Robust Object Matching for Persistent Tracking with Heterogeneous Features

Yanlin Guo, *Member*, *IEEE*, Steve Hsu, *Member*, *IEEE*, Harpreet S. Sawhney, *Member*, *IEEE*, Rakesh Kumar, *Member*, *IEEE*, and Ying Shan, *Senior Member*, *IEEE*

Abstract—This paper addresses the problem of matching vehicles across multiple sightings under variations in illumination and camera poses. Since multiple observations of a vehicle are separated in large temporal and/or spatial gaps, thus prohibiting the use of standard frame-to-frame data association, we employ features extracted over a sequence during one time interval as a vehicle fingerprint that is used to compute the likelihood that two or more sequence observations are from the same or different vehicles. Furthermore, since our domain is aerial video tracking, in order to deal with poor image quality and large resolution and quality variations, our approach employs robust alignment and match measures for different stages of vehicle matching. Most notably, we employ a heterogeneous collection of features such as lines, points, and regions in an integrated matching framework. Heterogeneous features are shown to be important. Line and point features provide accurate localization and are employed for robust alignment across disparate views. The challenges of change in pose, aspect, and appearances across two disparate observations are handled by combining a novel feature-based *quasi-rigid* alignment with *flexible* matching between two or more sequences. However, since lines and points are relatively sparse, they are not adequate to delineate the object and provide a comprehensive matching set that covers the complete object. Region features provide a high degree of coverage and are employed for continuous frames to provide a delineation of the vehicle region for subsequent generation of a match measure. Our approach reliably delineates objects by representing regions as robust blob features and matching multiple regions to multiple regions using Earth Mover's Distance (EMD). Extensive experimentation under a variety of real-world scenarios and over hundreds of thousands of Confirmatory Identification (CID) trails has demonstrated about 95 percent accuracy in vehicle reacquisition with both visible and Infrared (IR) imaging cameras.

Index Terms—Video object tracking and reacquisition, object matching, feature matching, image alignment and matching.

1 Introduction

Object tracking from aerial platforms requires data association over long periods of time. The object of interest, vehicles for the purposes of this paper, may not remain in the field of view continuously through the course of tracking. The tracked objects leave the field of view because of occlusions and inaccuracies in platform pointing directions. When the vehicles appear again, the tracker needs to verify if the currently observed vehicles are indeed the same as the ones being tracked earlier. Another important visual surveillance task requires multiple observations of the same vehicle viewed from different spatial sightings to be reliably associated. In both applications, we need to compute matching scores between a model (learning) sequence and a query sequence, assuming that frame-to-frame tracking is given as input. Several representative learning and query "object chips" are shown in Fig. 1. It is obvious that standard frame-to-frame association techniques cannot be directly applied to match the learning and query sequences in these applications because of the amount of object scale, pose and appearance change, the background clutter, and the lack of temporal and spatial continuity.

Despite a flurry of research on object matching and recognition [1], [2], [3], [4], [5], [6], [7], [8], online object fingerprinting still remains a very challenging problem because of the following reasons:

1. Limited training data is available. In contrast with traditional approaches to object identification in visual imagery, we cannot assume that every object has been modeled beforehand.
2. There can be drastic pose changes between the learning and query sequences. It is difficult to find reliable invariant feature representations because of occlusion and aspect change.
3. There can be large appearance changes. The presence of shadow and specularity makes matching even more challenging.
4. Video objects captured from various platforms and resolutions (2-20 cm/pixel typically) have to be handled.
5. It is not realistic to require that the object be accurately segmented from the background, thus object masks may not be accurate.
6. There may be multiple similar objects present at the same time.

To match objects under large pose, scale, and appearance changes and with background clutter and confusers, it is crucial to utilize as much information as possible. In this paper, we propose a novel object matching technique based on the exploitation and combination of heterogeneous

- *Y. Guo, H.S. Sawhney, R. Kumar, and Y. Shan are with the Sarnoff Corporation, 201 Washington Road, CN5300, Princeton, NJ 08543. E-mail: {yguo, hsawhney, rkumar, yshan}@sarnoff.com.*
- *S. Hsu is with Canesta, Inc., 965 West Maude Avenue, Sunnyvale, CA 94085. E-mail: shsu@canesta.com.*

Manuscript received 4 Oct. 2005; revised 3 Apr. 2006; accepted 31 July 2006; published online 18 Jan. 2007.
Recommended for acceptance by S. Baker.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0537-1005.
Digital Object Identifier no. 10.1109/TPAMI.2007.1052.

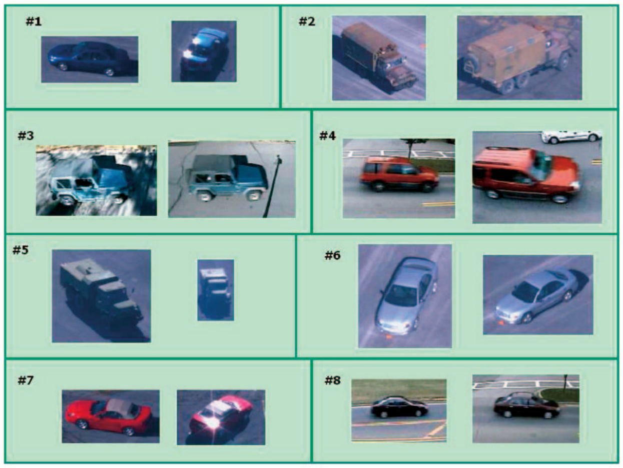


Fig. 1. Some representative object matching examples. Objects separated by a temporal or spatial gap from aerial and ground platforms are required to be matched against each other.

features: corner-like and line features for reliable geometric alignment and blob-like region features for comprehensive coverage, delineation, and matching of the object region. Each feature type is represented with suitable unique invariant representations and plays a different role in matching object geometry, appearance, and topology. Specifically, blob-like features [9], [10] provide good coverage for an object, but they are usually not suitable for image alignment due to the lack of localization accuracy. However, they can be consistently tracked across frames over a short period of time. Utilizing blob-like features *within* a sequence provides an accurate object mask for subsequence object matching across a sequence, if appropriate region descriptors and matching criteria are utilized [11], [12]. Outliers such as background clutter can be eliminated in the process of region matching. In addition, blob features can be used for overall object part configuration (topology) matching. Corner-like features and line features cannot provide sufficient extent of object coverage, but they possess good localization property and they are effective for object geometry matching (alignment), especially in *cross* sequence matching (between query and learning sequences).

Key contributions of our approach are:

1. Development of heterogeneous feature descriptors and respective match measures that utilize corner, line, and region features in different stages of object matching.
2. Development of a framework of using "within-sequence matching" using region features to obtain precise and sufficient object coverage plus "across-sequence matching" with point and lines features to achieve accurate image alignment.
3. Development of a robust blob detector and a match metric (earth mover's distance, EMD) to effectively match and track regions.
4. Development of a quasi-rigid alignment method based on invariant corner and line features to align images under large pose and appearance changes. The method avoids the explicit computation of a nonparametric 3D motion field by approximating it with a feature constrained quasi-rigid piecewise parametric motion model. It does not need explicit camera calibration, or dense reconstruction of 3D scenes. It can handle both parametric and nonparametric motion models, which are suitable for video data captured from various platforms and resolutions.
5. Development of a novel flexible template matching scheme with entropy-based adaptive scale determination in oriented energy bands.

We review the literature in Section 2 and outline our approach and present algorithm details in Section 3. Experimental results are the subject of Section 4 and we conclude in Section 5.

2 Related Work

The object matching in this paper primarily focuses on vehicle instance recognition or fingerprinting. Koller et al. [13] employed 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. This method works well when enough image resolution is available.

Feature-based object recognition methods have flourished in recent years. An extensive review of local feature descriptors can be found in [14]. A large body of work is based on the development of corner-like interest point and associated invariant description [8], [15]. The interest point finds distinctive features with precise location, but its descriptor may not be stable under large perspective change. A representative work using local region-like features is the scale-invariant feature transform (SIFT) method [4]. SIFT-like features cannot be extracted reliably in low resolution images. There is a whole body of work on wide baseline matching that deals with quasi-invariant feature-based matching using 2D/3D constraints. In [7], a stable region feature called the Maximally Stable Extremal Regions (MSER) is developed. MSERs are invariant to affine transformation in both image coordinates and intensity. A robust similarity measure is also developed to establish feature correspondences. An improved blob detector is developed in [10], where a robust method is exploited to move across scale space and overlapping regions are allowed. Our regions features adapts this representation.

For object extraction and grouping, Sivic et al. [6] presented a work on grouping object hypotheses in video frames by tracking image patches over long sequences. Affine covariant patches that can be tracked over a large number of frames and move semirigidly over the sequence are grouped into objects. Queries are matched to learned object representations by matching the patch-based multiview feature groupings. The strength of this approach is that multiple parts of an object could be matched from many different frames. However, the representation and matching may not lead to exact matches but is more suited to similarity searches. Our strategy of object extraction within sequence is motivated by this approach, but we use different feature representation and matching metric. We customize our across sequence alignment and flexible matching components to suit the resolution constraints as well as the goal of exact matching.

For object matching and classification, there has been significant development in part-based approach in recent years. In [15], objects are represented as a flexible constellation of parts. Scale invariant features (parts) are first detected and a probabilistic model is used to represent the appearance, scale, occlusion, and shape (configuration between parts) of the object class. The model parameters are learned using an EM framework and images are classified in a Bayesian manner. Training is required in this approach and object coverage from the detected features is not guaranteed. Another part-based approach is by [5]. In their work, "informative" overlapping parts (fragments) are selected on the basis of maximizing the information delivered by the fragments about the class (faces, cars, etc.) they represent. Offline training has to be conducted in this approach. The representations developed in these works are too coarse for the purpose of object instance matching. Other related work includes [3], [16], [17], [18], [19], etc.

Another representative part-based object (especially vehicular object) detection method is developed in [20]. A vocabulary of distinctive object parts is automatically constructed from a set of training images. Images are then represented using parts from this vocabulary and the spatial configuration between parts is also modeled. Based on this representation, a learning algorithm is used to automatically learn to detect instances of the object class in new images.

Another vehicle identification algorithm is proposed by Ferencz et al. In [2], they used a hyperfeature for object instance matching, where both local object appearance and location saliency are encoded. By modeling the distributions of comparison metrics on the salient patches and applying the mutual information-based feature selection, a compact representation of the features with high saliency can be build from a single example and efficient object identification can be achieved.

3 Proposed Approach

3.1 Overall Approach

Fig. 2 illustrates our overall approach and it consists of four major steps. We briefly summarize the four steps next and more details follow in Sections 3.3, 3.4, 3.5, and 3.6.

