A Feature Clustering and Multi-Object Alignment Framework for Detailed Vehicle Recognition

Vladimir Y. Mariano
Institute of Computer Science
University of the Philippines Los Banos
College, Laguna, Philippines 4031
(63-49) 536-2313/2302 vymariano@uplb.edu.ph

Rangachar Kasturi
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802, USA
kasturi@cse.psu.edu

ABSTRACT

We present a novel method for detailed recognition of vehicle images. Given the output of a general image retrieval system, an iterative scheme uses multi-object feature clustering and image alignment to reduce the number of images in the search space. In each iteration, the reduced search space becomes more homogeneous and better aligned. The reduction scheme uses the cluster membership of object features to decide which objects will be pruned from the search space. The method is applied to recognition of recurring vehicles in a street scene.

Keywords

Object recognition, image registration, feature clustering.

1. INTRODUCTION

Vehicles are one of the most common objects found outdoors. They are used to go from place to place and transport all kinds of cargo. Human activity directly affects their occurrence and behavior. Part of studying and monitoring human activity is recognizing the objects that we commonly use and interact with.

Each vehicle is unique. In order to distinguish each one, a license plate is attached to the front or the rear. Like a person's nametag, a license plate gives a vehicle its identity. Visual features can also give a vehicle its unique identity. A good example would be finding our own car in a parking lot when we forgot where he parked it. Our first visual cues are typically shape and color, then if a match is found we look for more smaller features like stickers and dents. And finally to make sure, we look at the license plate to verify that it is my car.

Throughout the life of a vehicle, its visual identity gets established. The owner can attach new wheels, put membership stickers like AAA, repaint the body, and the car may develop rust in lower areas. The acquired features enable us to distinguish vehicles even of the same make, model and color. As a human can distinguish vehicles by visual features, we seek to understand how we can make a computer do it using computer vision.

The application of interest in this paper is a system for monitoring a limited view of a street scene. The goal is to observe each passing vehicle and determine whether the currently observed vehicle has appeared before in the same scene. In the light of recent heightened security in major buildings like airports, embassies and other federal buildings, this application can be used to observe vehicles and detect any suspicious activity. We explore a computer vision strategy towards realizing this application.

1.1 Previous Work on Vehicle Features and Recognition

In the literature, there has been significant work on extracting vehicle features. Work has been focused on extracting the general shape and classification into the major vehicle types (sedan, truck, wagon, etc.). Color features are also extracted for vehicle matching. Identification of vehicles using body features were limited to recognition of license plates [1][2].

Collins, et al [3] developed a system for detecting, tracking and classifying moving blobs. A neural network is trained to classify the moving blobs into single human, human group, vehicle, and clutter. The input features to the network are blob dispersedness, blob area, blob aspect ratio and camera zoom. Vehicles are further classified into different types (van, truck, or sedan) and color using linear discriminant analysis. For vehicle type, the feature vector consists of blob features: area, center of gravity, width, height, and the first three moments taken along the row and column pixel axes. The color feature vector has three dimensions (I1,I2,I3). They claim that the method was also trained to recognize specific vehicles like a UPS truck and the campus police car.

Tan and Baker [4] presented algorithms for localizing and classifying vehicles (high-roof van, minibus, saloon/sedan) based on image gradients in a small window. The vehicle pose is computed using a ground-plane constraint and the fact that the appearance of most vehicles is dominated by two sets of parallel lines.

Fung, et al [5] proposed a method for approximating vehicle shape using motion observed using a high video camera. The height of feature points (corners) are estimated to create a height profile. The basic idea is that feature points that are higher (thus closer to the camera) move faster than the lower feature points. The height profile can then be used later for vehicle classification.

Jolly, et. al. [6] used shape, color and edge features to match vehicles between two sites for measuring time of travel. Deformable templates were used to classify vehicles based on shape information. Side-view, 2-D deformable templates of five vehicle types (sedan, pickup, hatchback, station wagon and van) were fitted on the front vehicle's edge image. Color and edge features were taken from their earlier work on vehicle matching [7]. 3-D, RGB histograms of the vehicles were compared using histogram intersection. The set of edge points within the fitted template were compared to those of other vehicles using a modified Hausdorff distance between point sets.

