have the leader vehicle create a database of visual signatures composing of natural landmarks or objects. The follower vehicle will then match these landmarks and calculate it's pose. There is a major challenge with this scenario: there may be hundreds or even thousands of features detected. Extracting landmarks may yield a number of distinguishable features that must be indexed in order to determine which landmark the robot may be viewing [2].

1.2 Contributions

In an autonomous leader-follower caravan, extracting landmarks or other visual signatures is helpful for maximizing performance when GPS signals are blocked. The objective of this research is to investigate the challenges of landmark extraction by the leader vehicle by the process of identifying features and using an unsupervised machine learning algorithm to group them into subsets. Key features are extracted using the Shi and Tomasi algorithm in [3]. K-means clustering is implemented to create subsets of data. The centroid of each cluster is established as a region of interest, where there is a high probability that there is a landmark or some other visual signature present. Key parameters in this algorithm are the number of features extracted for each frame in the video and the number of clusters formed. A study is performed to discover the optimal parameters in this algorithm. Testing is performed on data acquired from a vehicle driving down a road.

This paper is organized as follows. In Section 2, the background on feature extraction is discussed. Section 3 describes the machine learning algorithm used for creating subsets of data used for extracting landmarks. The regions of interest extraction process is in Section 4. A rough software outline and description is provided in Section 5. Experiments and results are in Section 6. Conclusions and future work are discussed in Section 7.

2 Feature Extraction Background

Common features tracked over many video frames are corners. Corners are typically used for tracking, because they have strong second-order derivatives, whereas individual pixels on smooth surfaces are difficult to extract from frame to frame.

Consider two grayscale images *I* and *J*, where they are captured sequentially in time. Tomasi and Kanade [4] describe this model as

$$J(x) = I(x - d) + n(x) \tag{1}$$

where *d* is the displacement vector between the two images, and *n(x)* is noise. The vector *d* is selected to minimize the error below:

$$\varepsilon = \int_W [I(x - d) - J(x)]^2 dx \tag{2}$$

, where *W* is a given window in the images. Assuming that the displacement vector *d* is small, *I(x-d)* can be approximated by the linear term from the Taylor series expansion.

$$I(x - d) = I(x) - g \cdot d \tag{3}$$

Rewriting equation (2) becomes

$$\varepsilon = \int_W [I(x) - g \cdot d - J(x)]^2 dx = \int_W (h - g \cdot d)^2 dx \tag{4}$$

where *h = I(x)- J(x)*. Least squares minimization is performed on the latter expression from above, and yields

$$\int_W (h - g \cdot d) g \ \ dA = 0 \tag{5}$$

Simplifying leads to

$$\int_W = (gg^T \ \ dA) \cdot d = \int_W h \cdot g \ \ dA \tag{6}$$

Now there are two scalar equations with two unknowns.

$$Gd = e \tag{7}$$

This equation is solved to find the displacement vector *d*, also known as optical flow. However for this research, optical flow is not computed. Instead, the gradient matrix *G* is desired to be computed.

$$G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{8}$$

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}^T \tag{9}$$

To extract strong corners, one must compute the gradient matrix *G* in (8) over each pixel $p(x, y)$ using an integration window size of w_x by w_y on the image. I_x and I_y are given by (9). This gradient matrix is derived in [3].

Strong features will have a gradient matrix *G* that is above the image noise level and well-conditioned[4]. This characteristic is determined by examining the eigenvalues of *G* for every integration window for each pixel. The following conclusions can be made about the texture of the image in the integration window:

- If both eigenvalues λ_1 and λ_2 are above a set threshold λ, then the window contains a corner.
- If one eigenvalue is large and the other is small compared to the threshold λ, then the window contains an edge.
- If both eigenvalues are under the threshold λ, then the window contains a smooth surface.

The method of computing the threshold λ is the same as in [5]. The steps are:

1. Compute the G matrix and its minimum eigenvalue λ_m at every pixel in the image.
2. Refer to λ_{max} as the maximum value of λ_m over the entire image.
3. Retain the image pixels that have a λ_m value larger than a percentage of λ_{max}.
4. From those pixels, retain the local max. pixels in a 3 x 3 neighborhood.
5. Keep the subset of pixels so that the minimum distance between any pair of pixels is larger than a given threshold distance.

The resulting data is a binary image containing the extracted features. A higher level of programming is needed to interpret the results.

3 Machine Learning

The purpose of using machine learning algorithms is to interpret raw data for implementing a difficult task such as pattern recognition, object tracking, face detection, etc. These learning algorithms analyze features, adjust weights, thresholds, or other parameters to maximize performance [6]. There are two different types of machine learning algorithms: supervised and unsupervised. Supervised learning classifies data into labels, such as a sign, face, tree, etc. Unsupervised learning partitions data into clusters, so that the data in each cluster share a common trait [7].