3.1.1 *Within Sequence Object Mask Generation*

In the standard frame-to-frame tracking process, the pixel-by-pixel ownership for the background and foreground cannot be perfectly assigned due to the inadequate background stabilization and subsequent change detection or imperfect background modeling and subtraction. However, reliable object matching requires the distraction from the background be reduced to the minimum. Therefore, we first need to obtain the precise object ownership mask, given an approximate bounding box for the object. We choose region features for the task since they have good coverage property. Blob-like regions for the key frames are extracted after they are aligned with their neighboring frames within the sequence and the blob configuration and appearance are simultaneously compared using an EMD-based metric. Outliers due to background clutter are also rejected and an accurate object mask is generated in the blob matching process.

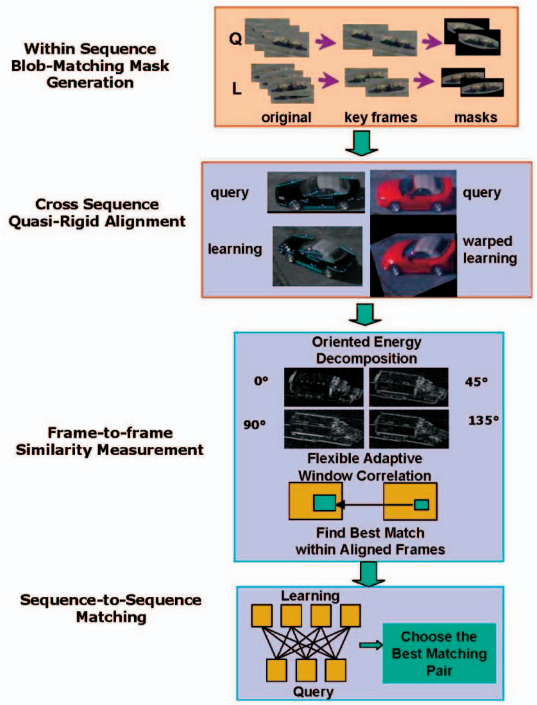

Within Sequence Blob-Matching Mask Generation
Q
L
original
key frames
masks
Cross Sequence Quasi-Rigid Alignment
query
query
learning
warped learning
Frame-to-frame Similarity Measurement
Oriented Energy Decomposition
0°
45°
90°
135°
Flexible Adaptive Window Correlation
Find Best Match within Aligned Frames
Sequence-to-Sequence Matching
Learning
Choose the Best Matching Pair
Query

Fig. 2. Overall image matching framework. It consists of *within* sequence mask generation (cream), *across* sequence image alignment (blue) steps, following the similarity measurement. A sequence-to-sequence matching strategy is used to achieve robustness to occlusion, pose, and lighting changes.

3.1.2 *Across Sequence Image Alignment and Matching*

For across sequence matching between key frames, large pose and appearance change need to be dealt with. Since corner-like and line features have good localization characteristic, they are utilized to align query and learning images to the best possible accuracy.

3.1.3 *Matching Measurement*

A matching score is produced that consists of several terms (normalized color correlation, color similarity, etc.) that are computed within the object mask. More details are given in Section 3.6.

3.1.4 *Sequence-to-Sequence Matching*

Finally, we pose the problem of vehicle matching and fingerprinting with the aerial video context as sequence-to-sequence matching problem. Sequence-to-sequence matching can be robust to occlusion, pose, and lighting changes. One frame can potentially find a best match within a sequence of frames that may not be all affected by the same set of changes. We first choose all the representative *key frames* for both learning and query sequences. We then match each key frame in a query sequence to each key frame in a learning sequence. Key frames were selected based on the drastic change in appearance and motion. Simple appearance and motion model were used for this purpose. The frame-to-frame matching uses aggregated matching of local neighborhoods with flexible templates, as illustrated in Section 3.6. The best matching score is chosen as the final match measure. Choosing the best K pairs with/without temporal constraints is another option. Alternatively, one can build aggregated descriptors for both learning and query sequences and then match the descriptors.

3.2 Prewarping Stage

Since our focus is vehicle matching, we exploit imaging platform related metadata that is typically available from inertial sensors. The metadata includes time, object velocity, platform aspect angle, depression angle, slant range, resolution, sensor azimuth, sensor modality, object bounding box, etc. What is important to us is the extracted relative orientation between the camera and the object (the translation part is not reliable), the driving direction of the object, as well as the approximate resolution of the object. The resolutions and appearances in aerial videos can typically vary over a large range, typically 2-20 cm/pixel. To cope with large-scale difference, platform metadata can be used to preprocess the image data to approximately match resolutions. Image features (points and lines) for learning and query objects can then be derived and matched at similar resolutions. Furthermore, platform orientation and an object's direction of motion can be used to define three directions corresponding to the primary vehicle orientations in the image.

After scale and orientation compensation by metadata, we also use edge-based Chamfer Matching [16] to solve for the initial translation, thus completing the prewarping process.

3.3 Within Sequence Object Mask Generation

We adopt the robust blob features developed in [10] and match blobs in consecutive frames to generate object mask. The consecutive frames normally have moderate amount of motion and can be aligned using simple affine transformation. Within a frame, blobs may overlap with each other and, from frame to frame, they may split or merge because of shadow, lighting change, etc. A reliable metric is therefore needed to handle the multiple to multiple blob matching between consecutive frames. The Earth Mover's Distance is the natural choice for this purpose. The overall schema of mask generation through blob matching is depicted in Fig. 4. We introduce individual components of our region-matching scheme in the following subsections.

3.3.1 Robust Blob Features Extraction

First, we present our homogeneous regions as *blobs*. A similar approach is utilized in [11]; however, the method described there is based on using color segmentation and does not account for image motion. Compared with regions computed from segmentation algorithm, blobs need not follow the exact shape of the objects. They are more robust and invariant to scale, appearance, and view change. Stable regions are obtained by a hierarchical clustering scheme where cluster centers are formed based on pixel appearance and location. These blob features are similar to the MSER features [7], but they do not need to be darker or brighter than all their neighbors. Overlapping and nesting are allowed in the blob representation and features over a wide range of scales are detected. For matching regions, we detect as many as possible meaningful blobs in the initial stage and, in an iterative merging and pruning step, merge small regions belonging to the same spatial *and* temporal homogeneous regions, and remove most outliers regions caused due to background clutter and other moving objects. Some of the initial blob detection results are shown in Fig. 5. Each blob is represented by the average color, area, center, and the inertia matrix.

3.3.2 Earth Mover's Distance

The Earth Mover's Distance (EMD) [12], [21] is a flexible similarity measure between two multidimensional distributions in some feature space, where a distance measure between single features, called *ground distance*, is given. Intuitively, given two distributions, one can be seen as a mass of earth properly spread in space, the other as a collection of holes in that same space. Then, the EMD measures the least amount of work needed to fill the holes with earth. Here, a unit of work corresponds to transporting a unit of earth by a unit of *ground distance*. Computing the EMD is based on a solution to the well-known *transportation problem* [12].

Suppose there are M & N clusters in the first and second set of distributions P & Q, respectively, and each cluster is associated to a weight, w_{pi} (for the $i\mathrm{th}$ cluster in P) or w_{qj} (for the $j\mathrm{th}$ cluster in Q), that represents the fraction of the distribution for the cluster, then EMD is defined as

$$EMD(P,Q) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij} d_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij}},$$

where f_{ij} is the flow between the $i\mathrm{th}$ cluster in P and the $j\mathrm{th}$ cluster in Q and d_{ij} is the ground distance.

The EMD naturally extends the notion of a distance between single elements to that of a distance between sets or distributions. It can be applied to the more general variable-size sets of distributions and allows for partial matches in a very natural way. This is important to deal with occlusions and clutter in image matching.

3.3.3 Region Matching though EMD

If we represent an image by a distribution that consists of a set of clusters (blob-like regions), where each cluster is represented by its feature (color, location, and area) and by the fraction of the distribution that belongs to that cluster, then similarity between the images can be naturally computed with EMD that basically compares the similarity between the two sets of regions. The ground distance in this case is defined as the linear combination of differences in color and location for the corresponding blobs and the fraction (weight) for each blob is defined as the percentage of the area for the blob with respect to the total area for all the blobs in the same image.

The EMD metric defined above is a global match measurement for both object shape and appearance since it accounts for the combined difference in appearance (color) from all the blobs, and it also compares the object part (blob) configuration by incorporation location difference into the ground distance definition.

In addition to serving as a matching measurement, the flow matrix $F = \{f_{ij}\}$ produced by EMD optimization indicates the correspondences between the two sets of blobs. This can be demonstrated in Fig. 6. The EMD flow matrix successfully discovers that region 1 in the left image corresponds to regions 1-4 in the right image. We can utilize this property to iteratively merge small regions in one image based on the blob homogeneity in the other image and vice versa. Eventually, if the two images correspond to the same object, similar sets of blobs should be produced, where corresponding blobs should have similar size, orientation, and location. In addition, outliers corresponding to background clutter and confusers can also be removed from the EMD flow. The final EMD cost indicates the similarity of the two objects. The outcome of the region matching process for this example is shown in Fig. 7. Note the white vehicle moving in the opposite direction in the background is removed from the object mask.

3.3.4 Object Mask Generation

Since both the learning and query are represented by a short sequence in our framework, the region matching technique can be easily adopted to produce the object mask. For each frame in the learning (or query) sequence, first, we align the current frame with respect to its neighboring frame with affine transformation and then apply the region matching technique using EMD as introduced above. The mask is produced by taking the union of the blobs and applying a dilation operation afterwords. The final masks produced for the red van and other objects are shown in Fig. 8.

3.4 Object Matching Strategy

In order to address the challenge of significant pose change, it is no longer feasible to rely on precise alignment between learning and query images and global image template matching. Matching representations vary in the amount of appearance and geometry information they exploit. The richer the object representation, the better the discrimination between confusing similar targets. However, richer geometric representations demand greater alignment accuracy, which is difficult with moderate resolution imagery when there is 3D pose change and partial occlusion. Therefore, the most practical strategy is to use moderately rich representations that don't demand accurate 3D alignment.

Our approach will be able to accommodate significant pose and appearance changes, occlusion, and similar-looking confusers due to four key features:

1. the use of metadata,
2. sequence-to-sequence strategy,
3. quasi-rigid alignment to achieve moderate accuracy 3D alignment, and
4. flexible template matching to compensate for slight misalignment.

We now illustrate item 3 in more detail.

3.5 Line Feature-Based Quasi-Rigid Alignment

To handle large pose change, we adopt a feature-based alignment approach [22]. For many manmade objects such as buildings and vehicles, edges are the most dominant features. Ideally, if we can detect all the edges and reconstruct their 3D locations and orientations, together with the color/texture information for all the regions delineated by the edges, we can fully describe the geometry and appearance of the vehicle. However, given the relatively low resolution in aerial imagery, reliable bottom-up reconstruction and 3D matching is not possible. We exploit piecewise parametric feature matching to create a seed set of reliable feature matches based on edges. These matches are then used to morph between two frames using piecewise linear warping constrained by the feature matches. Therefore, lines and edges become the primitive operating elements in our approach. Of course, points can be easily incorporated in a similar way.