Zeng and Crisman [8] also used RGB histograms to match vehicles between two sites. Color information was used to track vehicles from one camera site to another to measure the time it takes to travel from point to point. The histograms were modified to compensate for differences in illumination between the two sites. Statistics based on bin-to-bin differences were used to compute the similarity between vehicles.

The literature on vehicle feature extraction (excluding license plate reading) works on general features like shape and color to classify and compare vehicle images. While these may be effective for reducing the search space to a few similar images, it is not enough to determine if two images are that of the same object. A more detailed image comparison would be needed toward this goal.

Detailed object comparison makes sense when corresponding object parts are compared. The images of the compared objects show a subset of these parts. The object observed in the scene (Figure 10) are almost in the same pose and corresponding objects parts are visible as image features. The logical first step would then be to align the object images with the goal of moving the corresponding image features in the same spatial location.

In searching for the most similar objects, the natural approach would be a *serial search* where the query image is aligned with each candidate image. The candidate images can then be ranked according to their spatial differences with the query.

Image registration methods seek to find the set of transformations necessary to align images. Most of work on registration [9] deal with alignment of images of the same object or scene. While work on aligning multiple images has been done in shape learning [10] and super-resolution from video [11], multiple-image alignment of different objects for recognizing a query object image has not been realized.

1.2 Proposed Approach to Recognition

We propose a multiple-object alignment strategy coupled with a search-space reduction scheme utilizing learned feature correspondences. The main idea is that given a set of candidate images output from a general image-retrieval system, the images are aligned in parallel and *outlier objects* are iteratively removed, making the set more homogeneous and better aligned. The strategy uses a multi-object feature clustering and registration approach.

Figure [1] shows the framework for recognition. A high-recall image retrieval system utilizing general features such as color and general shape then retrieves a loose set of similar images from the database of observed vehicles. The database is thus reduced to a smaller search space.

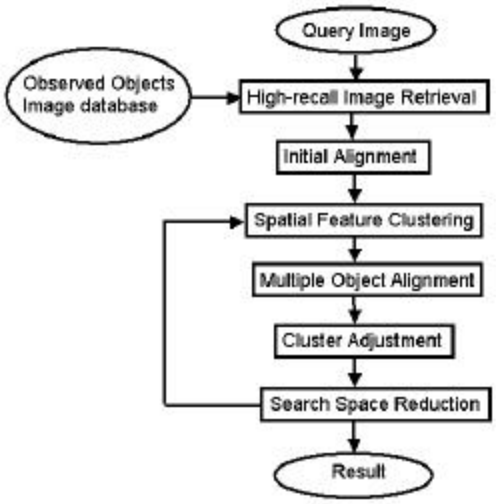

Figure 1. Overview of proposed vehicle recognition

The contribution of this work is the iterative reduction of the search space by detecting correspondences, re-alignment of the search space with the query, and removing *outlier objects*. Correspondences are detected not just between the query and each search space image but across all images in the search space. In effect, we are looking for correspondence clusters (Figure 2) where each cluster is a spatial feature common to a subset of the objects. Our hypothesis is that the outlier object will have an image with the least correspondence with the query and the rest of the search space (Figure 3). We also hypothesize that removing the outlier objects would allow a more precise re-alignment and allow a more accurate detection of correspondence clusters. Prior to this iterative process, the method starts with the retrieved set of images. An initial alignment step roughly aligns the image set to the query image.

The main challenge in this strategy is to define what is an ``outlier'' object in a set of object images. This definition is what we would seek to model using information in the correspondence clusters and image alignment results.

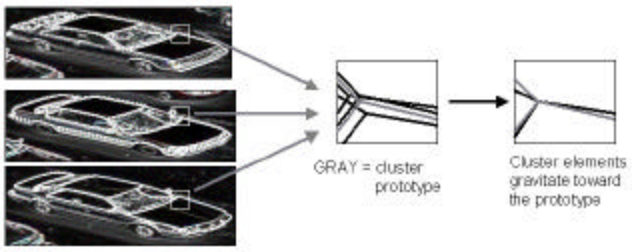

Figure 2. Detection and alignment of correspondence clusters. Common spatial features are detected in the query object image and search space object images. The common features form a cluster, with a cluster prototype as the center. During re-alignment, the object image features are driven toward the cluster prototype.

