EFFICIENT ALIGNMENT FOR VEHICLE MAKE AND MODEL RECOGNITION

Ninad Thakoor, Bir Bhanu

Center for Research in Intelligent Systems,
University of California, Riverside, Riverside, CA 92521, USA
email:{nthakoor,bhanu}@cris.ucr.edu

ABSTRACT

This paper presents a make and model recognition system for passenger vehicles. We propose a two-step efficient alignment mechanism to account for view point changes. The 2D alignment problem is solved as two separate one dimensional shortest path problems. To avoid the alignment of the query with the entire database, reference views are used. These views are generated iteratively from the database. To improve the alignment performance further, use of two references is proposed: a universal view and type specific showcase views. The query is aligned with universal view first and compared with the database to find the type of the query. Then the query is aligned with type specific showcase view and compared with the database to achieve the final make and model recognition. We report results on database of 1500 vehicles with more than 250 makes and models.

Index Terms— Make and model recognition, Alignment, Reference views

1. INTRODUCTION

The ability to detect make and model of a vehicle has important application in areas of surveillance, controlled access, law enforcement, traffic monitoring, collecting tolls, and other government and business functions. Currently, the identity of a vehicle is tied to its license plate alone, and today's automatic license plate recognition systems claim close to 100% accuracy. However, these systems have no reliable way to detect spoofing of a license plate - the use of a plate from another vehicle or alteration of the plate. If make and model information extracted from the video or images of a vehicle can be verified with information associated with the license plate, spoofing can be detected. Thus, the ability to automatically recognize the make and model of a vehicle provides the new capability to verify the identity of a vehicle with added certainty. Table 1 shows overview of some of the related make and model recognition work. The key limitation of these approaches is that they rely on the licence plate (LP) for alignment and at best account for affine deformation.

If one has a database of images of all the possible makes and models of vehicles, then in theory the make and model of a vehicle observed in a video can be found by matching it with the database. We call this database make and model recognition (MMR) database. The challenges in matching an image of an unknown vehicle with the database of vehicles are:

1. View variation: Changes in the viewpoints cause significant variations in the visual appearance of a vehicle.
2. Changing illumination: As a vehicle has to be observed in the real-world outdoor environment, inconsistencies in illumination between the stored database and observed data, which cause appearance changes, have to be dealt with.
3. Various body colors: In addition to the illumination variation, varying body colors of vehicles cause additional visual variations.
4. Large number of make and models: Given that there are hundreds of makes and models of vehicles, matching can become computationally expensive.

Our proposed system deals with these challenges at various stages. We assume that the images of the vehicles are taken from a fixed position where the vehicle pose is more or less consistent. This scenario is sufficient for most traffic monitoring applications where a vehicle is either seen from the front or back.

2. TECHNICAL APPROACH

The overview of our make and model recognition system is shown in Figure 1.

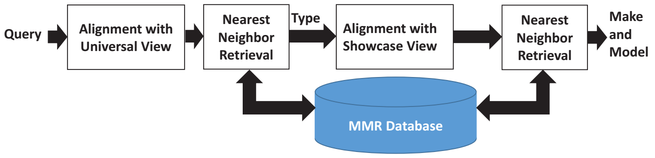

Query
Alignment with Universal View
Nearest Neighbor Retrieval
Type
Alignment with Showcase View
Nearest Neighbor Retrieval
Make and Model
MMR Database

Fig. 1. Make and model recognition system

2.1. Alignment

To compare an unknown vehicle with a vehicle in the database, i.e. to query a vehicle to a MMR database, the images have to be properly registered such that corresponding visual landmarks are aligned (e.g., taillights, brake light, logo, window, etc.). This is challenging due to perspective projection and as these landmarks are not necessarily coplanar. Simple image rectification based on affine transformation or homography is not effective in this case. Instead, a method that allows local deformation has to be used. State-of-the-art methods such as SIFTflow [6] allow for such deformation. However, SIFTflow is computationally expensive as it tries to solve 2D image alignment problem through belief propagation. In addition, the alignment results tend to overly deform the vehicles. We propose an efficient two-step alignment method which aligns rows first and then the columns. The minimization of the alignment cost is achieved as

This work was partially supported by NSF Grant 0905671 and a grant from the 3M Company.