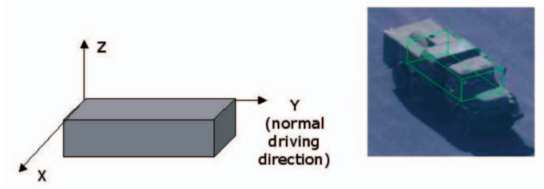


Fig. 3. A box model is superimposed on the image to demonstrate the orientation from metadata. Translation is not determined from metadata.

To obtain invariant representation of lines, we handle rotation by classifying and matching lines in the three principal directions, as shown in Fig. 9. Our line feature descriptor is invariant to moderate translation and scaling.

Reliance on discrete matches only does not use all of the information available in images, which is especially limiting when dealing with low resolution imagery. Our use of image metamorphosis technique [23] to interpolate the correspondence field constrained by the sparse set of features and establishing dense correspondences uses all the available data. This operation approximates a weighted piecewise affine motion model, which can handle both wide and narrow FOV imaging scenarios. When parallax cannot be ignored, no single parametric motion model can align the images well, but piecewise combination of multiple affine models suffices.

3.5.1 Detect and Classify Lines

We begin by performing Canny edge detection. From the metadata or using dominant orientation computation, we can obtain the approximate driving direction shown as the positive Y-axis in Fig. 3. The X-axis is defined to be perpendicular to the driving direction and the Z-axis is perpendicular to the ground. Edges whose orientations are close to the driving direction are classified as Y-edges. Edges that are approximately perpendicular to the driving direction are classified as X-edges. Since most vehicles do not have many edges that are exactly perpendicular to the ground (for example, most sedans have sloped edges), we don't define Z-edges; instead, we allow a large variation in the X-edge orientations. Initially, as many edges as possible should be detected, then short edges belonging to the same class are linked to form longer line segments if they are either close-by or overlapping. The classification results for a red car at two different orientations are shown in Fig. 9.

3.5.2 Line Segment Feature Descriptor

We need an appearance descriptor for each line segment feature to match them between views. The intensity/color transition from one side of a line to the other is a distinctive attribute, so we form a "band image" by collecting [Y R G B] samples around a small band along each edge. Because the image varies slowly parallel to the edge, the band image is insensitive to the instability of the segment endpoints. The sign of the edge orientation needs to be maintained since bright-dark and dark-bright transitions come from distinct

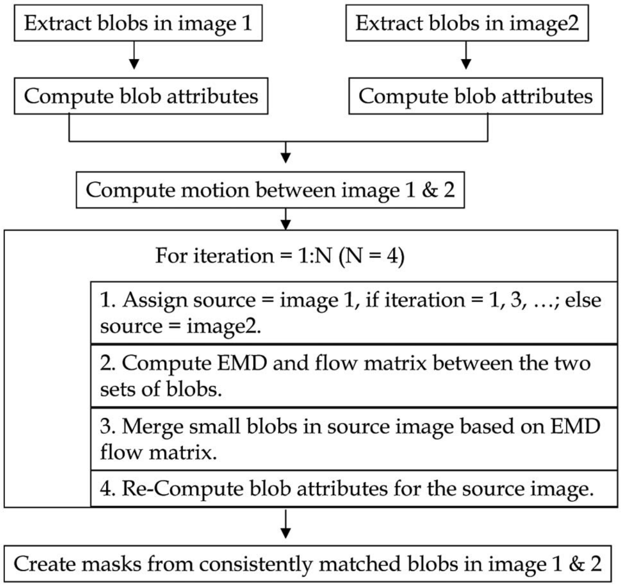

Extract blobs in image 1
Extract blobs in image2
Compute blob attributes
Compute blob attributes
Compute motion between image 1 & 2
For iteration = 1:N (N = 4)
1. Assign source = image 1, if iteration = 1, 3, …; else source = image2.
2. Compute EMD and flow matrix between the two sets of blobs.
3. Merge small blobs in source image based on EMD flow matrix.
4. Re-Compute blob attributes for the source image.
Create masks from consistently matched blobs in image 1 & 2

Fig. 4. Schema of mask generation with blob matching using an EMD flow matrix.

object edges. Some band image examples for the line segments on the red car are shown in Fig. 10.

3.5.3 Establish Line Correspondences

Once we have detected lines, classified them into two groups, and formed the line descriptors, we can use the normalized correlation between the band images to establish the line correspondences between two frames. The correspondences are established for the Y-Edges and X-Edges separately and the result for the red car is shown in Fig. 11. The numbers above each line segments denote the line indices. Note the figures and numbers in the subsequent pictures are shown in smaller size because of space limit.

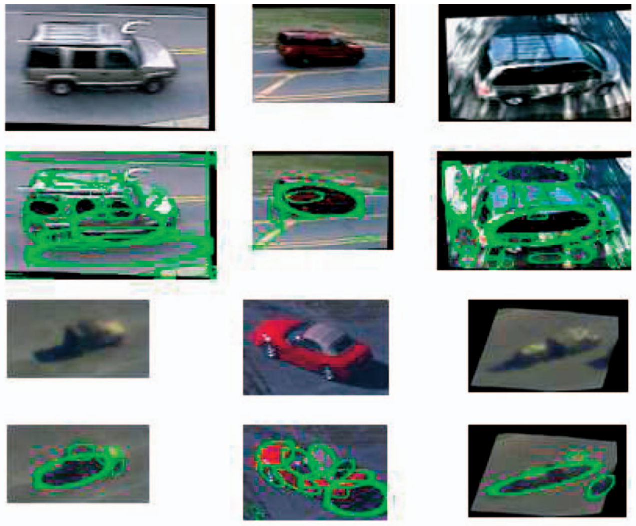


Fig. 5. Initial blob detection results. The first and third rows show the original chips and the second and forth rows are the detected blobs superimposed on the original chips. Note the size variation of the objects and blobs.

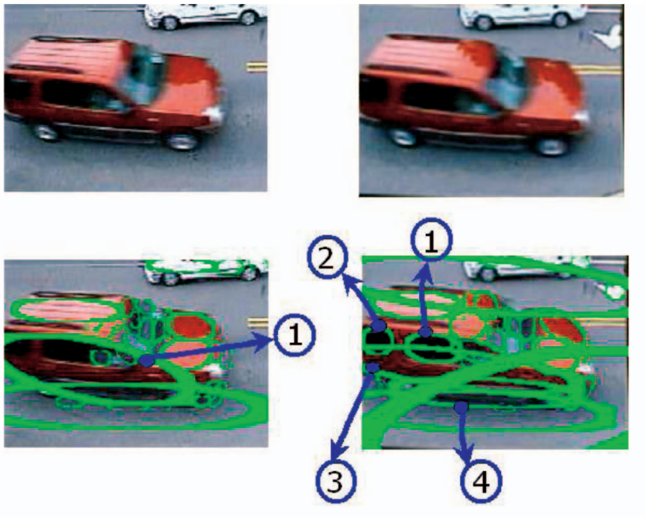


Fig. 6. The first row: Original two chips. The second row: Region 1 in the left image is split into four neighboring regions (1-4) in the right image. The left and right images are consecutive images in a time sequence. This region correspondence can be revealed by examining the EMD flow matrix.

3.5.4 Reject Outliers

Since some edges have similar appearance, the preceding process includes false matchers; moreover, edge matches in a cluttered background are useless for object alignment. To remove these false matches, object shape induced rigidity constraints need to be employed. The rigidity constraints are employed in a progressively restricted way. The initial step uses a simple approximate rigidity constraint to prune raw features and the subsequent step performs further pruning using a more rigorous constraint.

The initial step assumes the target depth variation is small compared to the target range. In each direction, parallel lines (X-edges or Y-edges) on a 3D plane in the scene will project to families of parallel lines in both views, where the distances d_i and d'_i of corresponding lines to the image origin satisfy a linear mapping of the form $d'_i = s \cdot d_i + t$, as shown in Fig. 12. s and t are scale and translation parameters in the 1D affine model (for the distance function) and line matches that fit this model poorly are rejected as outliers.

Second, we can explicitly reconstruct the 3D position and locations of lines from a pair of frames since we know the metadata and line correspondences, even though, in

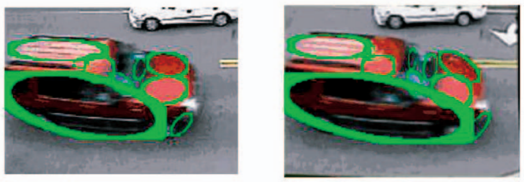


Fig. 7. Iterative region matching results for the same two consecutive images shown in Fig. 5. Note that corresponding blobs have similar size, orientation, and location, and the global blob configuration is almost identical. Outliers such as the white car are rejected. Blobs corresponding to background region have large EMD ground distances and are considered to be outliers and removed since their motion is not consistent with the dominant motion (that corresponds to the vehicle).

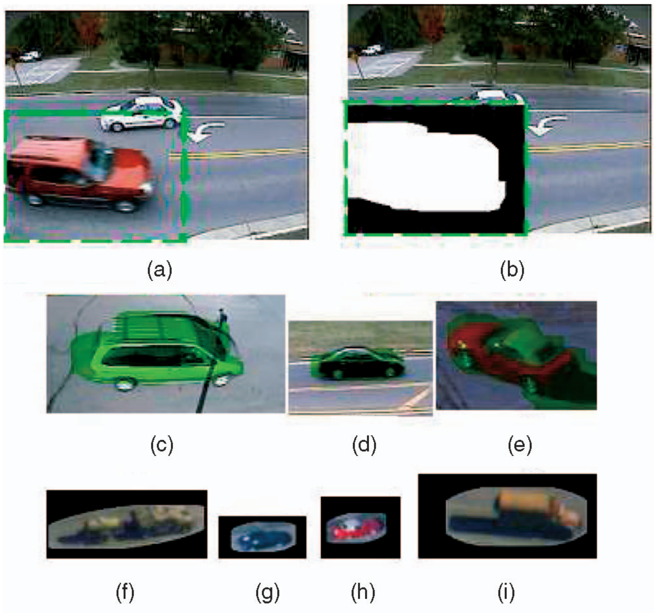


Fig. 8 Mask generation examples. (a) The original image with the object bounding box provided by the tracker. (b) The extracted mask (note it does not include the white car nearby). (c), (d), and (e) Masks are shown in green and superimposed on top of each object. (f), (g), (h), and (i) Ellipse fitting of precise object mask. Ellipses are enlarged by 20 percent, and pixels outside of masks are excluded.

general, three frames are required to reconstruct a 3D line. In 3D reconstruction, camera rotation is given by the metadata and camera translation and 3D line locations can be estimated from 2D line correspondences, as explained in Section 6. Because of the small baseline and other reasons, the 3D reconstruction and, therefore, the computed 3D line positions cannot be perfect. However, for the correctly corresponding line pairs (inliers), the reconstructed line location error is small; for the incorrectly corresponding line pairs (outliers), the 3D location error is very large. Since at this stage, most line correspondences are true correspondences, most reconstructed 3D edges belong to the same object and they tend to group together. Outliers are far away from inliers after reconstruction and will be removed. The line correspondences after outlier rejection are shown in Fig. 13. We can see that edges from the background and shadows in Fig. 11 are removed.