For the application of landmark extraction, an assumption is made that all corners grouped closely are apart of the same object. K-means clustering, an unsupervised learning algorithm is used to classify these features, and is implemented by the steps below:

1. Randomly place k centroids in the vector space.

2. Calculate the Euclidean distance or the root of square differences between each vector to the centroids given by the equation below:

$$D = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \qquad (10)$$

3. Assign vectors to the centroid with the minimum distance.

4. Calculate the new centroids for each cluster.

5. Repeat steps 2-4 until convergence.

The centroid of each cluster is established as a region of interest or ROI. These ROIs can be extracted so that a database of landmarks can be created. For this experiment the coordinates of each feature detected by using the equation in (8) are the data points partitioned by the k-means learning algorithm. An example of k-means clustering is in Fig 1. There are sets of data spread throughout the cartesian coordinate system. The algorithm groups the points into subsets. A circle with the origin at the centroid for each subset is drawn around the group.

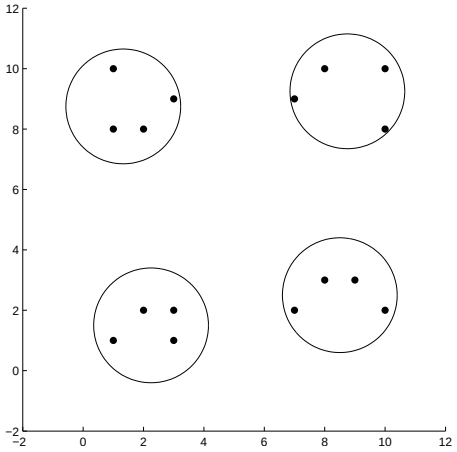

Figure 1. An example of k-means clustering results. All points belong to a subset.

After convergence is reached, an area around the ROI is extracted and stored in memory, where more digital image processing techniques can be used to characterize the area. This information will be used to create a database of features for the follower vehicle to navigate with.

4 Region of Interest

For each centroid, a 50 x 50 square pixel area is established as the ROI. Each centroid from the clusters will yield its own ROI. Note that the total resolution of the experimental data is 640 x 480.

5 Software Outline

A rough outline of the software implementation is presented below. The inputs are an .avi video, the number of desired clusters, and the maximum features extracted. To increase processing speed, memory is allocated for all variables and image results. Each frame is converted from RGB to grayscale by calculating the square root of the sum of squares for each channel. Then the Shi and Tomasi feature detection [3] is implemented. The coordinates of the detected features are partitioned by the k-means clustering algorithm. The k-means clustering algorithm is run iteratively until convergence or until it reaches the maximum number of ten iterations. The centroids are extracted and becomes the center of the ROI. The ROI is cropped and printed to a separate image folder. This process is repeated for every frame of the video data.

Inputs- .avi file, ClusterCount, MaxFeatures
Output- image files for each ROI

Allocate memory for variables and image results

for(;;)
    convertToGrayScale(frame)
    featureExtraction
    featurePoints → kMeansClustering
    extractCentroids
    cropROI
    printImageToFile
end

6 Experiment and Results

A video sequence from a car is taken for data. The camera is mounted on the inside of the vehicle on the upper windshield. The surrounding environment includes green signs posted every so often, as well as a wall of trees on both sides of the road. A sample frame of the video data is displayed in Fig 2.

Figure 2. A sample frame from the video sequence.

Figure 3. An example binary frame from the feature extraction method. Any window that has two large eigenvalues in the gradient matrix G will be white in the image. Notice there are two distinct areas that correspond to signs in the actual video frame.

Table 1. Coordinates of ten features extracted. This data is grouped by the k-means clustering algorithm.

x pixel	y pixel
353	105
353	100
148	120
148	120
631	69
617	33
621	28
383	114
393	114
613	49

Experiments were performed with varying parameters: cluster count and the number of features extracted. A notion of success of the algorithm is designed. A successful frame has at least one object or natural landmark selected within the region of interest. A resulting image for the features extracted is in Fig 3. The corners of the signs are clearly visible. The coordinates of these features are recorded in Table 1. This data is an example of what is used for the k-means clustering algorithm. The results for the ten strongest features is in Table 2. There is a 42 % rate of success for using one cluster for classifying the features extracted. The features are spread out through the frame, causing the centroid to be offset on a relatively smooth surface, which is difficult to extract any valid information for the follower vehicle. Using two clusters will yield two regions of interest. The results from the same video data are 86%, where 53% of that are two different objects or landmarks detected. Using three clusters yields a 98% success rate, in which 47% are three different objects or landmarks detected. The results for the twenty strongest features is in Table 3. Using one, two and three cluster groups yields results of 33%, 68%, and 90% respectively. It is apparent from these results that extracting the ten strongest features with three cluster groups are the optimal parameters.

