Shape-based Object Recognition in Videos Using 3D Synthetic Object Models

Alexander Toshev*
GRASP Lab
University of Pennsylvania
toshev@seas.upenn.edu

Ameesh Makadia
Google Research
makadia@google.com

Kostas Daniilidis*
GRASP Lab
University of Pennsylvania
kostas@cis.upenn.edu

Abstract

In this paper we address the problem of recognizing moving objects in videos by utilizing synthetic 3D models. We use only the silhouette space of the synthetic models making thus our approach independent of appearance. To deal with the decrease in discriminability in the absence of appearance, we align sequences of object masks from video frames to paths in silhouette space. We extract object silhouettes from video by an integration of feature tracking, motion grouping of tracks, and co-segmentation of successive frames. Subsequently, the object masks from the video are matched to 3D model silhouettes in a robust matching and alignment phase. The result is a matching score for every 3D model to the video, along with a pose alignment of the model to the video. Promising experimental results indicate that a purely shape-based matching scheme driven by synthetic 3D models can be successfully applied for object recognition in videos.

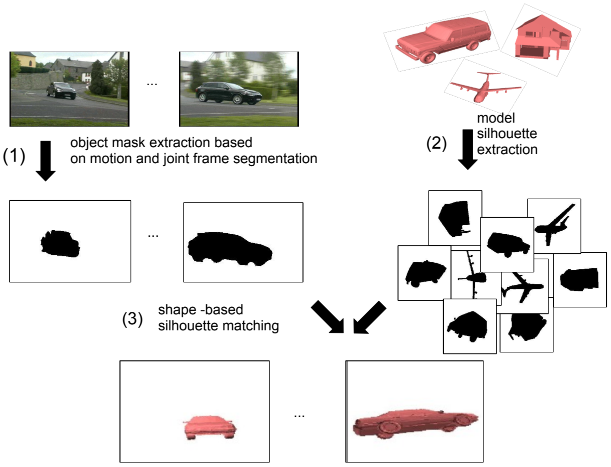


Figure 1. Recognition in videos by matching the shapes of object silhouettes obtained using motion segmentation with silhouettes obtained from 3D models.

1. Introduction

Many object recognition approaches rely on learning a 2D bag of features or feature constellations from a set of limited views as representation for recognition. This has been facilitated in the last decade through the plethora of images on the Internet as well as with the systematic annotation and construction of image benchmarks and corpora [4]. In general, representations learnt from images have a difficulty in leveraging properties of 3D shape for recognition. However, recent advances in range sensor technology, as well as easy to use 3D design tools, have enabled significant collections of 3D models in the form of VRML descriptions[1] or even unorganized point clouds. Use of 3D models makes a recognition system immune to intra-class texture variations and it frees us from the burden to capture as many views as possible. However, it comes with the cost that we cannot make use of the discriminative properties of appearances. As an intermediate representation of a 3D object we propose its aspect graph (here we really mean a compact representation of the set of all silhouettes when captured from a sphere of viewpoints, which we will refer to as a *view graph*). The use of silhouette projections gives us a stable shape representation that avoids the complexities and variability of a model's internal representation. Using the view graph, the recognition task is reduced to *shape-based* matching against extracted *silhouettes* in images. Here we work with moving objects in video – as opposed to multiple views – which facilitate foreground-background segmentation as well as a temporal coherency of the object views. We do not claim to propose a recognition system which should replace existing appearance or shape based video search engines [17]. Instead, we want to show the potential of utilizing existing collections of 3D models to accomplish recognition of moving objects in video.

In a nutshell, we propose to detect objects in videos by

*Thankful for support by NSF-IIS-0713260, NSF-IIP-0742304, NSF-IIS-0835714, and ARL/CTA DAAD19-01-2-0012.