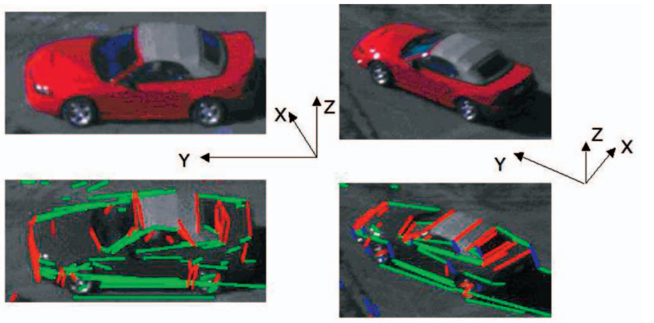


Fig. 9. Detect and classify lines for vehicles. All of the lines are classified either as Y-edges (along the driving direction, in green) or X-edges (perpendicular to the driving direction, in red). Z-edges (in blue) are ignored.

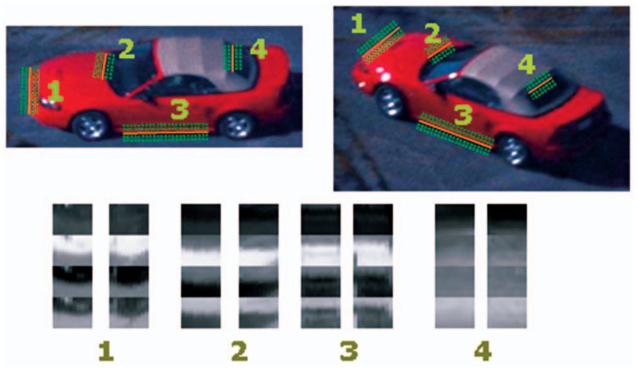


Fig. 10. Line descriptor examples. Band images are formed by collecting [Y R G B] profiles along a small band along each edge, shown in the top row. The descriptors are shown in the bottom row.

3.5.5 Extend Line Segments

As mentioned before, we need to use discrete matches to approximate the dense motion field that accounts for 3D object structure, with regions of different 3D orientation undergo significantly different transformations. As shown in Fig. 14, faces A1, A2, and A3 undergo different transformations. However, for pixels within region A1, their motion can be approximated by an affine transformation, which is defined by at least three lines surrounding region A1. All the local affine coordinate systems can be established by using groups of three line correspondences. However, since we cannot easily form regions without using explicit 3D models, we use the following method to define local similarity transformation from X-edges and Y-edges separately. Two edge correspondences from two distinctive directions define a local affine transformation.

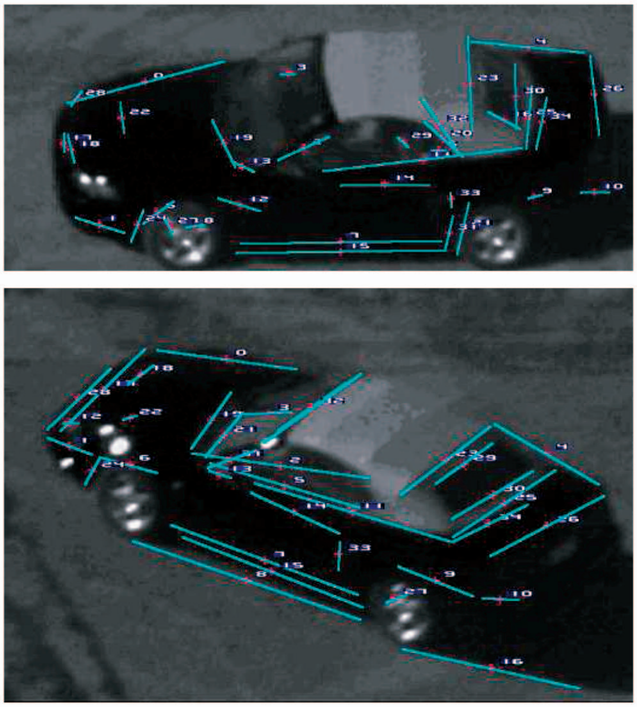


Fig. 11. Establish line correspondences. The numbers above each line denote indices.

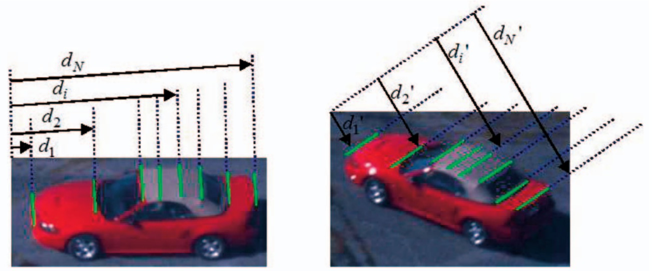


Fig. 12. In each direction, the distances d_is and d_i's for the line correspondences of the same object should satisfy an approximated 1D affine transformation, i.e., $d_i' = s \cdot d_i + t$, where s and t are scale and translation parameters, and they are computed from line correspondences.

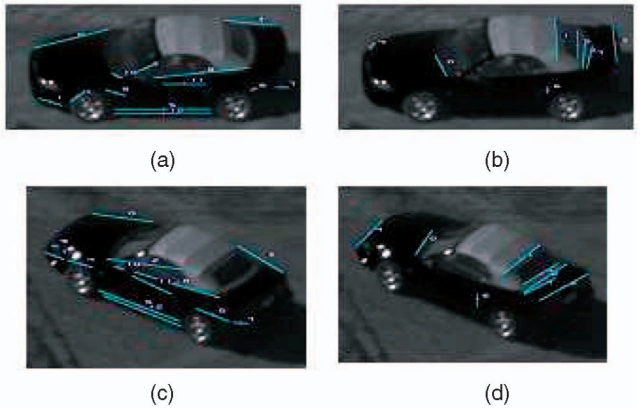

(a) (b) (c) (d)

Fig. 13. Outlier rejection results. Outliers are rejected for (a) horizontal and (b) vertical lines separately. Note that, compared with the originally established line correspondences shown in Fig. 11, false matches such as lines from the shadow are rejected.

To achieve a similarity transformation from a line correspondence, we need to compute the stable end points of lines. We modify the direction of each line to point toward either toward or perpendicular to the dominant orientation (driving direction), depending on whether it is in the Y-edge or X-edge group. Then, each line is intersected with the closest two lines in the other group, giving its refined endpoints. Note that the requirement for the orientation accuracy is not stringent in this step; what is important is that all the edges in the same group should have uniform orientations. The line extension results are shown in Fig. 15.

3.5.6 Interpolate Flow Fields

After establishing the line correspondences and their end point correspondences, for each line segment, we define one image flow field that is a similarity transformation that aligns those endpoints. As shown in Fig. 16, for two corresponding lines PQ and $P'Q'$ in the destination (Fig. 16a) and source image (Fig. 16b), for each pixel X in the destination image, we first find the corresponding (u, v), where u is the distance from X to P along the PQ direction and v is the distance from X to PQ. The pair (u, v) is then used to find the pixel X in the source image, i.e., destination Image(X) = sourceImage(X), with (u, v) is define similarly in the source image. Through this operation, each pixel coordinate is transformed by similarity. Pixels along each line in the source image are transformed to the corresponding line in the destination image. The whole image is transformed.

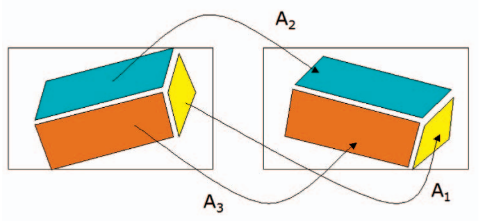


Fig. 14. For a 3D object, regions with different 3D orientation undergo different transformations.

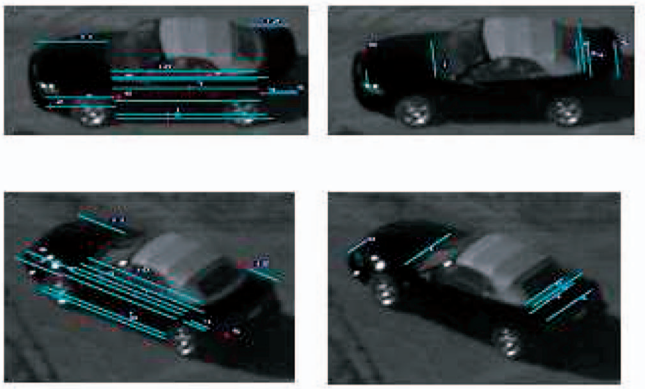
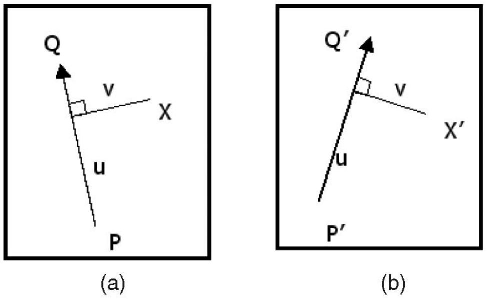


Fig. 15. Extend line segments. X-edges and Y-edges are extended to intersect with their closest Y-edges and X-edges. The orientations are modified to be the same for all of the X-edges and Y-edges, respectively.

Given multiple line segment correspondences, we form a weighted average of the flow fields. The weights decay away from each line segment, ensuring that a segment only affects the flow in its vicinity. This is exactly the approach proposed by [23]. The flow field obtained is smooth and coincides with the line segment correspondences. The use of the interpolated correspondence field also overcomes the coverage problem using sparse features and utilizes the appearance/texture information in between the features. Finally, we use the interpolated flow field to align images query and learning sequences, and we dub this alignment method as *"quasi-rigid"* alignment.

The quasi-rigid alignment scales well with respect to image resolution. When image resolution is low (in the case of Wide FOV data), we can only get a few pairs of line correspondences, but a simple parametric motion model such as affine is sufficient in this case. Theoretically, we



(a) (b)

Fig. 16. Illustration of the metamorphosis operation.