Table 2. Results for ten features extracted per frame.

	1 Cluster	2 Clusters	3 Clusters
Success Rate	42 %	86 %	98 %

Table 3. Results for twenty features extracted per frame.

	1 Cluster	2 Clusters	3 Clusters
Success Rate	33 %	68 %	90 %

Figure 4. A frame from the video sequence with two ROIs encircled. Both ROIs contain a road sign.

There is a simple user interface for observing the results without having to search through the output files. The ROIs are encircled on the video sequence as seen in Fig 4. Two of the output images are in Fig 5. A road sign and tree are extracted. Both ROIs have many contours and intensity distributions to use as a visual signature for the follower vehicle to detect.

7 Conclusion and Future Work

7.1 Conclusion

A common problem in using k-means clustering is the presence of outliers. All data points in the space are placed into a subset. When a stray point is placed in a subset, the centroid can dramatically change. This change can lead the

a) Street sign b) Tree from the horizon

Figure 5. ROI outputs

ROI to be over a smooth surface, which will not contain sufficient data such as contours, Hu moments, or other intensity distribution patterns.

However, a strong advantage for using this landmark extraction algorithm is that there are many digital image processing methods available for the follower vehicle to locate objects or natural landmarks. A common method used is template matching. Template matching uses a small image known as the template $t(x, y)$, and is shifted over an entire input image $f(x, y)$ until there is a match. Research to improve the speed and robustness of template matching is in [8] and [9]. Histogram matching is another method used for locating known objects in an image. Schiele and Crowley use histograms for object recognition in [10]. Hu moments are another popular method. Comparing Hu moments is a simple way for comparing two objects and their contours [6]. Potocnik performed research on using region-based moments for object recognition [11].

Another advantage of this algorithm is that it is designed for the situation when the line-of-sight is broken between the two vehicles. The database of features will be considerably smaller and less complex than storing hundreds or thousands of features, which is a common challenge in landmark navigation. By building a database of visual signatures, the follower vehicle will be able to traverse the same path as the leader.

7.2 Future Work

K-means clustering is one method of unsupervised machine learning. Alternative supervised and unsupervised methods will be researched as well as to make the algorithm more robust. After an ROI is extracted, there needs to be a method for determining if there is indeed a landmark in the image. Presently, all ROIs are printed to the output file. If there can be a method to reduce the number of outputs that are not strong for tracking, it would reduce the complexity of the data mining for the follower-vehicle.

Now that some landmarks and other visual signatures have been extracted by the leader vehicle, they need to be classified. Some information will include contours, color information, and Hu moments. These methods will be researched as well.

References

[1] Hariprasad Kannan, Vilas K. Chitrakaran, Darren M. Dawson, and Timothy Burg, Vision-Based Leader/Follower Tracking for Nonholonomic Mobile Robots. *In Proc. American Control Conference*, 2007, 2159-2164.

[2] Sala, P. Sim, R. Shokoufandeh, A. & Dickinson, S, Landmark Selection for Vision-Based Navigation Robotics, *IEEE Transactions on 22*(2), 2006, 334-349.

[3] Shi, J. & Tomasi, C, Good Features to Track, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, 593-600.

[4] Tomasi, C. & Kanade, T, Detection and Tracking of Point Features, *International Journal of Computer Vision*, 1991.

[5] Bouguet, J, Pyramidal Implementation of the Lucas Kanade Feature Tracker: description of the algorithm. Technical report, OpenCV Document, Intel Corporation Microprocessor Research Labs, 2003.

[6] Bradski, G. & Kaehler, A, *Learning OpenCV* (Sebastopol, CA: O'Reilly, 2008).

[7] Fernando, W.; Udawatta, L. & Pathirana, P, Identification of moving obstacles with Pyramidal Lucas Kanade optical flow and k means clustering, *Information and Automation for Sustainability, 2007. ICIAFS 2007. Third International Conference on*, 2007, 111-117.

[8] Omachi, S. & Omachi, M, Fast Template Matching With Polynomials Image Processing, *IEEE Transactions on 16*(8), 2007, 2139-2149.

[9] Shin, B. G.; Park, S. & Lee, J. J, Fast and robust template matching algorithm in noisy image Control, *Automation and Systems, 2007. ICCAS '07. International Conference on*, 2007, 6-9

[10] Schiele, B. & Crowley, J. L, Object Recognition Using Multidimensional Receptive Field Histograms, *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I* 1996, 610-619.

[11] Potocnik, B, Assessment of Region-Based Moment Invariants for Object Recognition, *Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006*, 2006, 27-32.