[1] http://sketchup.google.com/3dwarehouse/

(1) extracting the silhouette of the moving object in each video frames (see sec. 3); (2) create a set of representative model views organized in a view graph (sec. 4); and (3) match the object and model silhouettes based on their shape while maintaining motion coherence over time as explained in sec. 5 (see fig. 1 for overview). Our approach provides the following contributions to the state of the art:

1. A unified framework for *detection* of moving objects as well as their *tracking* and *rough pose* estimation in videos. We show that pose estimation is possible even with similar but not exact object models and without use of explicit motion models.
2. The approach is purely *shape-based*, which frees us from the need to model highly variable appearance. Since we use videos as input, we accumulate shape information from several object views, which makes shape information discriminative.
3. Using the proposed method, 3D model datasets, which contain a large number of object classes, can be successfully applied for recognition. This is done in a plug-and-play fashion without the need of manual interaction. Additionally, 3D models give us the ability to match to any model view and thus we do not need to learn recognizers for each object view.
4. The approach relies on good motion segmentation, which in our case is achieved by jointly segmenting the video frames.

2. Related work

Much of the early work in 3D model recognition was performed by matching wire-frame representations of simple 3D polyhedral objects to detected edges in an image with no background clutter and no missing parts (a nice summary can be found in Grimson's book [7]). An exception was aspect graphs, which first appeared in [8]. Aspect graphs in their strict mathematical definition (each node sees the same set of singularities) were not considered practical enough for recognition tasks. However, the notion of sampling in the view-space for the purpose of recognition was introduced again in [2], and is the closest to our use of an aspect graph but is applied for matching synthetic 3D models.

Regarding recognition from still images, Ullman introduced the representation of 3D objects based on view exemplars [19] and several recent approaches use a sample of appearance views deliberately taken to build a model [12, 5, 13]. Savarese et al. [13] propose to learn object category models encoding shape and appearance from multiple images of the same object category by relating homographies between the same plane in multiple views. Rothganger et al. [12] extract 3D object representations based on local affine-invariant descriptors of their images and the spatial relationships between surface patches. To match them to test images, they apply appearance feature correspondences and a RANSAC procedure for selecting the inliers subject to the geometric constraints between candidate matching surface patches. In terms of models, Liebelt's approach [11] is very close to ours since it works with a view space of rendered views of 3D models. Appearance features are selected based on their discriminativity regarding aspect as well as object category and they are matched to single images in the standard benchmark datasets.

Sivic and Zisserman [16] use matching based on affine co-variant regions across multiple video frames to create models of all seen parts of 3D objects. It is the closest approach to ours regarding the nature of the data while we use an aspect representation which is the closest to [2].

3. Silhouette Extraction from Video

To recognize moving 3D objects within videos, we need to extract the necessary object shape information from the video sequences. As we described earlier, the shape information we use is a silhouette representation of the moving object in the video. To extract object silhouettes we propose a system that fuses two processes (see fig. (2)): (1) feature tracking and motion-based clustering of the resulting tracks as either object or background; (2) video segmentation into region tracks which represent parts of the scene evolving over time. In a subsequent step, we combine the feature track labeling with the segment tracks to obtain masks for the object in each frame. The motivation for this approach is that through feature tracking we can achieve robust sparse motion segmentation, while region tracks will propagate this motion segmentation to the whole image, and thus the object silhouette can be extracted. The approach is similar to [20], where the authors use a different segmentation algorithm.

(1) Sparse figure-ground labeling We assume that each video contains at most two different motions (the object and the background; the algorithm can be extended to handle multiple motions [20]) and that these two motions are well approximated by an affine motion model, motivated by large distance of the outdoor objects from the camera relative to the object depth variations.

More precisely, we seek to compute two motions $M_l = \{A_l^{(2)}, \ldots, A_l^{(T)}\}$, $l \in \{\textit{object}, \textit{background}\}$. Here $A_l^{(t)}$ is the affine motion which transforms features labeled as l from their locations in frame $t-1$ to their locations in frame t, and T is the total number of frames in the video. As we are assuming affine motion, we extract and track features using the KLT tracker [14]. The output is a collection of tracks $\bar{x}_1, \ldots, \bar{x}_n$. Each track has a start frame $\textit{start}(\bar{x}_i)$, an end frame $\textit{end}(\bar{x}_i)$, and a sequence of image locations