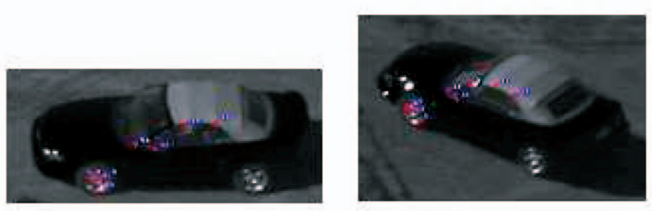


Fig. 17. Point feature correspondences. Note that the point feature coverage is poor and, therefore, the computed global motion model is not reliable.

only need two pairs of line correspondences to align the low-resolution (Wide FOV) images. When the resolution becomes higher (in the case of Narrow FOV data), a simple parametric motion model is not sufficient and parallax cannot be ignored any more. Fortunately, we can extract and establish much more corresponding line pairs because of the high resolution and the motion field established by interpolating the correspondences from these line pairs can well approximate the true motion field.

3.5.7 Incorporation of Points

Points can also be utilized in our framework. An example of surviving point correspondences after outlier rejection is shown in Fig. 17. Points are extracted using the Harris Corner detector [24] and matched using normalized correlation. In this example, only points on the near side of the car have good correspondences, very few points correspond well on the far side, the coverage is poor, and no single global motion model estimated from these points can explain the whole object well. However, points can be used together with lines to constrain the flow field and the final alignment result is shown in Fig. 18. More alignment results are shown in Fig. 19.

3.6 Flexible Local Template Matching

In object matching, we need to account for approximations in alignment as well as appearance differences due to a variety of unmodeled changes. We propose matching a patch to a local distribution of patches within the constraints provided by aligned images. Specifically, we represent patches using oriented energy filter outputs [25]. These capture the significant features in a patch while ignoring certain illumination effects. Each patch captures the spatial arrangement of edge energy and orientations within the patch. In order to account for local alignment differences, we perform nearest-neighbor matching of the patch to a collection of patches in the target image. Specifically, the score of the patch is computed as that of the best matching patch within a small range of translations around the patch to be matched. Scores from all the local patches are aggregated to compute a single score between a query and a learning image. The aggregated score is a weighted sum of the correlation score from all the pixels. Only the patches containing energy above a dynamic threshold are retained in the aggregation to avoid irrelevant background. The energy is summed over color channels in order to include edges that only appear in color but not in luminance. Fig. 20 shows a pair of aligned images. The oriented energy images in four directions (0°, 45°, 90°, and 135°) are shown in the center for the left image and the comparison of the flexible versus rigid template matching results are shown on the bottom. The brighter the pixel, the better the matching.

The size of the local patches is an important consideration, for if a patch is much smaller than the nominal scale of features in a neighborhood, the image pattern in the patch will match at many shifts and with many objects, but if a patch is too large, it becomes too sensitive to appearance changes and misalignments and may not be sensitive enough to small but discriminative patterns. We adapt the patch size by choosing, for each location in an image, the size of a window centered there whose gray-level distribution has maximum entropy [9].

The local template matching score fails to discriminate objects that have the same geometric structure and differ only in color. Therefore, we augment the flexible matching score with two additional terms to give an overall similarity

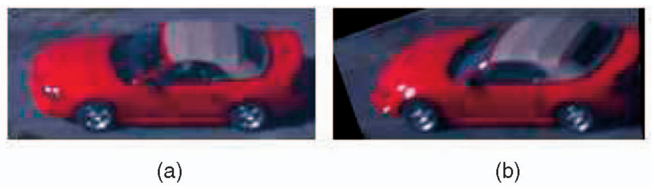


Fig. 18. Quasi-rigid alignment results for the pair of images in Fig. 10 ((a) original model image, (b) warped query image). The pose change is around 33°.

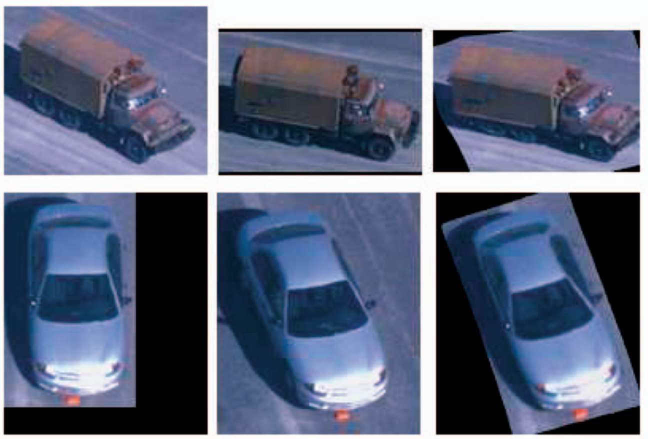


Fig. 19. More quasi-rigid alignment results. In each row, the first and center images are the original query and model images and the right one is the warped query image. Note the large aspect change in these examples.

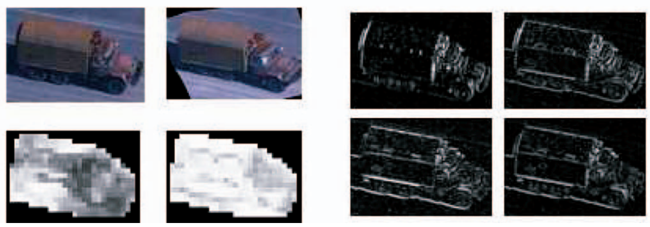


Fig. 20. Flexible local template matching. Top Left: Learning and query frames after quasi-rigid alignment. Right: Oriented energy in four directions. Bottom Left: Local correlation scores with global affine versus quasi-rigid alignment with 11 pixel search range.

metric. One term is local template matching of RGB images and another term is average color similarity, measured by the angle between RGB vectors. The template matching for the RGB images is a modified version of normalized correlation. The modification deals with the texture less region better. For the similarity score combination from three sources, we first generate each score separately for all the trials and we obtain the weights by searching to obtain the optimal matching performance.

Although the query and model images are rescaled to a common resolution, the absolute GSD can vary from one instance to another. Certain parameters of the approximate 3D alignment process and flexible local template matching are varied according to the GSD.

4 Performance Evaluation

4.1 Performance Evaluation Methodology

We have extensively evaluated our algorithm for 109 vehicles for Electro-Optic (EO) data and 88 vehicles for Infrared (IR) data. The database has a wide variety of vehicle models with different colors, shapes, and sizes. Some representative vehicles are shown in Fig. 1. Our experimental setup is designed to test the following aspects of the algorithm:

1. comparison with the traditional feature-based global affine alignment + rigid template matching method,
2. performance for both the wide and narrow FOV videos,
3. temporal gap test,
4. performance on the aspect angle and Ground Sampling Distance (GSD),
5. performance on the aspect angle difference and GSD difference, and
6. performance on the difference match measurements.

For each set of experiments, we conduct a large number of trial tests. Each trial contains one query and N = 5 learning sequences, where the targets in the learning sequences are all distinct and one of the learning sequences contains the same object (but obviously 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 the probability of correct association, P_{CA}, is defined as the number of correct outcomes divided by the number of trials.

For the experiments that involve the overall quality such as temporal gap, aspect angle, GSD, aspect angle difference, and GSD difference, all of the vehicles were used. For the comparison of the low resolution data (wide FOV) versus high-resolution data (narrow FOV), wide FOV and narrow FOV data was used separately. For the comparison of quasi-rigid alignment + flexible template matching versus global affine alignment + rigid template matching, we use the subset of narrow FOV data since parallax is more prominent. For the comparison of using precise masks versus not using precise masks, we chose data that exhibit more background clutter for better comparison.

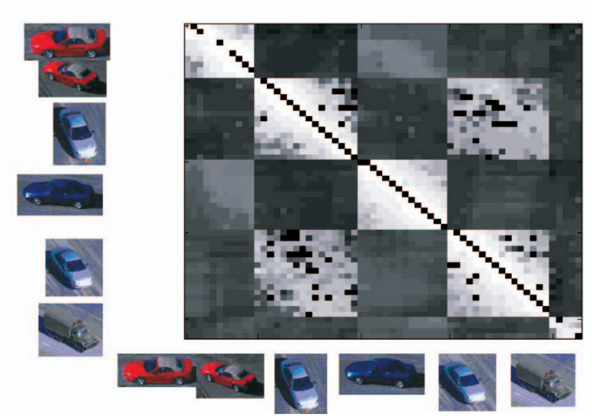


Fig. 21. The similarity matrix for quasi-rigid alignment + flexible matching algorithm on five narrow FOV video objects. $P_{CA} = 91\%$ for pose change $\leq 33^{\circ}$.

4.2 Comparison of Feature Based Global Affine and Quasi-Rigid Algorithms

We first compare the quasi-rigid alignment and flexible template matching (QM) algorithm with a traditional feature-based global 2D affine alignment and rigid template matching algorithm (GM). Fig. 21 shows the similarity matrix for QM on a set of learning and query sequences drawn from a narrow FOV data of four civilian vehicles and one military vehicle. Each row of the matrix is a query and each column of the matrix is a learning sequence. The five distinct bands of rows and columns correspond to the five different vehicles, illustrated by the sample image chips. Brighter matrix elements indicate higher likelihood scores and completely black elements indicate (query and learning) pairs that were excluded. Pairs are excluded if their pose change is greater than 33 degrees. 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 this experiment, there are moderately bright off-diagonal blocks between targets #2 and #4, which are Chevy Cavaliers of the same color but different number of doors. Notice that, within each main diagonal block, the score is highest near the central diagonal and slightly decreases away from the center, i.e., the score decreases slightly as the temporal gap increases, indicating resilience of our algorithm to pose change. In summary, the correct association performance for this data set with QM is $P_{CA} = 91\%$.

Fig. 22 shows the similarity matrix for GM for the same set of sequences as above. Notice that, within each main diagonal block, the score is highest near the central diagonal and quickly decreases away from the center, i.e., the score decreases quickly as the temporal gap increases, indicating poor resilience of GM to pose change. The performance with GM is $P_{CA} = 80\%$. The overall P_{CA} from GM to QM has increased 11 percent, out of which around 7-8 percent increment is due to alignment improvement and 3-4 percent is due to matching measurement improvement.

An 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, Psame versus Pdiff. Ideally, the distributions should be well

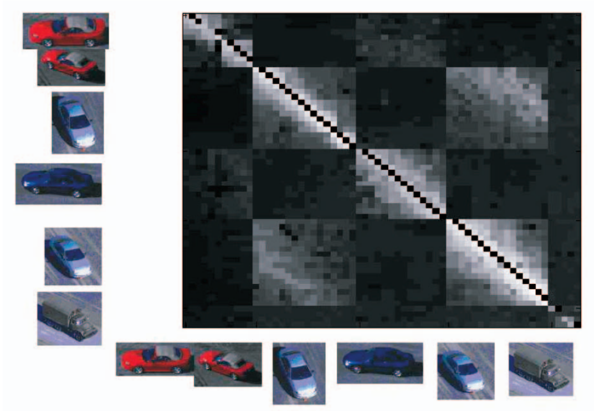


Fig. 22. The similarity matrix for global 2D affine alignment + global correlation algorithm on five narrow FOV video objects. $P_{CA} = 80\%$ for pose change $< 33^\circ$.

separated, in order to reliably discriminate between the correct and incorrect matches. Fig. 23 shows that the separation between the same and different object distributions is weak for GM and significantly better for QM.

