Matching Vehicles under Large Pose Transformations using Approximate 3D Models and Piecewise MRF Model

Yanlin Guo*, Cen Rao, Supun Samarasekera, Janet Kim, Rakesh Kumar, and Harpreet Sawhney
Sarnoff Corporation,
201 Washington Rd, Princeton, NJ 08543
yguo@setcorp.com, {crao,ssamarasekera,jkim,rkumar,hsawhney}@sarnoff.com

Abstract

We propose a robust object recognition method based on approximate 3D models that can effectively match objects under large viewpoint changes and partial occlusion. The specific problem we solve is: given two views of an object, determine if the views are for the same or different object. Our domain of interest is vehicles, but the approach can be generalized to other man-made rigid objects. A key contribution of our approach is the use of approximate models with locally and globally constrained rendering to determine matching objects. We utilize a compact set of 3D models to provide geometry constraints and transfer appearance features for object matching across disparate viewpoints. The closest model from the set, together with its poses with respect to the data, is used to render an object both at pixel (local) level and region/part (global) level. Especially, symmetry and semantic part ownership are used to extrapolate appearance information. A piecewise Markov Random Field (MRF) model is employed to combine observations obtained from local pixel and global region level. Belief Propagation (BP) with reduced memory requirement is employed to solve the MRF model effectively. No training is required, and a realistic object image in a disparate viewpoint can be obtained from as few as just one image. Experimental results on vehicle data from multiple sensor platforms demonstrate the efficacy of our method.

1. Introduction

Unconstrained object recognition has become an increasingly important task in security, surveillance and robotics applications. For example, in persistent surveillance over an extended area, object association has to be carried out across videos acquired from multiple types of platforms. Due to the unconstrained conditions in viewing angle/position, illumination, occlusion, and background clutter, robust recognition is extremely challenging.

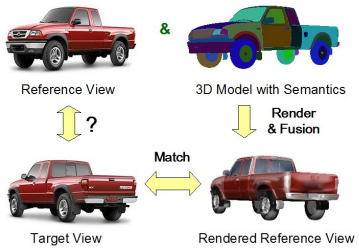


Figure 1. A compact set of 3D models are used to assist matching objects between disparate viewpoints.

A large body of work on object recognition has focused on appearance based methods, where either global or local methods have been exploited. Global methods build an object representation by integrating information over an entire image. Global methods [15] take into consideration the entire object attribute, but they are sensitive to viewpoint change, background clutter and occlusion. Local methods [6] represent images as a collection of features extracted based on local information. Recent research based on local invariant features [14, 11, 4] has demonstrated good performance on object recognition under limited viewpoint changes and occlusion. Despite the progress, these approaches still have limited success in many challenging viewing conditions. For example, in the presence of large large scale/viewpoint changes and/or occlusion, only a sparse set of distinguished features can be reliably extracted, and only a small portion of the object is covered with matched features. It is obvious that to increase the discriminative power of any recognition scheme, dense coverage is desirable since it incorporates the identifying evidence from all parts of an object. For this reason, several recent approaches attempt to increase the coverage of local features by expanding the initial set of corresponding features and integrating information from mul-

*This work was done while the first author was at the Sarnoff Corp. The current address for this author is: SET Corporation, Arlington, VA 22201.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

tiple frames [5, 16, 10]. In addition, some geometry constraints such as affine and homography transformations are employed to provide a more comprehensive representation of 3D objects.

We reason that when object domain is known, to perform well in unconstrained object recognition tasks, the explicit utilization of 3D models can largely alleviate the problem of feature matching and achieve robust object recognition under large viewpoint changes, occlusion, and background clutter. For example, in the vehicle recognition domain, many 3D vehicle models exist. Detailed 3D models provide rich constraints to match objects reliably. However, to require that an exact model is available for each instance is unrealistic. Furthermore, there can be large variations of object instances in a broad category. How to utilize a *compact* set of representative 3D models that can provide sufficient constraints for robust object recognition is the main thrust of this paper.