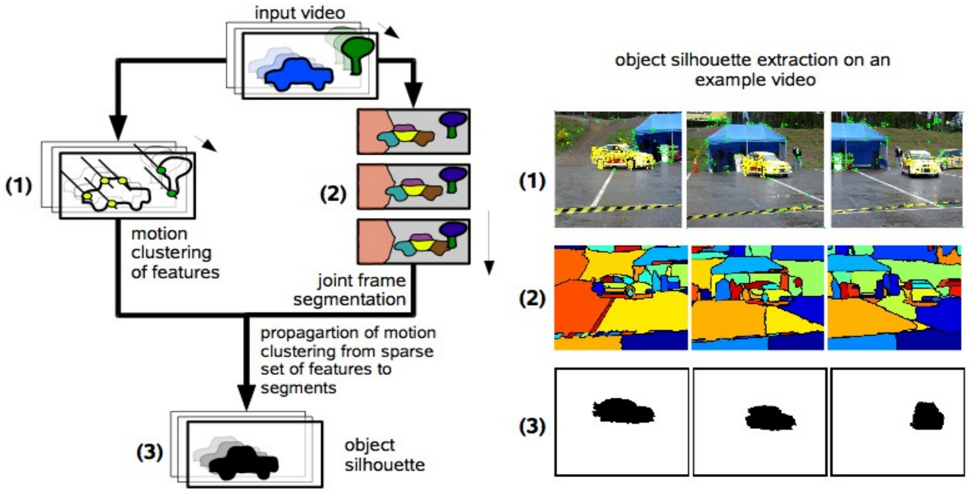

input video
(1)
motion clustering of features
(2)
joint frame segmentation
propagartion of motion clustering from sparse set of features to segments
(3)
object silhouette
object silhouette extraction on an example video
(1)
(2)
(3)

Figure 2. On the left is a schematic of the object silhouette extraction, and on the right is an example on a car video: (1) feature clustering based on common motion; (2) segment tracks; (3) object silhouette.

$\bar{x}_i = \{x_i^{(t)} | \mathit{start}(\bar{x}_i) \leq t \leq \mathit{end}(\bar{x}_i)\}$. We enforce the feature labels to be consistent over the entire track and denote this by l_i for track $\bar{x}_i$, with $L = \{l_1 \dots l_n\}$ being the set of all track labels.

Our goal is to recover the two motions $M = \{M_{\text{obj}}, M_{\text{bckg}}\}$ as well as an assignment from the feature tracks to the motions. This can be achieved by minimizing the fitting error of all tracks to their motion models:

$$E_{\text{motion}}(M, L) = E(M_{\text{obj}}, L) + E(M_{\text{bckg}}, L) \qquad (1)$$

where $E(M_l, L)$ is the fitting error of a particular motion model M_l defined as the combined fitting error of all tracks with label l w. r. t. the affine motions in M_l:

$$E(M_l, L) = \sum_{i=1}^{n} \sum_{t=\mathit{start}(\bar{x}_i)+1}^{\mathit{end}(\bar{x}_i)} \delta(l, l_i)\epsilon(A_l^{(t)}, x_i^{(t)}) \qquad (2)$$

where $\epsilon(A_l^{(t)}, x_i^{(t)}) = \|A_l^{(t)} x_i^{(t-1)} - x_i^{(t)}\|_2^2$ is the fitting error of a feature and $\delta(l, l_i) = 1$ if $l = l_i$ and 0 otherwise.

The energy in eq. (1) can be minimized by employing the EM algorithm [3]. In the E-step we assign a label l to a feature track by identifying the motion M_l with the smallest fitting error. In the M-step, we update the affine motions of M_l using the tracked features with label l. Instead of estimating the affine transformations from all the feature tracks with label l (in closed form by minimizing the least-squares error), we use RANSAC [6] to stay robust to possible outlier tracks. After the final iteration, we discard all tracks that are counted as outliers from the final RANSAC estimation. After estimating two motion models we assign label *object* to the one whose features have larger scatter in the image defined as the variance of the feature locations.

(2) Segment Tracks Due to our sole reliance on shape information for our object detection approach, we depend on the quality of the object silhouettes obtained in each frame. To achieve good object extraction we segment the frames jointly, i.e. by imposing inter-frame constraints in the segmentation cost function. This leads naturally to better accuracy – segmentation mistakes in individual frames are rectified provided the neighboring frames are partitioned correctly.