Finally, Fig. 24 characterizes the sensitivity of the vehicle matching algorithm performance with respect to the degree of orientation change between learning and query sequences. Each data point in this plot is derived by restricting the set of trials to the indicated amount of pose change; thus, the performance plotted at 17 degrees includes trials with pose change from 0 degrees to 17 degrees, not just trials that are exactly 17 degrees. P_{CA} drops quickly for GM, reaching 95 percent at only 12 degrees, while QM's performance drops less rapidly, reaching 95 percent at 17 degrees and maintaining half the error rate of traditional algorithm. QM performs better than 90 percent even up to 35 degrees pose change, which is especially significant given the low resolution data.

4.3 Comparison of Sensors

We compared the performance of our algorithm using narrow FOV and wide FOV videos. Generally, the resolution in narrow FOV data is higher and the size of vehicles ranges from 50 to 120 pixels. Wide FOV data usually has lower resolution and the size of vehicles ranges from 10 to 50 pixels. One strength of our approach is that it scales well with respect to resolution. As stated in Section 3.4, for the low-resolution images, the line-based quasi-rigid alignment degrades gracefully and sufficiently to parametric 2D motion model (such as affine) with only two pairs of line correspondences. For the high-resolution images, more line correspondences can be established and more complicated motion field can be modeled.

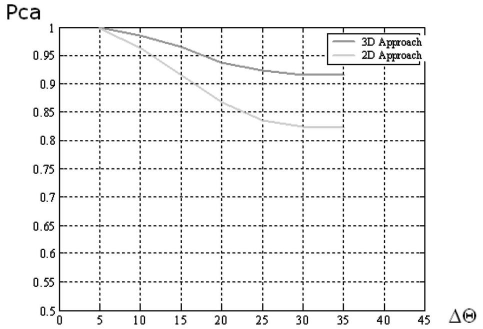


Fig. 24. Sensitivity with respect to pose change for the global affine + rigid template matching versus the Quasi-Rigid + flexible template matching approaches on five narrow FOV targets.

Fig. 25 shows the similarity matrix for our algorithm on a set of learning and query sequences drawn from the narrow FOV data containing four civilian vehicles and three military vehicles. This is a more extensive test than the one shown in the previous section. The orientation change is limited to 30 degrees and resolution change is limited to 30 percent. The correct association performance for this experiment $P_{CA} = 94\%$.

Fig. 26 shows the similarity matrix for our algorithm on a set of learning and query sequences drawn from the wide FOV data containing six civilian vehicles and five military vehicles. The orientation change is limited to 30 degrees and resolution change is limited to 30 percent. The correct association performance for this experiment is $P_{CA} = 96\%$ (for $\Delta\Theta < 15^\circ$) and $P_{CA} = 86\%$ (for $\Delta\Theta < 30\%$), respectively.

4.4 Temporal Gap Performance

We systematically evaluate our algorithm on around a quarter million trials drawn from a data set of around 100 vehicles. We mostly used around 100 vehicles in the data set where the query and learning sequences for the same object separated in time up to 30 seconds. The overall performance for EO and IR sensors are shown in Fig. 27. We have achieved 98.8 percent P_{CA} for EO video data and 95.0 percent P_{CA} for IR video data.

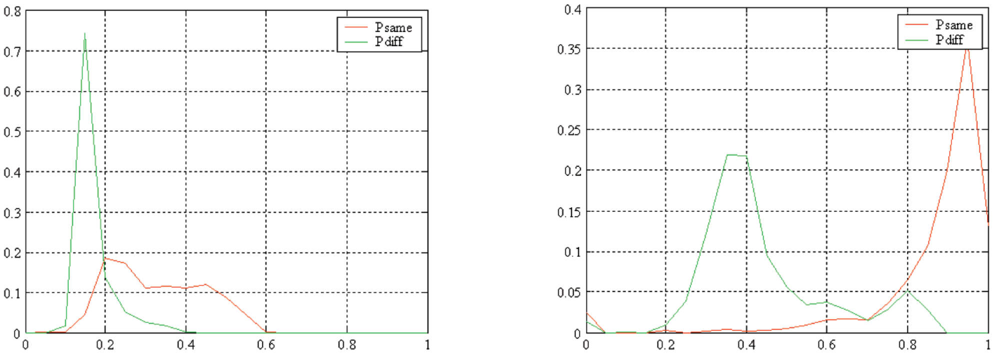


Fig. 23. Distribution of same-object and different-object similarity scores for global affine + rigid template matching versus Quasi-Rigid + flexible template matching approach on five narrow FOV targets. Pose change $< 15^\circ$ in both cases.

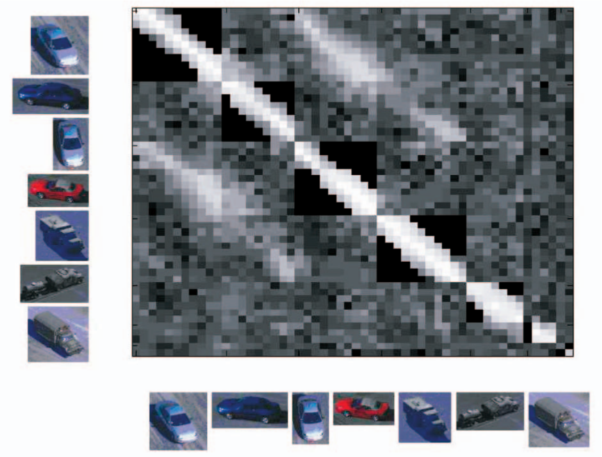


Fig. 25. Similarity matrix for our algorithm on 12 narrow FOV targets. $P_{CA} = 94\%$ for pose change $< 30°$.

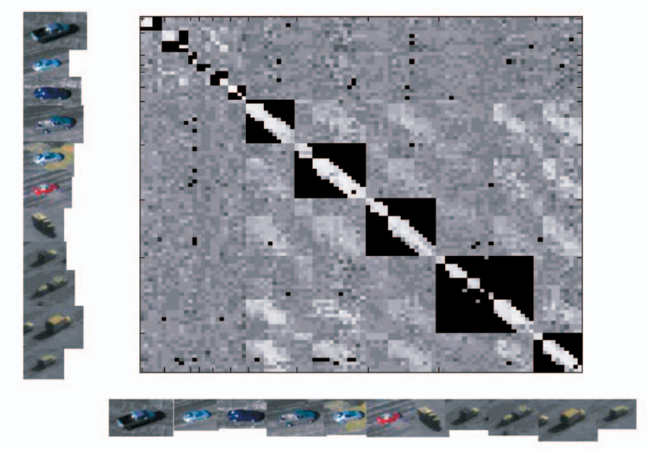


Fig. 26. Similarity matrix for our algorithm on 11 wide FOV targets. $P_{CA} = 96\%$ for pose change $< 15°$, $P_{CA} = 86\%$ for pose change $< 30°$.

4.5 Performance on Pose and Image Resolution: Aspect Angle,GSD, Aspect Angle Difference, and GSD Difference

Our extensive evaluation on the algorithm robustness to pose change is demonstrated through the percentage of correct association with respect to aspect angle and aspect angle difference between learning and query sequences. The performance on image resolution is explicitly demonstrated in the Ground Sampling Distance (GSD) and GSD difference between learning and query sequences since GSD reflects the scale ratio between size of the actual object and its image. The unit of GSD is usually meter/pixel. Again, we can see that our algorithm scale well with different resolution video data, as discussed in Section 4.3.

The overall performance on aspect angle and GSD is shown in Fig. 28 and Fig. 29 and the performance on aspect angle difference and GSD difference is shown in Fig. 30 and Fig. 31.

4.6 Mask Generation Results

To demonstrate the importance of the accurate mask generation, we used the eight vehicles (shown in Fig. 1). The GSD for the set ranges from 0.04-0.08 m/pixel. The maximum pose change is 23°. There are two pairs of similar vehicles, i.e., the beige vans and the dark red vans. Heavy shadows are also presented. Fig. 32 shows the similarity matrix for the data set using region matching-based mask generation plus quasi-rigid alignment and flexible template matching. Each row of the matrix is a query and each column of the matrix is a learning sequence. The bright bands of rows and columns correspond to the eight different vehicles, illustrated by the sample image chips. Brighter matrix elements indicate higher likelihood scores. Notice that, within each main diagonal block, the score is highest near the central diagonal. In summary, the correct association performance for this data set is = 84.3 percent. By comparison, if we do not use the object mask algorithms described in Section 3.3, this drops to 80.1 percent for the one query versus five learning association problem. Note that we get similar improvement if we use perfect hand generated masks.

A more extensive test is conducted on a data set of 553 trials, which contains around 50 different vehicles with up to 120° pose change, and object GSD ranges from 0.06-0.15 m/pixel. Fig. 33 demonstrates the superiority of the QM algorithm with mask generation over a large range of pose change.

4.7 Comparison of Match Measurement

To demonstrate that it is imperative to exploit as much information as possible to achieve good matching for a broad range of pose change between learning and query sequences, we compare the performance of different matching measurements with respect pose change after performing exactly the same alignment, as shown in Fig. 34. The following seven measurements are used in our experiments and they encode different aspects of the object:

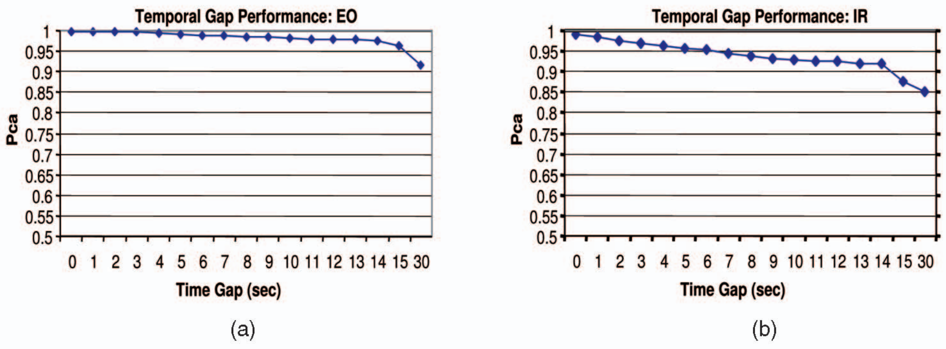


Fig. 27. Temporal gap performance for (a) EO (upper) and (b) IR (lower) data.