In this paper, we propose a robust object recognition method based on *approximate* 3D models that can effectively match objects under large viewpoint changes, partial occlusion and background clutter. Our domain of interest is vehicles, but the approach can be generalized to other rigid man-made type of objects. As shown in Fig. 1, to match an object seen from two disparate viewpoints (reference and target views), a set of 3D models that are representative for their categories are first chosen. A 3D model (from the set) that is closest to the image object is selected and its 3D poses with respect to both reference and target images are estimated. The approximate 3D model geometry, together with its poses, are utilized to transfer the object appearance features from the reference view to the target view through photo realistic rendering. Our utilization of the 3D model enables us to compute a global appearance model for each semantic part such as windows and doors of a vehicle. The semantic part ownership is used to *extrapolate* appearance information that is not visible in the reference image. A piecewise Markov Random Field (MRF) model is employed to combine observations obtained from each individual pixel and from the corresponding semantic part. A Belief Propagation (BP) method that reduces the size of required memory is used to solve the MRF model effectively. No training is required in our method, and a realistic object image in a disparate viewpoint can be obtained from as few as just one reference image. Experimental results on manufacturers' vehicle data and real data from multiple platforms demonstrate the efficacy of our method.

We review related work in Section 2. We introduce the approach in Section 3, and present experimental results in Section 4. We conclude in Section 5.

2. Literature Review

Tremendous progress has been made in recent years in recognizing objects with large variations in viewing conditions by utilizing both object appearance and geometry information [14, 11, 4]. Most methods represent object classes as collections of salient features with some invariant representations of their appearance. Geometry constraints are enforced in a loose or rigid manner to resolve appearance ambiguity and improve recognition performance. In general these methods only produce a sparse set of features that cover a small portion of the entire object, and therefore may miss some important and discriminative regions for reliable object recognition.

Most recently a flurry of research has attempted to enlarge the coverage of local feature sets while enforcing geometry constraints in a flexible fashion. Ferrari et al. [5] deal with the presence of background clutter and large viewpoint change by expanding the matching feature set after initial matched features are produced. The set of matched regions are partitioned into groups and integrated by measuring the consistency of configurations of groups arising from different model views. Savarese&Fei-Fei [16] recognize the class label and pose for each object instance by learning a model for each class. The model consists of a collection of canonical "diagnostic" parts that are viewed in the most frontal position and linked with some geometry consistency constraints. The linkage structure of canonical parts is built with multiple viewpoints. Kushal et al. [10] represent object parts as partial surface models (or PSMs) which are dense, locally rigid assemblies of texture patches.

In the model based vehicle recognition domain [13], [6] build 3D generic vehicle models with templates by projecting 2D features to 3D and clustering 3D features over the sequence of frames. [9] employs a 3D generic vehicle model parameterized by 12 length parameters to instantiate different vehicles. Line segments from the image are matched to the 2D model edge segments obtained by projecting a 3D polyhedral model of the vehicle into the image plane. An illumination model is used to handle lighting change and shadows. This method works well when enough image resolution is available. Another model-based approach is proposed in [8]. A simple sedan model and a probabilistic line feature grouping scheme are used for fast vehicle detection. The approach is more suitable for nadir (top) view detection. [18] also uses 3D CAD vehicle models and other sensor modalities for target identification. The number of vehicles of consideration is limited in their application. In [7], a quasi-rigid 3D model is used to establish dense matching from line correspondences. The scheme can reliably match objects up to $30 \sim 40^0$. The similar 3D model analysis-by-synthesis loop approaches were proposed for face recognition systems also [1, 2].

Markov Random Field (MRF) models provide a robust

and unified framework for early vision problems such as stereo and image restoration. Inference algorithms based on belief propagation have been found to yield accurate results [19, 20]. Despite recent advances these methods are often too slow for practical use. Several techniques [3] have been proposed to substantially improve the running time of loopy belief propagation.

Our approach in spirit is close to [12], where a high resolution face is synthesized from a low-resolution input using a two-step approach that integrates both a global parametric model and a local non-parametric model. However our domain of application is the matching and recognition of rigid objects with regular texture.

3. Approach

3.1. Overview of Approach

The objective of our approach is to match objects between unconstrained reference and target views. Since the viewpoint change can be quite significant, we need to utilize all the available object appearance and geometry information, and perform matching at every visible location or even beyond. We take a 3D model assisted rendering and matching scheme, and introduce a fusion step in between to improve the rendering quality before matching. The complete system includes: (1) Select the closest 3D models for the reference and target views respectively starting from some initial poses. The poses between the 3D models and images are also refined in this step. (2) From the aligned 3D model and the reference image, acquire pixel level (*local*) and region level (*global*) appearance for the 3D model, and transfer both local and global appearance into the target viewpoint. Corresponding rendering quality maps are also computed and transferred. (3) Combine the local and global appearance using a *piecewise* MRF model based fusion scheme with Belief Propagation method. Model semantic part ownership mask is naturally used to derive piecewise MRF model representation. (4) Compute a composite match measurement between the fused rendered image and the target image. Finally we decide if the two images belong to the same object or not when needed. We elaborate each step in the following sections.