Following [18], we enforce inter-frame consistency by employing between-frame correspondences resulting from feature tracking (as described in the previous subsection). Formally, this is written as a correspondence matrix $C_{t-1,t}$, where $C_{t-1,t}(i,j) = 1$ if pixel i in frame $t-1$ and pixel j in frame t belong to the same feature track, and 0 otherwise. Additionally, we encode the similarities between pixels in the same frame in a normalized intra-frame similarity matrix W_t. As in [18] we write the intra- and inter-frame segmentation terms in one optimization problem:

$$\min_{X_t} E_{\text{seg}}(X_t) = -\sum_{t=1}^{T} \text{tr}(X_t^T W_t X_t) - \gamma \sum_{t=2}^{T} \text{tr}(X_{t-1} C_{t-1,t} X_t)$$
$$\text{subject to } X_t^T X_t = I, X_t \in \{0,1\}^{N \times k}$$

Here X_t is the indicator matrix of k segments in an image of N pixels: $X_t(i,j) = 1$ if the i^{th} pixel of the image at time t lies in segment j, and 0 otherwise. The first term corresponds to the Normalized Cuts segmentation cost [21], while the second term evaluates the similarities between segments in subsequent frames.

Since the above formulation is NP-complete, we relax the variable domain to $X_t \in \mathbb{R}^{N \times k}$. Then, if we rewrite the above cost by combining the intra- and inter-frame similarity matrices as well as the indicator matrices in one matrix as

$$W = \begin{pmatrix} W_1 & C_{1,2} & 0 & \dots \\ C_{1,2}^T & W_2 & C_{2,3} & \dots \\ \vdots & \vdots & \ddots & \end{pmatrix} \qquad X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \end{pmatrix}$$

we obtain $E_{\text{seg}}(X) = -\text{tr}(X^T W X)$, which amounts to maximizing a Rayleigh quotient. It is known that this maximum is attained for all $X = ER$, where E contains the top k eigenvectors of W and $R \in O(k)$ [21]. However, the $\text{O}(M^{3/2}k)$ complexity of an eigensolver, where M is the number of pixels in the entire video, is prohibitive even for videos containing only a few frames. Therefore, we seek a discrete solution to the above problem which is close to the relaxed optimal solutions of the individual frame segmentations $E_t R_t$ under the inter-frame similarity constraint:

$$\min_{X,R} E'_{\text{seg}}(X) = -\sum_{t=1}^{T} \text{tr}(X_t^T E_t R_t) - \gamma \sum_{t=2}^{T} \text{tr}(X_{t-1} C_{t-1,t} X_t)$$
$$\text{s.t. } X_t^T X_t = I, X_t \in \{0,1\}^{N \times k}, R_t \in O(k)$$

Here E_t contains the top k eigenvectors of W_t. We optimize the above function by iteratively minimizing E'_{seg} w. r. t. X and R. At iteration i we estimate:

1. $R_t^{(i)} = UV^T$ where ${X_t^{(i-1)}}^T E_t = VSU^T$ is the SVD decomposition. This results in minimizing E'_{seg} w. r. t. R given $X^{(i-1)}$ from the previous iteration.
2. $X_t^{(i)}$ is estimated by minimizing $-\sum_{t=1}^{T} \text{tr}(X_t^T E_t R_t^{(i-1)}) + \gamma \sum_{t=2}^{T} \text{tr}(X_{t-1}^{(i-1)} C_{t-1,t} X_t)$ given $R^{(i-1)}$ which is the solution of a linear program.

The initial values $X_t^{(1)}$ and $R_t^{(1)}$ are obtained by optimizing the intra-frame term of E'_{seg} as in [21]. Details for many of the derivations presented above can be found in [21, 18].

Following the approach above, we can generate a segmentation for each frame that incorporates the constraints of the inter-frame feature tracking. Next we will show how to group the segmentations in each frame to extract the foreground object silhouettes.

(3) Object Silhouette Detection Now we show how to use the track labels and the segments to obtain object silhouettes for the foreground object in each frame. The main idea is to propagate the motion labeling of the tracks to labels of the segments, while maintaing spatial and temporal smoothness of the segment labeling. We model the interplay between tracks and segments using an MRF over the segments $S = \{s_1 \dots s_K\}$ from all frames, $s_i \in \{\textit{object}, \textit{background}\}$ denoting the label of the i^{th} segment. The energy function

$$E_{\text{silhouette}}(S) = \sum_{i=1}^{K} E_{\text{propg}}(s_i) + \sum_{i,j=1}^{K} E_{\text{smooth}}(s_i, s_j) \quad (3)$$