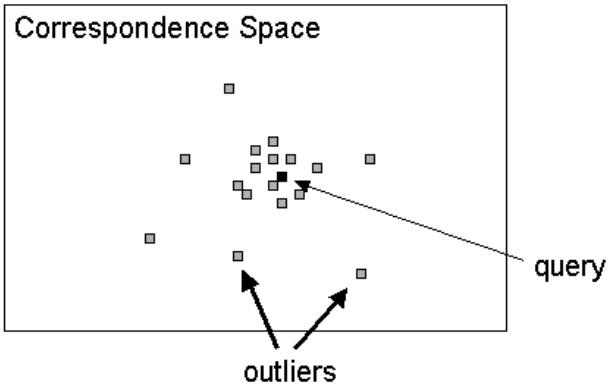

Figure 3. Analogy of ``outlier objects''. Points in this space represent objects in the reduced search space, clustered around the query. Outliers are the objects with the least correspondence with the query. Removing the outliers results in a smaller, more homogenous and better aligned search space.

2. DETAILED RECOGNITION OF VEHICLE IMAGES

The recognition method starts with a set of images retrieved by a high-recall image-retrieval system. General features such as size, color and shape can be used to achieve high-recall. For example, the query image could be a white sedan thus the initial query could be the set of ``light-colored, medium-sized" vehicle images. The set of retrieved images would be referred to as our initial *search space*.

Detailed recognition is done by aligning the images in the search space and iteratively reducing it by removing outlier objects. At each iteration, the objects in the search space become more homogenous and more closely aligned to each other. The alignment strategy was chosen because detailed object comparison is possible when corresponding features are moved closer spatially.

Corresponding features are learned using a clustering algorithm that works spatially and across multiple images. A cluster could thus contain multiple features from multiple objects. The learned clusters are centered around the prototypes (or cluster means) which serve as the reference points toward which the cluster elements gravitate in the alignment step. The properties of the clusters are used to determine the outlier-ness of a each object.

We first define how alignment is performed. Given a set of source points $[(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)]$ and a set of corresponding reference points $[(x_1',y_1'),(x_2',y_2'),\ldots,(x_N',y_N')]$, the goal is to align each point (x,y) with its corresponding (x',y') using the following transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

where T_X and T_Y are parameters for translation in the x and y directions, and S_X and S_Y are scaling factors in the x and y directions. Instead of one factor, scaling was separated into S_X and S_Y because the aspect ratio of the vehicle image varies with the vehicle's lateral position on the road. Our chosen view of the scene is above and beside the road which made the scaling separation necessary. For some views, a single scaling factor would suffice.

A set of source points is aligned to a set of reference points by estimating the transformation parameters defining the alignment. Each source-reference pair of points represent an equation. The N equations are collected in the following matrix equation:

$$\begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ \vdots \\ x_N' \\ y_N' \end{bmatrix} = \begin{bmatrix} x_1 & 0 & 1 & 0 \\ 0 & y_1 & 0 & 1 \\ x_2 & 0 & 1 & 0 \\ 0 & y_2 & 0 & 1 \\ x_3 & 0 & 1 & 0 \\ 0 & y_3 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & 0 & 1 & 0 \\ 0 & y_N & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x \\ S_y \\ T_x \\ T_y \end{bmatrix}$$

$$R = M * P$$

where R is the reference and M is the source. P is the transformation parameter vector which can be estimated using Least Squares:

$$P = (M^t M)^{-1} M^t R$$

If the source points are transformed using the estimated P vector, the source points may not be perfectly aligned with the reference points. The alignment error measures this mis-alignment:

$$AlignmentError = \sum_{i=1}^{N} \sqrt{(x_i^t - x_i')^2 + (y_i^t - y_i')^2}$$

where (x_i^t, y_i^t) is the transformed source point and (x_i', y_i') is the corresponding reference point.

2.1 Feature Extraction and Initial Alignment

Image gradients are stable image features. Under verying illumination conditions, the image gradients remain significant. In the case of a vehicle, the structural edges produce image gradients useful for aligning and differentiating objects. We used a Sobel edge detector and a gradient magnitude threshold that sufficiently highlights the structural edges of the object.