3.2. Model Selection and Pose Estimation

To enable the matching of rigid objects such as vehicles with large viewpoint variations, we use 3D models. Unlike previous approaches that require precise fitting of a 3D model to each 2D image [9, 13], we associate each image with an *approximate* 3D model that is selected from a few representative model categories, and subsequently estimate the relative pose between the chosen model and the image. Our rendering approach, when equipped with a fusion scheme, is able to generate realistic rendering from any viewpoint from as few as just *one* real image.

Figure 2. 3D model examples and their representation. Each vertex of a 3D model is represented by both its position and semantic ownership information. We render each part in a unique pseudo color (first row), and the edges between parts are extracted (second row) for shape matching.

Our 3D model database consists of 11 models that are drawn from 5 vehicle categories including sedan, SUV, mini-van, pickup truck, and delivery van/bus. In most commercial 3D modelers, each vertex of a 3D model is not only represented with 3D position, but also with the semantic ownership such as "front bumper", "hood", "rear window", etc. Some of the representative rendered vehicle prototypes and their edge maps are shown in Fig. 2. Note that each semantic part of a vehicle is rendered with a unique pseudo color, so that the ownership for each pixel is easily read off in the image. The edges between different parts can also be easily extracted from the pseudo-colored rendered images or directly from the 3D models.

The initial pose of the 3D model with respect to the image is obtained either from meta-data (for cameras from moving aerial platforms) or calibration (for cameras from stationery ground platforms). Note that model selection and pose estimation are iterative processes: Better pose parameters can be obtained with better model representation, and vice versa. For each object image, our joint model selection and pose refinement process consists of the following steps:

1. Select top 3 matched models from the 11-model database by 2D matching. With initial poses, the matching is performed between the projected pseudo-colored model edge and image edge maps. Chamfer distance [17] is used for similarity measurement between edge maps. This step eliminates the majority of dissimilar models, therefore alleviates the computational burden in later steps.

2. For each of the 3 model candidates, update the scale, translation, and coarse rotation parameters for each model using discrete sampling in the pose space. Scale and translation only need to be adjusted in 2D. And we sample 3D rotation angles along three axes using $3 \times 3 = 27$ samples. The Iterative Closest Point (ICP) algorithm is used to adjust the pose parameters with edge maps.

3. Fine tune the 3D rotation parameters using gradient de-

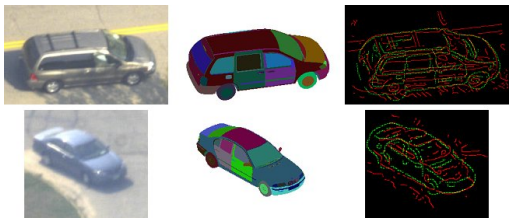


Figure 3. Model selection and pose estimation examples. Closest models (2nd col.) are selected for the corresponding images (1st col.), and their poses are refined by matching edge maps (3rd col., projected model edges are in green, and image edges are in red).

scent method. The reprojection error between the projected 3D edge points and their corresponding closest points in the 2D edge map is used in the optimization.

4. Finally, select the best model with the best pose.

Fig. 3 shows two examples of the model selection from two different images, with their projected edges (using estimated poses) overlaid on image edges.

3.3. Rendering with Approximate Models

After associating the closest 3D model and its corresponding pose with respect to a reference image, we can acquire the texture from the reference image and render it for any viewpoint (target view). However, the rendering is accurate and faithful to the true object only if the chosen 3D model is exact and all the areas in the object are visible in the reference view. Obviously this is not practical in real word applications. We present an approach that can relax this requirement by combining both pixel level and region level appearance cues to achieve as realistic a rendering as possible while attempting to transfer to as many target views as possible. The two types of renderings are dubbed as local and global rendering, respectively. In the following, we discuss how local rendering with an approximate model can achieve rendering with small residue, as well as how global rendering can complement local rendering and transfer appearance features to a large range of target viewing angles.