Table 1. Related work

Publication	Approach	Pros	Cons
Petrovic and Cootes [1]	Normalization for scale and location using LP, Features: Square mapped gradient, Classifier: k-NN with dot product	Gradient information is color independent; Weighting scheme based on variance	Pixel based representation ; Features must line up; Reference LP location limits performance
Dlagnekov and Belongie [2]	Normalization for scale and location using LP (Not as strict as others), Features: SIFT, Classifier: WTA, maximum number of SIFT points matched	Real traffic videos used; affine distortion allowed	Point matching is computationally expensive, only 38 query images
Negri et al. [3]	Normalization for scale and location using LP, Features: Oriented contour points, Classifier: Weighted discriminant/ kNN	Uses positive and negative weights; Robust to partial occlusion	Needs multiple examples per class; Only a small local deformation allowed
Zafar et al.[4]	Normalization for scale and location using LP, Features: Contourlet transform, Classifier: 2D LDA + SVM	Localized features	Localized features must line up
Pearce and Pears [5]	Affine normalization using LP, Features: Harris corners, Classifiers: kNN and naïve Bayes	Recursive partitioning for representation; Local normalization	Relies on LP for alignment

two separate 1D optimization problems, making the alignment computationally efficient.

2.1.1. *Two-step Alignment*

We want to align a query image Q (Figure 2(b)) to a target image T (Figure 2(a)). For these images, dense local features such as dense-SIFT [7] can be extracted which are denoted by S_T and S_Q respectively. The row-to-row alignment cost D_r for target row r_T and query row r_Q can be computed as

$$D_r(r_T, r_Q) = \sum_{c=1}^{N_c} d(S_T(c, r_T), S_Q(c, r_Q))$$

where N_c is the number of columns and $d(\cdot)$ represents a distance measure such as Euclidean, cosine etc. A classical technique such as dynamic time warping (DTW) [8] can be applied to matrix D_r (Figure 2(c)) to find optimal alignment. We instead propose a subpixel shortest path algorithm which provides better alignment results. The algorithm is described in the next subsection. Based on the outcome of the algorithm, we align the query vertically Q_v and recompute the dense descriptors S_{Q_v}. We proceed with the horizontal alignment by computing column-to-column alignment cost D_c (Figure 2(d)) as

$$D_c(c_T, c_Q) = \sum_{r=1}^{N_r} d(S_T(c_T, r), S_{Q_v}(c_Q, r))$$

where N_r is the number of rows. After computing the optimal horizontal alignment, we generate the final aligned query Q^* (Figure 2(e)).

2.1.2. *Subpixel Shortest Path*

We explain the algorithm to align the rows of query and target image. Each possible pair of corresponding rows forms vertices V of an oriented graph ($|V| = N_r^2$) which can be represented as a square grid as shown in Figure 3. Each node can be represented as an ordered pair (r_T, r_Q) where $r_T, r_Q \in \{1, 2, \ldots, N_r\}$. A directed edge between $v_i \equiv (r_T^i, r_Q^i)$ and $v_j \equiv (r_T^j, r_Q^j)$ should only exist only if $r_T^i \leq r_T^j$ and $r_Q^i \leq r_Q^j$ to preserve spatial order of the solution. To reduce the complexity of the shortest path search, only a vertices from a small square window of size W are connected such that $(r_T^j - r_T^i) \geq W$ and $(r_Q^j - r_Q^i) \geq W$. As the amount of deformations is much smaller compared to the entire image, this approximation still performs well practically. The weight of an edge is computed by adding row-to-row alignment cost along line connecting start and end nodes of the edge and then normalizing it.