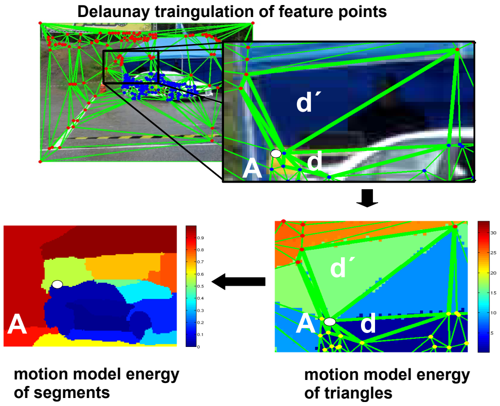

Delaunay traingulation of feature points
d´
A
d
d´
A
d
A
motion model energy of segments
motion model energy of triangles

Figure 3. Upper left: all features with their motion (blue denotes object, red - background) and their Delaunay triangulation; upper right - a zoom in of feature A and its triangles; lower right: motion model energy of each triangle (dark blue means object); lower left: propagation of the triangle energy to the segments. (For further explanation see text.)

consists of unitary energy term E_{propg} which propagates motion labels of features to segments, and a term E_{smooth} assuring that the resulting labels do not violate spatial and temporal smoothness. A detailed definition of both terms follows below.

Propagating from feature tracks to segments Although the segment tracks provide good segmentation, the segment boundaries are not precise enough to measure directly their compatibility with the estimated affine motion models. Therefore, we propose to label the segments by propagating labels from features to the segments. A simple way could be through assigning features to segments based on their locations. However, many of the tracked features tend to lie close to the boundary of a segment which makes feature-to-segment assignment ambiguous (see feature A in fig. (3)). To resolve this problem, we propose to use the Delaunay triangulation of the features. We define a motion energy term for each of the resulting triangles $\{d_1, \dots, d_m\}$ (we denote by d_i the i^{th} triangle as well as its label) based on the average fitting error of its vertices: $E_{\text{tri}}(d_i = l) = \frac{1}{3}\sum_{x \in d_i} \epsilon(A_l, x)$ where ϵ is the fitting error as defined in eq. (2) applied for the motion model A_l of the frame of d_i. The motivation for this definition is that we can assign a motion label robustly to the triangles since a triangle accumulates information from several vertices. For example, in fig. (3) the boundary point A, moving with the object, is the only vertex labeled as *object* on the background triangle d', while it correctly supports triangle d to be labeled as object. We propagate the motion label energy of the triangles to entire segments as the average of the triangle energies weighted by the area overlap o_{ij} between the i^{th} segment and j^{th} triangle:

$$E_{\text{seg}}(s_i) = \frac{1}{O_i} \sum_{j=1}^{m} o_{ij} E_{\text{tri}}(d_j = s_i) \qquad (4)$$

where $O_i = \sum_{j=1}^{m} o_{i,j}$.

Spatial and temporal smoothness The smoothness constraint is imposed among neighboring segments using a Potts smoothness term and is based on the normalized color histogram h_i of the segments:

$$E_{\text{smooth}}(s_i, s_j) = \alpha + (1-\alpha)\exp\left(-\frac{\|h_i - h_j\|^2}{2\sigma_c^2}\right)$$

if the j^{th} segment is in the neighborhood of the i^{th} one and $s_i \neq s_j$, 0 otherwise. The neighborhood of a segment i is defined as the set of segments which are either in the same frame and share a common boundary, or are in the consecutive frame and share a common boundary with a segment from the segment track of segment i. In this way we incorporate spatial as well as temporal relationships. In our experiments, we use 6-dim. RGB color histogram and set $\alpha = 0.2$, $\sigma_c = 0.3$, and $b = 6$.

Finally, the labeling of the segments S is obtained by minimizing the energy in eq. (3). This is a submodular binary labeling problem, hence it can be solved exactly using Graph Cuts [9].

4. Model View Graph

The matching of video frames to 3D model silhouettes requires a compact yet complete model representation. We propose to use a small set of model silhouettes, called *view graph*[2] (see fig. 4). Each silhouette is a compact representation of a subset of all possible model views, while the whole graph covers the entire viewing sphere. The edges connect neighboring silhouettes represent view transitions which can be induced by ego-motion of the model.