3.3.1 Rendering with Local Cues

A 2D point $\mathbf{x_1}$ observed from a reference viewpoint, represented by a 3×4 projection matrix $P_1 = K_1\ [I \mid \mathbf{0}]$ and center C_1, projects to a 3D line: $\mathbf{X}(\lambda^*) \simeq P_1^+\ \mathbf{x_1} + \lambda^*\ C_1$, where P_1^+ is the pseudo-inverse matrix of P_1. This line intersects with a 3D plane $\mathbf{n}^T\tilde{\mathbf{X}} = d$, with normal $\mathbf{n}$ and distance to the origin d. $\tilde{\mathbf{X}}$ & $\mathbf{X}$ are inhomogeneous and homogeneous coordinates respectively. We can compute the intersection of the line and the plane, and project the intersecting point to the target viewpoint with a projection matrix $P_2 = K_2\ [R \mid \mathbf{t}]$ as:

$$\mathbf{x_2} \simeq K_2\ (R + \frac{\mathbf{t}\ \mathbf{n}^T}{d})\ K_1^{-1}\ \mathbf{x_1}. \tag{1}$$

If the model is not accurate (but close enough), and the line intersects with a nearby plane: $\mathbf{n}^T\tilde{\mathbf{X}} = d'$, where $d' = d + \Delta d$, then the projection in the target viewpoint becomes:

$$\mathbf{x_2'} \simeq K_2\ (R + \frac{\mathbf{t}\ \mathbf{n}^T}{d} - \frac{\mathbf{t}\ \mathbf{n}^T}{d}\ \frac{\Delta d}{d})\ K_1^{-1}\ \mathbf{x_1}. \tag{2}$$

Ignoring the scaling part, the projection residual is $(\frac{\Delta d}{d})\ \{K_2\ (\frac{\mathbf{t}\ \mathbf{n}^T}{d})\ K_1^{-1}\ \mathbf{x_1}\}$, which corresponds to the parallax caused by the inaccurate plane representation. If the approximated plane is close enough to the true plane, $\frac{\Delta d}{d}$ is small, and the residual is also small, the appearance acquired in the reference image will be rendered in an approximately accurate position in the target view, and the overall rendering will be realistic.

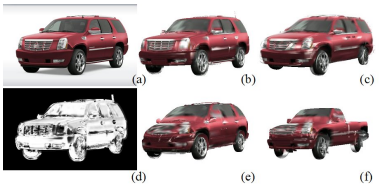


Figure 4. Local rendering examples. (a) is original image. (b) & (c) are rendered images with different 3D models from SUV category after the best poses are computed. (e) & (f) are rendered images with dissimilar models and wrong poses, the artifacts are prominent. (d) is the normalized correlation map between the rendered image (b) and original image (a).

Fig. 4 shows some rendering examples with the pixel level appearance using approximate models. The rendering quality is good when model and pose are close enough, but degenerates with either a dissimilar model or poor pose. Note that even when the selected model is not exact, for the same object instances, renderings reveal *similar* artifacts or distortion, therefore the approximate rendering is still suitable for object fingerprinting.

We should point out that we prefer a model that covers as many regions of an image as possible. We modify the distance measure in Chamfer matching to encourage matching within model region, but discourage matching outside of region.

3.3.2 Rendering with Global Cues

Using local pixel level rendering, we are able to generate realistic object images in a new viewpoint with an approximate model. However, if the viewpoint change is drastic, the appearance information will be missing for the regions that are not visible in the reference view. Even though

Figure 5. Extrapolate the appearance for the occluded part using the visible part information. Left is the reference view, middle and right columns are rendering with & without extrapolation.

Figure 6. Obtain global semantic part level appearance from image segmentation. For each part segment(such as the red box for the front window), compute the main mode of the color from all the intersecting segments from real image segmentation (left) and assign the appearance. Right column is a global rendering example.

multiple viewpoints can be used to provide better coverage, many times either only one reference image is available, or the viewpoint variation is limited within multiple reference images. To effectively address this problem, we take advantage of the symmetric nature of the vehicular object, as well as the availability of semantic ownership representation in the model. For example, the left and right windows usually have the same appearance. Parts with the same semantic labels (for example, door 1 and door 3) should also look similar. In addition, for the area of a semantic part that is not visible, it should obtain its color from the visible area of the same part. Even though this assumption is not always true, in situations where there no other information can be derived from the reference view, this is the best we can do to *extrapolate* the object appearance. We shall see in the experiments section that *matching with appropriate extrapolation is better than no extrapolation*. The observation can also be applied to the majority of man made objects. One example is shown in Fig. 5. We need to generate a back-side (*bs*) view of the red SUV from the front-side (*fs*) view. The appearance in the back is missing in the *fs* view, but can be "hallucinated" using symmetry and semantic ownership information.