Given the retrieved set of images, edge features are detected to produce a point set for each image. Each point set is roughly aligned to the query's point set prior to the iterative search space reduction. Nine points are computed from the spatial distribution of the edges. The three horizontal coordinates are the 25^{th}, 50^{th}, and 75^{th} percentile in the X-coordinate spatial distribution. The three vertical coordinates are also computed as the 25^{th}, 50^{th}, and 75^{th} percentile of the y-coordinates. Figure [4] shows the nine points for an object image.

For each image M in the search space (the retrieved image set), the transformation parameters are estimated by aligning the nine source points of M with the nine target points of the query image. The estimated parameters are then used to transform the edge image of M, thus roughly aligning it with the query edge image.

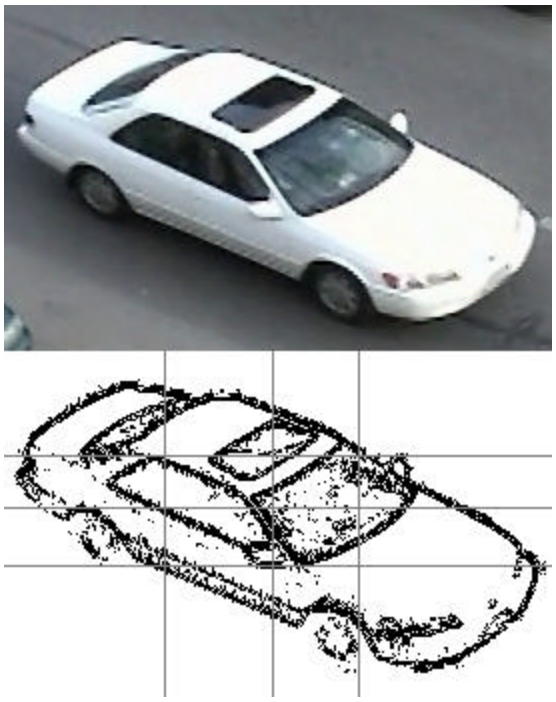

Figure 4. Nine points for initial alignment. The three horizontal coordinates are the 25th, 50th, and 75th percentile in the X-coordinate spatial distribution. The three vertical coordinates are computed in the same way.

2.2 Detection of Correspondence Clusters

Once the objects are initially aligned with the query, the iterative loop starts. The purpose of the initial alignment is to bring the objects' corresponding features close enough to start the convergence of multi-object alignment. When sufficiently aligned, features common to some objects lie in close spatial proximity. When the objects are superimposed, the common features appear as clusters in image coordinates.

A K-means clustering algorithm is employed to detect features common to objects, including the query. The coordinates of the edge features of all the object images are gathered in one list and the K cluster prototypes are allowed to gravitate toward significant edge clusters. The number of cluster prototypes K is made sufficiently large to detect major features as well as small and isolated features. Each cluster prototype is computed as the mean (x,y) coordinate of the edge features belonging to that cluster.

2.3 Re-alignment of Object Features and Cluster Prototype Adjustment

The purpose of re-alignment is to bring correponding features, which are detected by the clustering algorithm, closer together. With each iteration, the goal is to more accurately detect feature correspondences and make the remaining objects better aligned.

Upon convergence of the of the K cluster prototypes' locations, each object is aligned with the cluster prototypes. For a given object, the edge features are allowed to gravitate toward their respective cluster prototypes. The transformation parameters are estimated using the edge feature coordinates as the source points and the cluster prototype coordinates as the reference points.

The transformation of the objects re-positions the elements of the clusters and renders the prototype locations invalid. Thus the cluster prototypes need to be adjusted to the new feature locations. This is done by allowing the K-means algorithm to run a few more iterations to allow the prototypes to converge.

2.4 Search Space Reduction by Outlier Removal

The initial retrieved set of images is reduced iteratively by removing *outlier* objects. We define an outlier as the object with the least correspondence with the query object. Each cluster is observed and the amount of correspondence of a candidate with the query is computed and summed over all clusters.