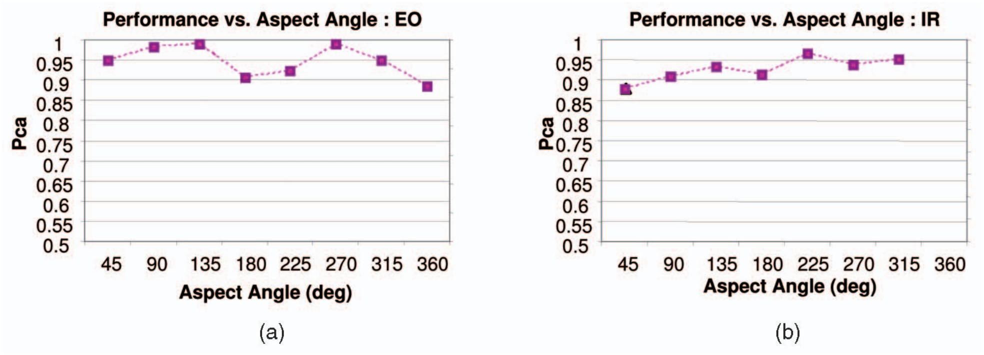


Fig. 28. Performance on aspect angle for both (a) EO and (b) IR data.

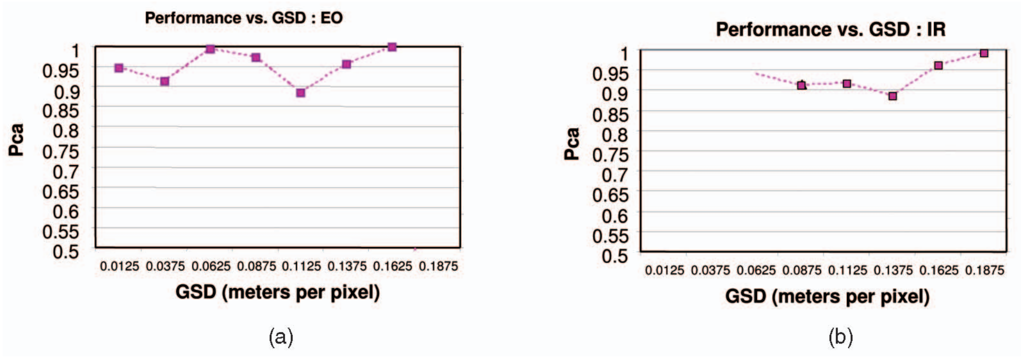


Fig. 29. Performance on GSD for both (a) EO and (b) IR data.

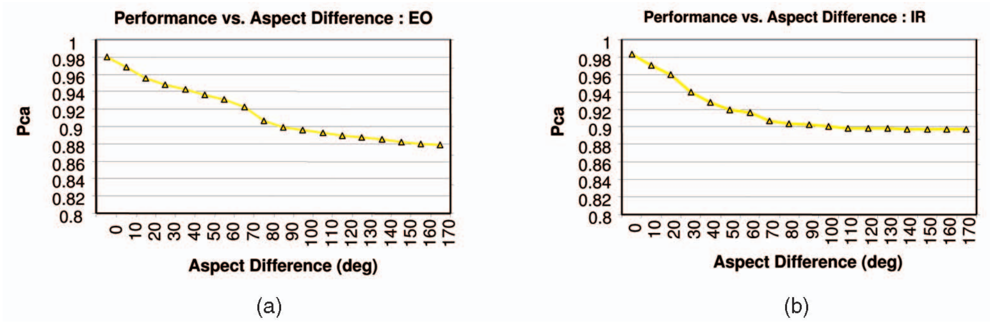


Fig. 30. Performance on aspect angle variation for both (a) EO and (b) IR data.

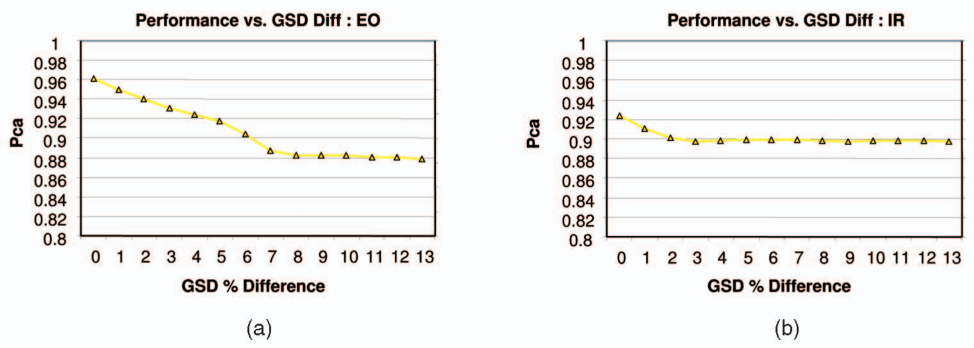


Fig. 31. Performance on GSD variation for both (a) EO and (b) IR data.

1. **Color Correlogram**. It compares global color and some extent of object topology between images. See [26] for detail. We can see that color feature alone in general is quite stable over a large range of pose change, but the overall performance is limited. The upper bound is around 80 percent.
2. **Chamfer Distance**. Chamfer distance incorporates edge location, orientation, and gradient magnitude difference into the match measurement and it facilitates fast object matching using edge maps. It is a global 2D object shape matching measurement and robust to occlusion, obscuration, and noise. It produces a quite good result when the pose change

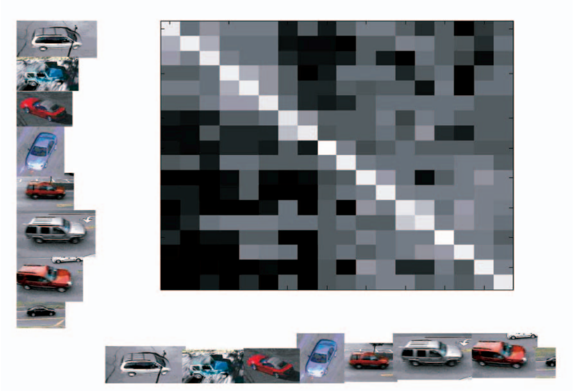


Fig. 32. Similarity matrix for quasi-rigid alignment + flexible matching algorithm on eight video objects. Pca = 84.3 percent for pose change $< 23^\circ$.

is small, but drops rapidly as pose difference becomes larger.

3. **Normalized Correlation**. In general, it performs well since it utilizes both shape and appearance information, but, since it lacks global information such as global appearance, therefore it cannot deal with some easy cases when objects differ largely in color.
4. **Comprehensive Match Measurement**. We combine both appearance and geometry measurements at both the local and global levels and properly choose the combination weight; the combined measurement performance is shown as the red curve in Fig. 34. We can see that it achieves the best result since all information is used.

4.8 Representative Trials and Matching Results

For better visualization purposes, we created Web pages to view the matching results for all the trials. Please see the attached HTML files for more detail. Some representative trials and matching results are shown in Fig. 35. There is a large variation in object size, aspect, and appearance and our algorithm is able to establish the correct association between query and learning sequences.

5 Conclusion

In this paper, we have demonstrated the efficacy of Confirmatory Identification (CID) as a means for reliable tracking of vehicles in aerial videos under real-world operational constraints. We designed, implemented, and thoroughly tested novel CID algorithms and software. The software was tested and evaluated using hundreds of thousands of trials consisting of learning sequences and a query sequence that simulated breaks in frame-to-frame tracking that is typical in aerial video tracking scenarios. Under operating conditions consistent with 10 secs. of temporal gap in tracking, the CID module has achieved 98 percent P_{CA} for EO data and 95 percent P_{CA} for IR data. The operating conditions consisted of realistic depression angles, sun angles, aspect changes, pose changes, GSD, and GSD changes, as well as video quality and platform effects.

Our approach consists of a heterogeneous matching algorithm to compare vehicle image sequences taken across time and space. The heterogeneous algorithm uses regions, lines, and points in order to align and compare the images. It is able to compute accurate masks in the presence of occlusion and clutter. Each of the descriptors (regions, lines, and points) used is invariant to different changes in the scene and viewing geometry. We believe this is the first system which combines all three into an overall system. We

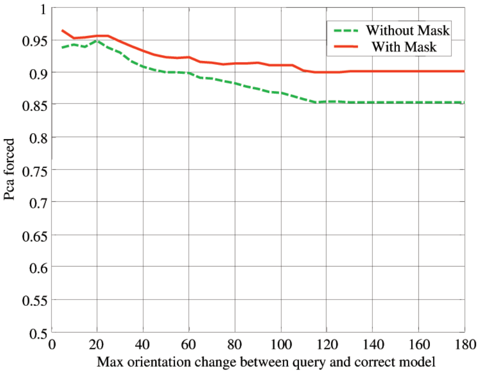


Fig. 33. Performance comparison of the QM algorithm with (red solid curve) and without (green dashed curve) masks.

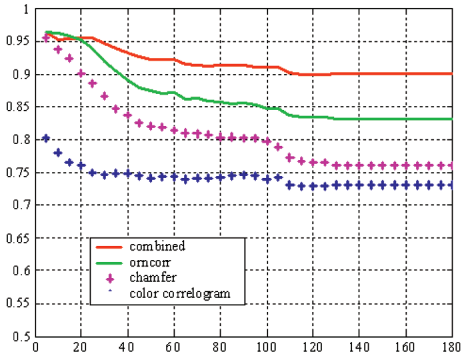


Fig. 34. Comparison of different matching measurements.

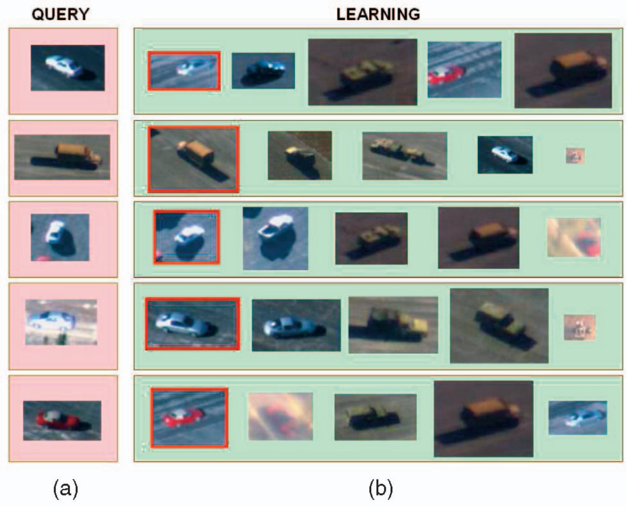


Fig. 35. Representative examples for vehicle matching. Each row is a trial that consists of (a) one query and (b) five learning (right) sequences. The learning chips are arranged (from left to right) according to their matching scores with regard to the query. The correct matches are marked with red boxes.

achieved correct ID rates in the range of 90 percent and above for large pose changes for a wide variety of sensors and resolutions.