To compute the appearance for the region corresponding to each semantic part, we first perform color segmentation for the original image and the pseudo-colored rendered model image, as shown in Fig. 6. For each pseudo-colored segment, we find all the intersecting segments in the real image (such as the segments in the red box for the front window part shown in Fig. 6). We assign the major mode computed from all the segments as the color for the semantic region. Rendering using the region level appearance for the same vehicle in Fig. 4 is shown in the right column of Fig. 6. The quality of this type of rendering depends on the level of detail of the model, many times it is quite coarse, as is obvious in the grill part of the vehicle in this example.

3.3.3 Rendering Quality Measurement

It is obvious that artifacts exist in either local or global rendering. We need to have some criterion that can tell which rendered pixels are faithful to the original values. The criterion is also utilized later on to combine the rendering with both cues. We use the normalized correlation between the original reference image and the rendered image. We also create a gray level correlation image and texture map the 3D model with this image, and render the correlation map in the target viewpoint. One example is shown in Fig. 4.

3.4. Fusion of Global and Local Rendering

3.4.1 Motivation

The 3D model based rendering and matching approach described in Section 3.3 will render the exact appearance for each pixel of the object in a new viewpoint only if: (1) For any vehicle image, an exact 3D model is included in the model database and that model is correctly selected; (2) The poses for all views are accurately computed; (3) The corresponding pixel is visible in the original view for every visible pixel in the new view; and finally (4) The 3D models have sufficient resolution. Obviously not all of the above conditions can be satisfied in an unconstrain environment. For the computational efficiency, for each image, we can only afford to select an approximate model from a list of 10 - 20 candidates out of all the available models, which might be in the order of thousands or more. Subsequently, poses computed based on an approximate model cannot be precise. Moreover, error can occur in the pose estimation even with the exact model. The rendering error from an inaccurate model or pose is not that prominent in the inside area of an object, as pointed out in Section 3.3, but becomes severe in the border of the object. The rendering error caused by the foreground occlusion such as trees is more prominent in the inside area, as shown in Fig. 7. In both situations, we need a scheme to compensate for both types of artifacts. In addition, we need a scheme to fill in the appearance as realistic as possible for the pixels that are occluded in the original view but become visible in the new view. MRF model provides a principle way to account for all the aforementioned factors, and approximation inference algorithm such as Belief Propagation (BP) exist to solve the MRF model.

Despite of recent advances, solving MRF models are still computationally demanding especially in the case when a relatively large number of possible labels are required. We propose to take advantage of the semantic part based model representation, and adopt a piecewise algorithmic technique that substantially improve the running time and memory usage of belief propagation for solving the realistic rendering

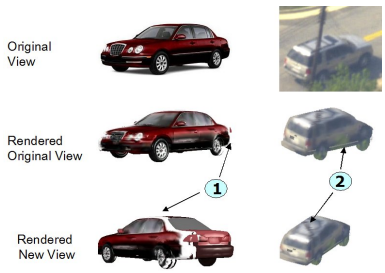


Figure 7. Examples of rendering artifacts caused by inter-object and intra-object occlusion. Left: Inaccurate pose in reference view can cause "hollowing" artifacts in target view. Right: Occlusions by foreground objects can cause "smeary" artifacts in both reference and target views.

problem.