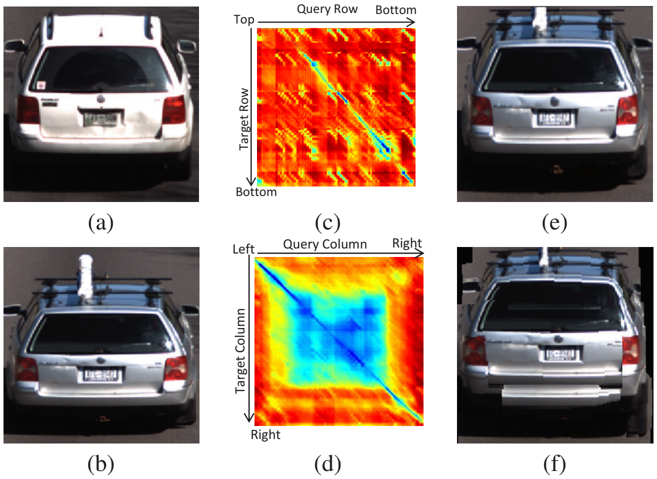
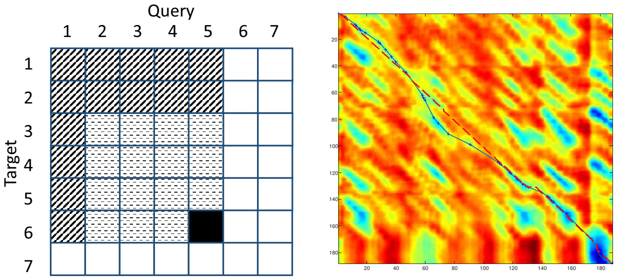


Fig. 2. Two Step alignment: (a) Target image (b) Query Image, (c) Vertical alignment cost (Hotter colors indicate higher cost) and solution (blue line), (d) Horizontal alignment alignment cost and solution, (e) Aligned query, (f) Query aligned with SIFTflow

$$\Omega_{ij} = \frac{\text{Dist}(v_i, v_j)}{|\text{Line}(v_i, v_j)|} \sum_{\forall (r_T, r_Q) \in \text{Line}(v_i, v_j)} D_r(r_T, r_Q)$$

In the square grid representation in Figure 3, one can draw a line between v_i to v_j. The function Line(v_i, v_j) represents a digital line drawing procedure [9] which returns set of vertices falling on this



(a) (b)

Fig. 3. (a) Illustration of nodes in the oriented graph with $N_r = 5$ for the shortest path algorithm. Node $(5, 6)$ shown as a dark square can terminate edges starting from nodes shown with diagonal and horizontal fill. Horizontally filled nodes fall inside a window of size $W = 3$. (b) Comparison of solutions achieved with DTW (red dotted line) and shortest path (blue solid line).

line and Dist(v_i, v_j) gives Euclidean distance between v_i and v_j on the grid.

2.1.3. *Alignment with Reference Views*

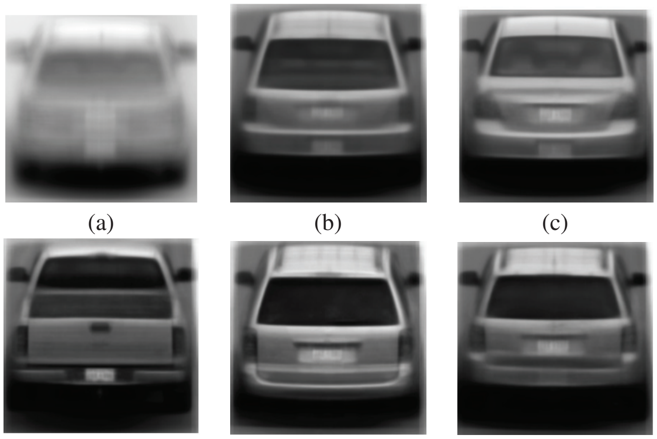
Aligning each query with hundreds of targets in the database is computationally expensive. Instead, a reference can be chosen to which both the query and target images can be aligned. We call this reference the *universal view*. All the images in the database can be pre-aligned to the universal view, and only one alignment would be needed for each incoming query. However, selecting the universal view is challenging as it has to be chosen in such a way that any vehicle image irrespective of its type can be aligned to it. In our system, we define 4 type of vehicles: sedan, pickup, minivan, and SUV. Note that additional types such as hatchback are possible but are not used as they are not widespread. We use iterative approach for generation of the universal view. During the first iteration, the universal view is nothing but the average of various vehicle images (see Figure 4(a)). In the subsequent iterations, each image is aligned to the universal view from the last iteration, and then aligned images are averaged again to update the universal view. This process produces a universal view which combines visual features from different types of vehicles (see Figure 4(b)).