To create such a view graph, we orthographically render a large number (500 in our experiments) of silhouettes from approximately uniformly distributed viewing angles and cluster them into a few representative views. However, for 260 models, which is the dataset size we use, this results in 130000 silhouettes, which not only contain redundant information, but also pose a computational challenge for matching. To obtain the graph, we perform k-medoids clustering of the 500 silhouettes with 20 modes [2].

The clustering for view graph creation requires a feature vector for each silhouette representing its shape. We compute shape contexts [1] centered at silhouette boundary points (we uniformly sample 20% of the boundary). Since a shape context is not rotation invariant, each silhouette is pre-rotated to a canonical orientation, given by the rotation that brings the offset vector q (which connects the offset of the shape context to the centroid of the silhouette) to the X-axis. Hence, the descriptor $\text{sd}_k = (\text{sc}_k, q_k)$ assigned to boundary point k contains both the shape context sc_k and the offset vector q_k. We use the extracted shape contexts from all silhouettes to compute a codebook of size 200, using k-means clustering. Using nearest neighbor binning we can build a histogram over codewords for each silhouette image. The resulting 200-dimensional histogram vector is used for the view clustering.

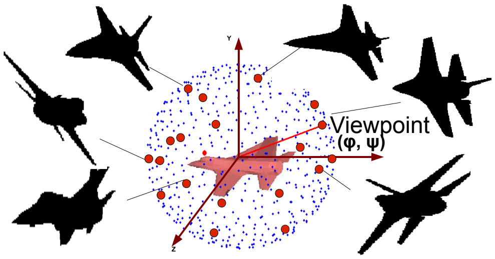


Figure 4. View graph: 500 viewing points (blue) extracted initially and the view graph (red) after clustering. Some of the silhouettes are displayed.

5. Matching of Object Silhouette Sequences to Models

After we have extracted a sequence of silhouettes from the video and a set of model silhouettes organized in a view graph, we cast the matching problem as alignment of the object silhouettes with the view graph (see fig. 5). The alignment procedure should measure the shape similarity between the video and model silhouettes, while assuring smooth model view transitions on the view graph consistent with the sequence of video silhouettes. The benefit of the approach is two-fold – the best aligned model gives the class of the observed object in the video, while the path on the view graph gives a rough pose estimate of the object motion.

More precisely, for a given video silhouette sequence $F = \{f_1 \dots f_T\}$ we seek a model silhouette sequence $V = \{m_1 \dots m_T\}$, where each model silhouette $m_i = (v_i, \theta_i, s_i, p_i)$ is parametrized by the its shape v_i, its orientation θ_i around the silhouette centroid, its scale s_i, and the viewpoint $p_i = p_i(\varphi_i, \psi_i)^T$ on the model view sphere from which it was extracted. Then we can use a conditional random field [10] to define a joint distribution over the aligned model views V (see fig. 5 as well):

$$P(V|F) = \frac{1}{Z(F)} \prod_{i}^{T} P(m_i|F) P(m_i, m_{i-1}|F) \qquad (5)$$

[2] We deviate from [2] and do not use the term *aspect graph* because we do not follow its mathematical definition.

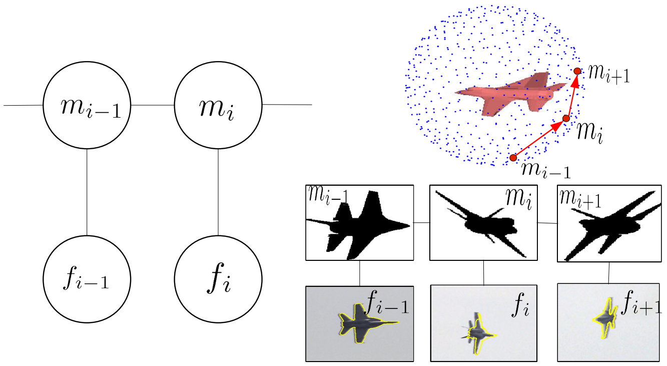


Figure 5. Left: CRF model of the video-to-model alignment. Right: alignment shown for 3 frames of a video and a model view graph.

where $Z(F)$ is the partition function. Estimating the model silhouette sequence V for which the above distribution is maximized amounts to finding the alignment with the highest score. This inference can be solved exactly by backwards-forwards algorithm for CRFs [10]. Below we define the different terms.