As shown in Fig. 8, we need to estimate the set of labels (color/intensity values) that are modeled as random variables $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ based on the observations $\mathbf{y} = \{y_1, y_2, ..., y_N\}$, where $\mathbf{x}$ & $\mathbf{y}$ exist in a grid $\mathcal{P} = \{1, 2, ..., N\}$ with a 4-connected neighborhood system $\mathcal{N} = \{\mathcal{N}_{p,q}\}$. The observations $y_p's$ come from two sources based on the local and global rendering methods. The first type of observation is the local image color/intensity itself I_l, and the second type is from the average semantic part color I_g. The possible label set $\mathcal{L}$ in our case is a collection of piecewise label assignments $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_M\}$, where $\mathcal{L}_m$ is the m-th semantic part of the model, and M is the total number of parts. For each $\mathcal{L}_m$, the number of possible assignments can be limited, i.e, $\mathcal{L}_m = I_g + \{1, 2, ..., K\}$, and $K = 16$ in our case, which is much smaller than the standard 256 label assignment problem. The optimal solution can be found by maximizing the following a posterior probability (MAP):

$$x^* = \arg\max_{x_{p}, p \in \mathcal{P}} \; p(x_1, x_2, .., x_N | y_1, y_2, ..., y_N). \quad (3)$$

That is equivalent to maximizing:

$$\propto \Pi_{p \in \mathcal{P}} \; p(y_p | x_p) \; \Pi_{p,q \in \mathcal{N}} \; p(x_p, x_q). \quad (4)$$

The negative log-likelihood of Eq. 4 corresponds to minimizing the following energy:

$$E(\mathbf{x}) = \sum_{p \in \mathcal{P}} \alpha_p \; E_D(x_p) + \sum_{p,q \in \mathcal{N}} \beta_{p,q} \; E_S(x_p, x_q). \quad (5)$$

$E_D(x_p)$ consists of the following two terms:

$$E_D(x_p) = \alpha_l \; (x_p - I_l)^2 + \alpha_g \; (x_p - I_g)^2, \quad (6)$$

where $\alpha_l = const1 \; * \; corr$, and $\alpha_g = const2 \; * \; (1 - corr)$. And $corr$ is the normalized correlation value at site p, and $const1$ & $const2$ are constants. If the rendering quality using local pixels is good (usually due to accurate model and pose estimation and minimum occlusion), the correlation value is large, we trust the local rendering more. Otherwise, we resort to the global rendering assignment.

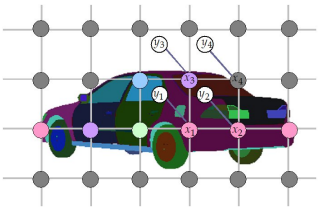


Figure 8. Piecewise MRF model for fusion of global and local rendering. Note that the sites belonging to the same semantic part (shown in same color) have the same "base" appearance, therefore the label assignment is piecewise and becomes tractable.

The smoothness term is defined as:

$$E_S(x_p, x_q) = \beta_{p,q} \; (x_p - x_q)^2 \quad (7)$$

where $\beta_{p,q} = const3 \; * \; f_{p,q}(corr)$. $const3$ is a constant, and $f_{p,q}(corr)$ takes the following form:

$$f_{p,q}(corr) = \begin{cases} 20 & if(part(p) = part(q) \;\; \& \;\; corr < \sigma) \\ 2 & if(part(p) = part(q) \;\; \& \;\; corr \geq \sigma) \\ 10 & if(part(p) \neq part(q) \;\; \& \;\; corr < \sigma) \\ 1 & if(part(p) \neq part(q) \;\; \& \;\; corr \geq \sigma) \end{cases} \quad (8)$$

where $part(p)$ & $part(q)$ are the part ownerships for site p & q, and σ is the correlation threshold. This smoothness term encourages pixels from the same semantic part to have small appearance variation, and also encourages more diffusion when the rendering quality using local image intensity is poor (small σ).

3.5. Match Measurement

After rendering an object to the target viewpoint, it is imperative to exploit as much information as possible to match it with the real target image. A combination of the following measurements that encode different aspects of an object are used: (1) Color correlogram, (2) Chamfer distance, (3) Normalized correlation. The combination utilize both appearance and geometry information at local and global levels for robust object matching.

4. Experimental Results

4.1. Performance Evaluation Methodology

We evaluate our algorithm for vehicle images acquired from manufacture vehicle catalog, as well as real data captured with sensors from both aerial platform and ground platform. The database has a wide variety of vehicle models with different colors, shapes and resolution.

Our experimental setup is designed to test the following aspects of the algorithm: (1) Overall matching performance under large pose change. (2) Comparison w/o model assisted matching methods. (3) Comparison with local rendering vs. local + global rendering. (4) Comparison with simple combination of local and global rendering vs. MRF based fusion. (5) Performance with approximate and accurate pose estimation.