Let C_{ji} be the set of feature points belonging to object j and classified into cluster i. An object's score is computed as:

$$ObjScore(D_j) = \sum_{i=1}^{K} ClusterScore(C_{ji})$$

where ClusterScore(C_{ji}) is the amount of correspondence, within cluster i, between object j and the query object. This is defined as:

$$ClusterScore(C_{ji}) = \min\left(\frac{|C_{ji}|}{N_j}, \frac{|C_{qi}|}{N_q}\right)$$

where N_j and N_q are the number of feature points in object j and the query, respectively.

The min() operation acts as an intersection of the two objects within a certain spatial locality. A high min() means there is strong correspondence in that cluster's location. In each iteration, the object with the minimum ObjScore() is declared as outlier and removed from the search space.

3. AN EXPERIMENT

An experiment was performed to test the iterative reduction scheme. A 10-minute video of a street scene was captured on a sunny day. To create a vehicle recurrence event, a friend is asked to drive his white sedan three times through the street scene in seven-minutes intervals . Prior to the third drive-through, four black stickers are attached to the roof and the shaded side of the car. Figure 5 shows the white sedan in the three drive-throughs. Using the second occurrence of the white sedan as the query object, the other two occurrences should be among the last ones to be removed from the search space.

The output of an image-retrieval system is simulated by manually picking out all the light-colored vehicles that crossed within the 10-minute interval. Nineteen vehicles were picked, including the the first and the third drive-through of the white sedan. The second drive-through is the query object. The image of the white sedan is about 400x200 pixels.

Figure 10 shows the history of outlier object removal. Starting with nineteen vehicles, the search space is iteratively reduced to one car. The first drive-through of our white sedan was the last object in the search space, with the black-stickered sedan coming in second to the last. The removal sequence starts with vehicles that are least similar, and ending with the most similar.

Figure 6 shows the ObjScore() (defined in Section 2.4) of the removed object at each iteration. The trend shows an increase in the amount of per-cluster correspondence as the search space is reduced. With each iteration, the trend shows the search space's increase in similarity with the query. Intra-cluster deviation, defined as the standard deviation of all feature point distances from their respective cluster prototypes, is used to measure the variation of features among the search space objects. Figure 7 shows a general decrease of the intra-cluster deviation as the search space is reduced. The decrease of feature variation around the clusters may be attributed to fewer, more similar, and aligned set of objects.

A good alignment scheme allows a more accurate detection of common feature because corresponding features are brought closer to each other. To characterize the performance of our alignment method, we define the average alignment error as the per-object average distance of the feature points from their respective cluster prototypes. This is computed after the re-alignment of objects' feature points. Figure 8 shows the decrease of this alignment error as the search space becomes smaller. For each iteration, the maximum alignment error among all objects is shown in Figure 9. Both graphs show continuous improvement of the alignment of objects as the search space is reduced.

Figure 5. Three passes of the same car. A friend is asked to drive the same car through the imaged street scene. The second drive is used as the query. On the third drive, four black stickers are attached on the roof and side.

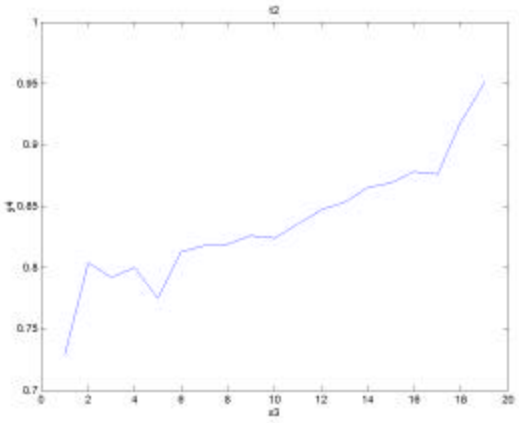

Figure 6. Outlier score vs. Iteration. In each iteration, the score of the removed object is recorded. The trend shows increasing correspondence with the query.