Appendix

Reconstruct 3D Lines from 2D Line Correspondences with Affine Camera Model

A line $\tilde{\mathbf{l}}$ in an image is projected by the projection matrix $\mathbf{P}$ to a plane $\tilde{\pi}$ in 3D as: $\tilde{\pi} = \mathbf{P}^T\tilde{\mathbf{l}}$. For two affine cameras,

$$\mathbf{P}_1 = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{P}_2 = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & \gamma_{14} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & \gamma_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

the reconstructed planes are:

$$\pi_1 = \begin{bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{bmatrix} = \begin{bmatrix} \omega_{11}l_{x1} + \omega_{21}l_{y1} \\ \omega_{12}l_{x1} + \omega_{22}l_{y1} \\ \omega_{13}l_{x1} + \omega_{23}l_{y1} \\ \omega_{14}l_{x1} + \omega_{24}l_{y1} + l_{z1} \end{bmatrix}$$

and

$$\pi_2 = \begin{bmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{bmatrix} = \begin{bmatrix} \gamma_{11}l_{x2} + \gamma_{21}l_{y2} \\ \gamma_{12}l_{x2} + \gamma_{22}l_{y2} \\ \gamma_{13}l_{x2} + \gamma_{23}l_{y2} \\ \gamma_{14}l2 + \gamma_{24}l_{y2} + l_{z2} \end{bmatrix}.$$

The intersection of the two planes π_1 and π_2 produces a line with its general equation:

$$\begin{cases} A_1X + B_1Y + C_1Z + D_1 = 0 \\ A_2X + B_2Y + C_2Z + D_2 = 0. \end{cases}$$

If we choose the world coordinate system as shown in Fig. 3, for the constant z planes, $x = 0$, we have

$$\begin{cases} B_1Y + C_1Z + D_1 = 0 \\ B_2Y + C_2Z + D_2 = 0 \end{cases}$$

and the constant z can be solved as

$$z = (B_2D_1 - B_1D_2)/(B_1C_2 - B_2C_1).$$

Or, for any $x = x_\circ$, define

$$\begin{cases} DD_1 = D_1 + A_1X_0 \\ DD_2 = D_2 + A_2X_0, \end{cases}$$

we have $z = (B_2DD_1 - B_1DD_2)/(B_1C_2 - B_2C_1)$. Similarly, we can recover Y = constant planes.

Acknowledgments

The authors would like to thank Cen Rao for the color correlogram implementation and various testing effort. This work was performed under a US Defense Advanced Research Projects Agency/Department of the Interior Contract No. NBCHC030039.

References

[1] A.K. Jain, Y. Zhong, and S. Lakshmanan, "Object Matching Using Deformable Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 3, pp. 267-278 Mar. 1996.

[2] A. Ferencz, E. Learned-Miller, and J. Malik, "Learning Hyper-Features for Visual Identification," *Neural Information Processing Systems,* 2004.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[4] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. IEEE Conf. Computer Vision,* pp. 1150-1157, 1999.

[5] S. Ullman, E. Sali, and M. Vidal-Naquet, "A Fragment-Based Approach to Object Representation and Classification," *Proc. Fourth Int'l Workshop Visual Form,* 2001.

[6] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Object Level Grouping for Video Shots," *Proc. Eighth European Conf. Computer Vision,* 2004.

[7] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proc. British Machine Vision Conf.,* vol. 1, pp. 384-393, 2002.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-Local Affine Parts for Object Recognition," *Proc. British Machine Vision Conf.,* vol. 2, pp. 959-968, 2004.

[9] T. Kadir and M. Brady, "Scale, Saliency and Image Description," *Int'l J. Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.

[10] P.-E. Forssen and G. Granlund, "Robust Multi-Scale Extraction of Blob Features," *Proc. Scandinavian Conf. Image Analysis (SCIA),* 2003.

[11] H. Greenspan, G. Dvir, and Y. Rubner, "Region Correspondence for Image Matching via EMD Flow," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* 2000.

[12] F.L. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," *J. Math. Physics,* vol. 20, pp. 224-230, 1941.

[13] D. Koller, K. Daniilidis, and H.-H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int'l J. Computer Vision,* vol. 10, no. 3, pp. 257-281, 1993.

[14] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[15] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[16] D.M. Gavrila and V. Philomin, "Real-Time Object Detection for Smart Vehicles," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 87-93, 1999.

[17] U. Grenander, Y. Chow, and D.M. Keenan, *Hands: A Pattern Theoretic Study of Biological Shapes.* Springer-Verlag, 1991.

[18] D. Huttenlocher, D. Klanderman, and A. Rucklige, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.

[19] M. Oren et al., "Pedestrian Detection Using Wavelet Templates," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 193-199, 1997.

[20] S. Agarwal, A. Awan, and D. Roth, "Learning to Detect Objects in Images via a Sparse, Part-Based Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 11, pp. 1475-1490, Nov. 2004.

[21] Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 59-66 1998.

[22] Y. Guo, S. Hsu, Y. Shan, H. Sawhney, and R. Kumar, "Vehicle Fingerprinting for Reacquisition and Tracking in Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[23] T. Beier and S. Neely, "Feature-Based Image Metamorphosis," *Computer Graphics,* vol. 26, no. 2, 1992.

[24] C.J. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Confs.,* pp. 147-151, 1988.

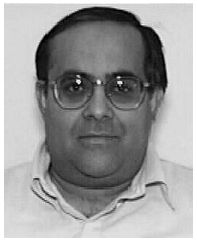
[25] W. Freeman and E. Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 891-906, Sept. 1991.

[26] J. Huang et al., "Image Indexing Using Color Correlograms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 762-768, 1997.

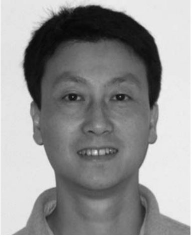
Yanlin Guo received the BS and MS degrees in electrical engineering in 1993 and 1995 from Tsinghua University, Beijing, People's Republic of China, and the PhD degree in electrical and computer engineering in 1998 from the University of Florida. She is a senior member of the technical staff in the Vision and Visualization Group at Sarnoff Corporation. She has extensive experience in computer vision, video and image processing, pattern recognition, and medical image analysis. At Sarnoff Corporation, she has been a key member of algorithm development and technical lead in many commercial and government research and development projects in the areas of video/image registration, image and video enhancement, object detection, object recognition, 3D motion and structure estimation, 3D modeling, and medical image processing. She has published more than 20 papers, holds two US patents, and has several others pending. She has served as a reviewer for top journals and conferences. She was a program committee member of IEEE ICCV 2003, ICIP 2004-2006, ECCV 2006, ICPR 2006, and ACCV 2006. She is a member of the IEEE.

Steve Hsu received the BS degree from Caltech in 1982 and the PhD degree from Massachusetts Institute of Technology in 1988, both in electrical engineering. He is currently with Canesta, Sunnyvale, California, where he is developing range imaging software and automotive computer vision applications for Canesta's focal plane array LIDAR system. From 1988 to 2005, he was with Sarnoff Corporation, Princeton, New Jersey, where he became technical manager, Mobile Vision Systems Group. He led Sarnoff's technology thrust in 3D LIDAR data registration and exploitation. His research in digital image, video, and range data processing spans aerial video and geospatial image processing, image registration and mosaicking and pose estimation, 3D scene reconstruction and modeling, object recognition, biometrics, and video compression. Dr. Hsu pioneered the Topology Inference Local to Global Alignment framework for robust automatic mosaicking of large image sets, enabling the VideoBrush™ consumer product as well as gigapixel-sized mosaics for geospatial applications. He is a member of the IEEE and is a contributor to 15 patents and more than 30 publications.

Harpreet S. Sawhney received the PhD degree in computer science in 1992 from the University of Massachusetts, Amherst, focusing on computer vision. His areas of interest are object recognition, motion video analysis, 3D modeling, vision and graphics synthesis, video enhancement, video indexing, data mining, and compact video representations. He is the technical director of the Vision and Learning Technologies Lab at the Sarnoff Corporation. Since 1995, he has led government and commercial programs in immersive telepresence, image-based 3D modeling, video object fingerprinting, video mosaicing, geo-registration, 2D and 3D video manipulation, and object recognition. Dr. Sawhney was one of the key technical contributors toward the founding of two Sarnoff spinoffs, VideoBrush Inc. and Lifeclips Inc. Between 1997 and 2004, he was awarded the Sarnoff Technical Achievement Award seven times for his contributions in video mosaicing, video enhancement, 3D vision, and immersive telepresence. Between 1992 and 1995, he led video annotation and indexing research at the IBM Almaden Research Center in San Jose, California. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. He has also served on the program committees of numerous computer vision and pattern recognition conferences. He has published more than 60 papers and holds 15 patents. He is a member of the IEEE.



Rakesh Kumar received the BTech degree in electrical engineering from IIT-Kanpur, India, the MS degree in electrical and computer engineering from the State University of New York at Buffalo, and the PhD degree in computer science from the University of Massachusetts at Amherst in 1992. He is currently the senior technical director of the Vision and Robotics Laboratory at Sarnoff Corporation, Princeton, New Jersey. Prior to joining Sarnoff, he was employed at IBM. His technical interests are in the areas of computer vision, computer graphics, image processing, and multimedia. At Sarnoff, he has been directing and performing commercial and government research and development projects in the areas of visual navigation, video surveillance and monitoring, video and 3D exploitation and analysis, object recognition, immersive tele-presence, simulation and training, 3D modeling, medical image analysis, and multisensor registration. He was one of the principal founders from Sarnoff for multiple spin-off and spin-in companies: VideoBrush, LifeClips, and Pyramid Vision Technologies. He was an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* from 1999 to 2003. He has served in different capacities on a number of computer vision conferences and US National Science Foundation review panels. He has coauthored one book on video registration, more than 50 research publications, and has received more than 22 patents, with numerous others pending. He is a member of the IEEE.



Ying Shan received the BE degree in chemical engineering, focusing on automatic process control, from Zhejiang University, People's Republic of China, in 1990, and the MS and PhD degrees in computer science from Shanghai Jiaotong University, People's Republic of China, in 1993 and 1997, respectively. Dr. Shan is currently a senior member of the technical staff in Sarnoff Corporation's Vision and Learning Laboratory. His research interests include computer vision, object recognition, machine learning, and computer graphics, with applications in video understanding, video data mining, video surveillance, 3D object and face modeling, 3D data mining, and 2D/3D image registration. From 1996 to 1997, he was a research assistant at Hong Kong Polytechnic University, Hong Kong. From 1997 to 1999, he was a postdoctoral fellow at Nanyang Technological University, Singapore, and from 1999 to 2001, he was a postdoctoral researcher at Microsoft Research, USA. Since he joined Sarnoff Corporation as a member of the technical staff in 2001, he has initiated, led, and contributed to a number of government and commercial projects that have been successfully delivered. He has published more than 25 peer-reviewed papers, holds eight US patents, and has 15 others pending. He is an active reviewer for top journals and conferences such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Image Processing*, the *International Journal on Computer Vision*, and SIGGRAPH. He was on the program committee of numerous international conferences such as ACCV, ICPR, ECCV, CVPR, and ICCV. He was the recipient of Sarnoff's Recognition Award in 2003 and the Innovation Award in 2003, 2004, and 2005. Dr. Shan is a senior member of IEEE.