For each set of experiments, we conduct a large number of trial tests. Each trial contains 1 query and N ($= 2 \sim 7$) learning sequences (images), where the targets in the learning sequences are all distinct, and one of the learning sequences contains the same object (but from a different sequence) as the query sequence. A trial outcome is considered correct if the highest score among the N scores corresponds to the learning sequence that contains the same object as the query sequence. The performance score computed as probability of correct association, P_{CA}, is defined as the number of correct outcomes divided by the number of trials.

4.2. Comparison: With/Without Model Assistance

4.2.1 Stationary Ground Platform

We test our algorithm on a collection of 54 vehicles on a distributed non-overlapping ground camera system. There are large illumination, scale, and aspect changes among cameras. The resolution varies from 130×70 to 300×160 pixels. For $N = 2$, our model-assisted method has achieved P_{CA} = 90.00%.

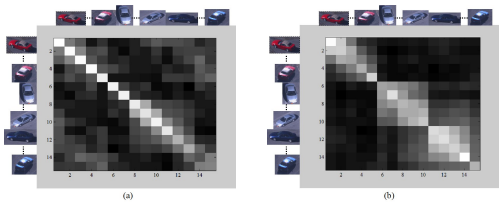


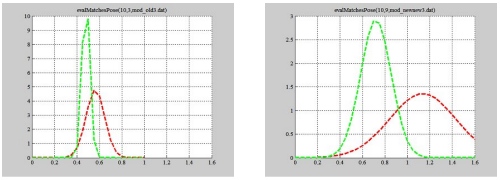
Figure 9. Similarity matrices for model-free (left) and model-assisted (right) methods. See text for detail.

4.2.2 Moving Aerial Platform

To test matching performance with large pose change, we use a seven vehicle data set from a moving aerial platform. The pose variation of each vehicle is around $90 \sim 120^0$. The Ground Sampling Distance (GSD) is about 0.4 - 0.6 cm/pixel. Fig. 9 shows the similarity matrix for model-free method (left) and our model-assisted method (right) for three sedans, and two of them have similar color. Each row of the matrix is a query, and each column of the matrix is a learning sequence. The three distinct bands of rows and columns in the right image correspond to the three different vehicles, illustrated by the sample image chips. Brighter matrix elements indicate higher likelihood scores. An ideal similarity matrix would have a block diagonal structure with consistently high scores on the main diagonal blocks and consistently low scores elsewhere. In the left image, there is no distinct diagonal block structure, while the right image demonstrates distinct diagonal block structure.

Another instructive way to contrast the performance of the two algorithms is to examine the distribution of similarity scores conditioned on when the learning and query sequences contain the same object versus different objects, P_{same} versus P_{diff}. Ideally, the distributions should be well separated, in order to reliably discriminate between the correct and incorrect matches. Fig. 10 shows that the separation between same and different object distributions is weak for model-free method (left) and significantly better for the model-assisted method (right).

The correct association performance for this dataset is P_{CA}= 83.52% for the model-free method, and is improved to P_{CA}= 97.33% for the model-assisted method.



(a) Pca = 83.52% (b) Pca = 97.33%

Figure 10. P_{same} & P_{diff} for model-free (left) and model-assisted (right) methods. X-axis is matching score. Y-axis is P_{CA}.

4.3. Comparison: Local Rendering vs. Local + Global Rendering & MRF vs. Simple Combination Scheme

To investigate different aspects of our algorithm, we collect a set of vehicle images from on-line manufacture car catalogs. We choose 7 vehicles from SUV, Mini-Van, sedan, and pickup truck categories. The mean resolution is around 400×250. To minimize the influence of color feature, we include many cars with similar color, with a subset shown in Fig. 11. Each vehicle is captured at 5 viewpoints: front (*f*), front-side (*fs*), side (*s*), back-side (*bs*), back (*b*). For comparison, we also manually choose three models (a sedan, a SUV, and a pickup), and use them to obtain initial approximate calibration data for each vehicle per view. Fig. 12 demonstrates the model selection, pose estimation, and rendering results. Fig. 13 is designed to test the following algorithms: (1) Use *local* rendering only to match vehicles, no adjustment of model or pose. (2) Use both global and local rendering, but with simple combination, no adjustment of model or pose. (3) Use both global and local rendering, with MRF based combination, no adjustment of model or pose. (4) Use both global and local rendering, with MRF based combination, and automatically select the best model (from the 11 model database) and refine its pose. The x-axis

Figure 11. A subset of manufacture vehicle examples. Many vehicles have similar color.