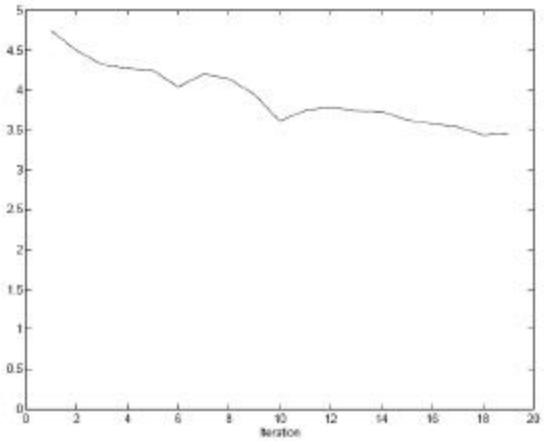

Figure 7. Intra-cluster standard deviation vs. Iteration. The shrinking clusters indicates smaller feature variation among objects.

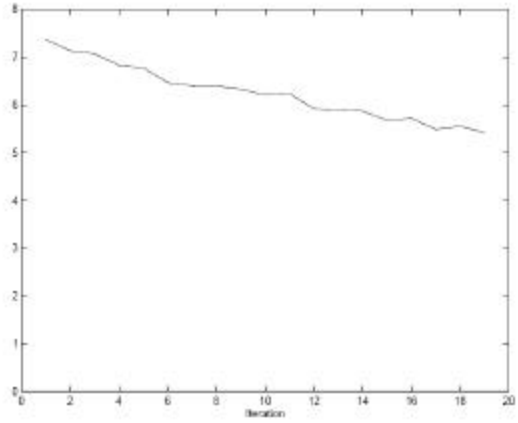

Figure 8. Average object alignment error vs. Iteration. Average error decreases, indicating closer alignment of search search.

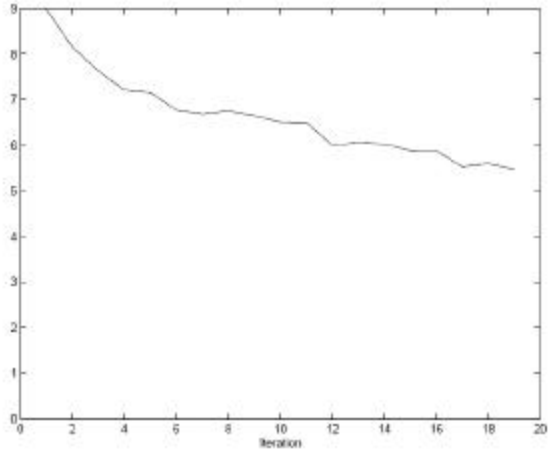

Figure 9. Maximum object alignment error vs. Iteration. At each iteration, the maximum error among objects is recorded. The decreasing error shows closer alignment of objects.