As vehicles of similar type have more visual similarity, it makes more sense to create a reference for each type. We call these references *showcase views*. However, type of the query vehicle is also unknown. To deal with this, a hierarchical approach is used in which query image is first aligned to the universal view and the type of vehicle is determined by matching it with the MMR database. Then the query is aligned with the corresponding showcase view, and the make and model of the vehicle is determined by searching the MMR database. The proposed hierarchical alignment and recognition significantly reduces complexity of the recognition by requiring only two alignments per query and tackles the challenge arising from the large number of makes and models.

2.2. Illumination Normalization and Description Generation

2.2.1. *Illumination Normalization*

To deal with illumination and body color changes, a normalization step has to be applied to the query image. The goal of the normalization is to throw away illumination information while retaining the visual cues such as edges, which help in recognition. There are various alternatives available for illumination normalization [10]. We perform gamma correction followed by difference of Gaussian (DoG) filtering [11]. The gamma correction removes extreme illumination changes such as specular reflections and normalizes the dynamic range of the image. DoG filtering acts as a band pass filter which removes flat areas and high frequency areas from the image.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Universal and Showcase views: (a) Universal view Iteration 1, (b) Universal view Iteration 5, (c) Sedan showcase view Iteration 5, (d) Pickup showcase view Iteration 5, (e) Minivan showcase view Iteration 5, (f) SUV showcase view Iteration 5

2.2.2. *Generation of Description*

After the illumination normalization is carried out, the query image can be compared to the database to find its type first and then the make and model. A direct pixel-to-pixel comparison is unreliable as alignment is not perfect. The cell and block partitioning scheme similar to one used by histogram of oriented gradients (HOG) [12] is combined with locally normalized Harris strength (LNHS) features [5] as the descriptors for comparison. LNHS uses Nobel's variation of corner strengths [13] as the base feature. LNHS descriptor partitions the image recursively into quadrants. If at a level of recursion A, B, C, D are the quadrants, then the feature for quadrant A is extracted as ratio of sum of Harris strength over A to sum over A, B, C, D. This scheme locally normalizes the features. This lacks fine control on the partitioning scheme and does not normalize across quadrants originating from different parent quadrants as they are non-overlapping. We adapt the cell and block structure instead. First the image is split into small cells (e.g. 16×16). Blocks are formed by combining the cells (e.g. 2×2). The feature is computed for each cell in the block by normalizing it with respect to the entire block. The block is then shifted with some overlap to slide across the entire image (e.g. shift by 1 cell at a time horizontally/vertically).

2.3. Summary of Computational Steps

Offline steps:

1. Generate the universal view.
2. Generate type specific showcase views.
3. Populate the MMR database. For each target image whose type, make and model are known, create universal view aligned image and type specific showcase view aligned image. Normalize

illumination of aligned images and store them along with type, make and model information in the MMR database.

Query steps:

1. Align the query with the universal view and conduct illumination normalization of the aligned image.
2. Determine the type of the query by finding the nearest neighbor of the query by comparing with Universal view aligned images from MMR database.
3. Align the query with the type specific showcase view and conduct illumination normalization of the aligned image.
4. Determine the make and model of the query by finding the nearest neighbor of the query by comparing with type specific view aligned images from MMR database.