Figure 12. Model selection, pose estimation, and rendering results for two pickup trucks. Col. 1 & 3: Original views; Col. 2 & 4: Model edges projected on image edges, a Mazda B-Series is chosen in both cases; Col 5: Rendered images from Col. 1 to 3.

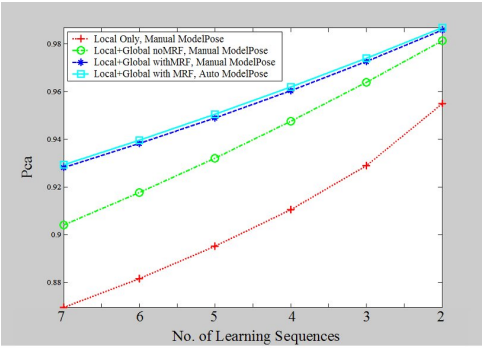


Figure 13. Manufacture vehicle matching performance. X-axis is the number of learning sequences N. Y-axis is P_{CA}.

is the number of learning objects $N = 7 \sim 2$, and y-axis is P_{CA}. We can see that rendering only at local pixel level (red) is not sufficient, both local and global level rendering is necessary. The simple combination (green) takes color value from global rendering whenever the local rendering quality is bad or it is invisible in the reference view. Simple combination is faster, but it lacks the fusion step in the MRF model (blue) to assure smoothness for each semantic part and also to fill in missing regions. Given enough resolution, our algorithm (cyan) is able to automatically select the best model, and compute its best pose, and the matching performance surpasses the one that uses manually selected model and calibrations. Altogether, our model assisted approach can achieve over 90% P_{CA} for 1 to 7 matching, and over 98% P_{CA} for 1 to 2 matching with pose change up to 90^0 for the manufacture car data set.

5. Conclusion

We propose an approach that can match vehicles over large viewpoint changes with the assistance of a compact set of 3D models. With approximate models and poses, we are able to render objects at pixel lever with small residue. We use symmetry and semantic ownership to render objects at region level. An piecewise MRF model with Belief Prorogation is used to combine rendering with both cues and achieve robust vehicle matching under large viewpoint changes.

References

[1] V. Blanz, P. Grother, P. Phillips, and T. Vetter. Face recognition based on frontal views generated from non-frontal images. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:454–461 vol. 2, 20-25 June 2005.

[2] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, Sept. 2003.

[3] Felzenszwalb and Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[5] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 4, 2006.

[6] N. Ghosh and B. Bhanu. Incremental vehicle 3-d modeling from video. In *ICPR*, Hong Kong, August 2006.

[7] Y. Guo, S. Hsu, H. S. Sawhney, R. Kumar, and Y. Shan. Robust object matching for persistent tracking with heterogeneous features. *EEE Trans. PAMI*, 29(5), 2007.

[8] Z. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *ICCV*, 2003.

[9] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3), 1993.

[10] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *CVPR*, Minneapolis, June 2007.

[11] S. Lazebnik, C. Schmid, and J.Ponce. Semi-local affine parts for object recognition. In *BMVC*, volume 2, 2004.

[12] C. Liu, H. Y. Shum, and C. Zhang. A two-step approach to hallucinating faces: Global parametric model and local nonparametric model. In *CVPR*, 2001.

[13] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank. 3-d model-based vehicle tracking. *IEEE TIP*, 14(10), 2005.

[14] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, Corfu, Greece, 1999.

[15] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1), 1995.

[16] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, Rio De. Janeiro, Brazil, October 2007.

[17] Y. Shan, H. S. Sawhney, and R. Kumar. Vehicle identification between non-overlapping cameras without direct feature matching. In *ICCV*, Beijing, China, 2005.

[18] M. R. Stevens and J. R. Beveridge. Using multisensor occlusion reasoning in object recognition. In *CVPR*, Puerto Rico, June 1997.

[19] J. Sun, H. Y. Shum, and N. Zheng. Stereo matching using belief propagation. *IEEE Trans. PMAI*, 25(7), 2003.

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *ECCV*, Graz, Austria, May 2006.

Acknowledgement

This work was funded in part under an ONR project Contract No. N00014-07-C-0219.