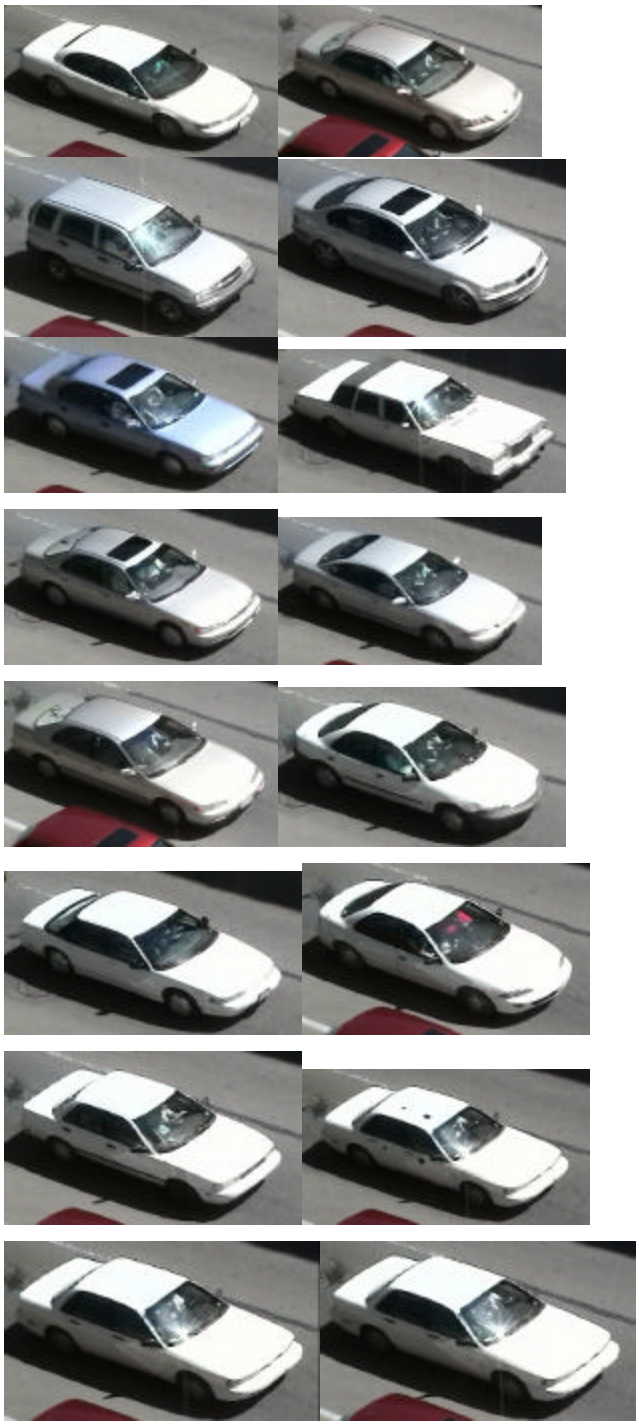

Figure 10. Outlier removal history. The sequence of outlier removal is shown top to bottom, left to right. For comparison, the query is shown in the last image. The two occurences of the same white sedan were the last in the history. The white sedan with the black sticker was removed before the one without the sticker.

4. CONCLUSION AND POSSIBLE EXTENSIONS

The feature clustering and alignment framework worked well for recognizing vehicles recurring in an outdoor street scene. Our simple experiment proved the feasibility of recognizing particular objects even among other very similar objects. The framework can be useful in outdoor surveillance applications like suspicious activity detection and example-based queries.

The feature points used are produced by an edge detector. Ideally, only a few salient points are needed but selecting them under unpredictable outdoor conditions would be a challenge. 2-D and 3-D object models (like those used in [4] and [6]) can use known object model to guide the detection of salient features. Features need not be limited to points but can be extended to higher features like lines, islands and polygons. The challenge will be to define the clustering criteria and distance measures. Our work is also limited to a transformations of scaling and translation. This can be extended to 3-D transformations since the objects are in 3-D.

5. REFERENCES

[1] Cui, Y.T. and Huang, Q., "Character extraction of license plates from video," in *Proc. IEEE Conf. On Computer Vision and Pattern Recognition, 502-507 (1997).*

[2] P. Comelli, P. Ferragina, M. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 4, 790-799 (1995).

[3] R.T. Collins, A.J. Lipton, H. Fujikoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," in *Proceedings of the IEEE*.

[4] T.N. Tan and K.D. Baker, "Efficient image gradient based vehicle localization," *IEEE Transactions on Image Processing*, vol. 9, no. 8, 1343-1356 (2000).

[5] G. Fung, N. Yung, and G. Pang, " Vehicle shape approximation from motion for visual traffic surveillance," in *Proc. IEEE Conference on Intelligent Transport System*, 608-613 (August 2001).

[6] M-P.D. Jolly, S. Lakshmanan, and A.K. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3 (1996).

[7] M-P. Dubuisson, A.K. Jain, and W.C. Taylor, “A vision-based vehicle matching system,” in *Intelligent Vehicles Symposium*, 266-271 (1994).

[8] Nan Zeng and J.D. Crisman, “Vehicle matching using color,” in *Proc. IEEE Conf. Intelligent Transport System*, 206-211 (1997).

[9] Lisa Brown, “A survey of image registration techniques,” *ACM Computing Surveys*, vol. 24, no. 4, 325-376 (1992).

[10] N. Duta, A. Jain, and M-P. Dubuisson-Jolly, “Automatic construction of 2-D shape models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, 433-446 (2001).

[11] S. Lertrattanapanich and N. Bose, “High resolution image formation from low resolution frames using Delaunay triangulation,” *IEEE Transactions on Image Processing*, vol. 11, no. 12, 1427-1441 (2002).