3. EXPERIMENTAL RESULTS

Data: Videos were collected by setting up a camera on top of a freeway lane over several days during daytime. Vehicle ROI were detected using moving object detection approach from [14] and they were further refined by removing shadows by enforcing bilateral symmetry [15]. For each vehicle only the closest view was retained for make and model recognition. There were 1664 vehicles extracted from the videos. The ground truth for make and model was generated manually by three individuals. As some of the vehicles could not be identified, only 1505 vehicles were labeled. These vehicles represented 256 different makes and models with 173 of them with multiple examples. Each vehicle was also associated with one of the 4 types namely sedan, minivan, pickup and SUV.

Implementation: The detected ROIs were resized to 200 × 200. the universal and showcase views were generated by averaging and re-aligning for 5 iterations. During the alignment window of size $W = 10$ was chosen for shortest path formulation. During the illumination normalization DoG filter with $\sigma_1 = 1.0$ and $\sigma_2 = 4.5$ was chosen. Harris strengths were computed with implementation from VLFeat [7] for $\sigma = 2.0$. For description, the Harris strengths are split into 32 × 32 cells and normalized in 3 × 3 block. Descriptions are generated by shifting block by 1 × 1. To compare the descriptions, cosine distance is used.

Results: Figure 5 shows examples of queries and retrievals. First column shows the query images. We show two example for each type sedan, minivan, SUV and pickup. Columns 2 to 6 show the top 5 retrievals by the system. The retrievals marked by green are the correct retrievals. For the second minivan example, the system returns the correct make and model as the rank-3 candidate. The top candidates only differ by manufacturer logos and otherwise are visually identical. For the second pickup example, the system fails to return the correct make and model as the top candidate. This is primarily due to the fact that the pickup trucks appear very similar across different make and models.

Cumulative match characteristic for top-20 retrievals can be seen in Figure 6. We use LNHS description from [5] as our baseline which was computed on the resized vehicle ROIs directly. Although no alignment was used under the baseline case, the vehicles appear in very similar pose in the ROIs. Rank-1 accuracy for base line was nearly 44%. With the block based normalization, the rank-1 accuracy improves to nearly 47%. Illumination normalization further improves the accuracy to 51%. The most improvement can be seen with universal view alignment which takes the accuracy to 61%. Finally the type specific alignment achieves 66% rank-1 accuracy. Thus our approach for make and model recognition shows significant improvements over the baseline.

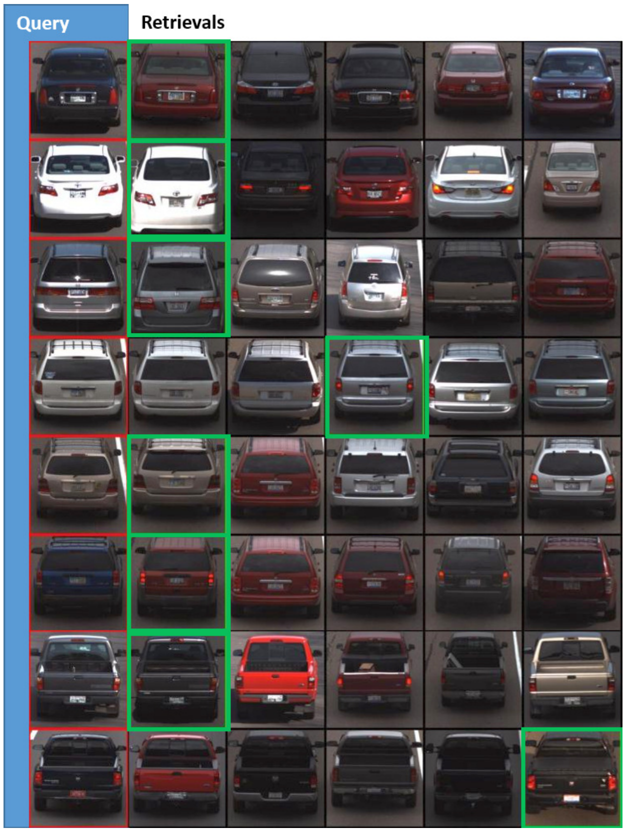


Fig. 5. Example queries and top 5 retrievals (green indicates the correct match).

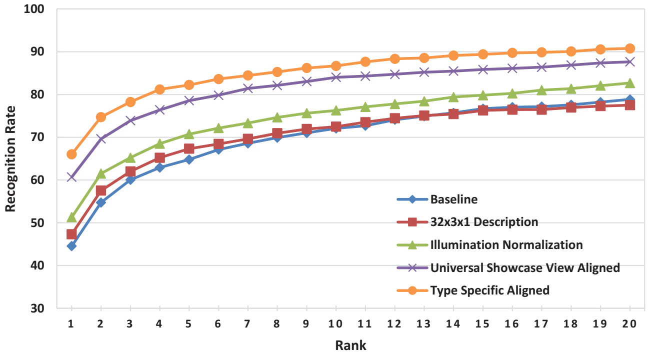


Fig. 6. Cumulative Match Characteristic

4. CONCLUSIONS

We presented a make and model recognition system in this paper. The system deals with challenges of view changes, illumination changes, body color variations, and large number of classes. For alignment we proposed an efficient 2 step process which used shortest path formulation for accurate alignment. Through use of universal view and showcase views, the alignment complexity is reduced to only two alignments per query. The proposed system outperforms the baseline significantly.

5. REFERENCES

[1] V. S. Petrovic and T. F. Cootes, “Analysis of features for rigid structure vehicle type recognition,” in *Proc. BMVC*, 2004.

[2] L. Dlagnekov and S. Belongie, “Recognizing cars,” Tech. Rep., UCSD CSE, 2005.

[3] P. Negri, X. Clady, M. Milgram, and R. Poulenard, “An oriented-contour point based voting algorithm for vehicle type classification,” in *Proc. ICPR*, 2006.

[4] Iffat Zafar, Eran A. Edirisinghe, and B. Serpil Acar, “Localised contourlet features in vehicle make and model recognition,” in *Image Processing: Machine Vision Applications II, Proc. of SPIE-IS&T Electronic Imaging*, 2009.

[5] G. Pearce and N. Pears, “Automatic make and model recognition from frontal images of cars,” in *Proc. IEEE AVSS*, 2011.

[6] C. Liu, J. Yuen, and A. Torralba, “SIFT Flow: Dense correspondence across scenes and its application,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[7] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” `http://www.vlfeat.org/`, 2008.

[8] Meinard Müller, *Information Retrieval for Music and Motion*, chapter 4. Dynamic Time Warping, pp. 69–84, Springer, 2007.

[9] Peter Shirley, Michael Ashikhmin, and Steve Marschner, *Fundamentals of Computer Graphics*, CRC Press, 2009.

[10] V. Štruc and N. Pavešić, *Advances in Face Image Analysis: Techniques and Technologies*, chapter Photometric normalization techniques for illumination invariance, pp. 279–300, IGI-Global, 2011.

[11] X. Tan and B. Triggs, “Enhanced local texture sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE conference on Computer Vision and Patter nRecognition*, 2005.

[13] A. Noble, *Descriptions of image surfaces*, Ph.D. thesis, Department of Engineering Science, Oxford University, 1989.

[14] N. Thakoor and J. Gao, “Automatic video object shape extraction and its classification with camera in motion,” in *IEEE International Conference on Image Processing*, Sept. 2005, vol. 3, pp. III – 437–40.

[15] N. S. Thakoor and B. Bhanu, “Structural signatures for passenger vehicle classification in video,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1796 – 1805, Dec. 2013.