Shape matching. As explained in sec. 4, we use rotation invariant shape contexts. We denote a correspondence between boundary point $\text{sd}_k = (\text{sc}_k, q_k)$ of the object silhouette and a boundary point $\text{sd}_n = (\text{sc}_n, q_n)$ of the model silhouette by $c_{k,n}$. The L_2 distance between the shape contexts sc_k and sc_n is a measure of the semi-local shape similarity of the silhouettes and can be used to define a probability for observing model silhouette shape v and the point correspondence $c_{k,n}$:

$$P(c_{k,n}, v|f_i) = \exp(-||\text{sc}_k - \text{sc}_n||^2/2\sigma_{\text{sc}}^2) \qquad (6)$$

Besides local similarity, we also evaluate the global shape similarity by measuring how well individual point matches agree with each other. For this we need a parameterization of the alignment of the model with the object. Since we know the centroids of the silhouettes, the alignment can be parameterized by a similarity transformation with a zero translational component. It is defined by a rotation $R(\theta) \in SO(2)$ around the mask centroid and a scale s. Hence this is the transformation which aligns the offset vectors q_k and q_n of the shape contexts. The probability of this alignment given the correspondence $c_{k,n}$ is:

$$P(\theta, s|c_{k,n}, v, f_i) = \exp(-||sR(\theta)q_k - q_n||^2/2\sigma_{\text{sp}}^2) \qquad (7)$$

By combining all shape context similarities and the global alignment the probability of a matching model silhouette $m = (v, \theta, s)$ can be written as:

$$P(m|f_i) = \frac{1}{A}\sum_{k,n} P(\theta, s|c_{k,n}, v, f_i)P(c_{k,n}, v|f_i) \qquad (8)$$

where A is a normalization factor guaranteeing that the above quantity is a probability distribution. We use eq. (8) to define the first term in the CRF in eq. (5).

Transition smoothness in the view graph. The second term $P(m_i, m_{i-1}|F)$ represents the transition of the model silhouette at time $i-1$ to the one at time i. We require smooth transitions – the viewing points p as well as the alignment defined as rotation θ to corresponding frames (see previous section) should be similar:

$$P(m_i, m_{i-1}|F) = \exp\left(-\frac{(\theta_i - \theta_{i-1})^2}{2\sigma_r^2} - \frac{\text{acos}(p_i^T p_{i-1})^2}{2\sigma_p^2}\right)$$

Voting for initial object detection. To perform full recognition we need to solve the problem in eq. (5) for each model. Since this can be computationally challenging, in a first step we use shape information from each video frame independently to detect a few matching models for the whole video – each individual frame votes for the best matching model class and we retain the class with largest votes. The full model from eq. (5) is solved in a second step only for models from the best matching class.

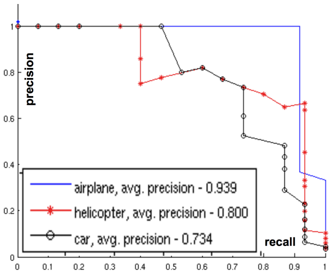
More precisely, we use the shape probability from eq. (8) to compute a score of the best alignment between each video frame and each model view: $s(v, f_i) = \max_{\theta,s} P(\theta, s, v|f_i)$. The maximum is computed using Hough transform – each match $c_{k,n}$ casts a vote for a rotation and scale and we pick the ones with largest vote accumulation. Having already computed a matching score we can use it to define the weight with which a frame f_i casts a vote for a model class l as $w_l(f_i) = \sum_v \delta(\textit{label}(v), l)s(v, f_i)$, where we sum over all model silhouettes in the model dataset which have label l. The model class assigned to the input video is the one with the highest vote from all frames: $\textit{label}(F) = \arg\max_l \sum_{i=1}^{T} w_l(f_i)$.

We used the following parameter values: $\sigma_{\text{sc}} = 0.25$ in eq. (6), $\sigma_{\text{sp}} = 4$ in eq. (7), and $\sigma_p = 60°$, $\sigma_r = 20°$. We rescale model silhouettes to have variance 70 pixels.

6. Experimental Evaluation

We perform experimental evaluation of our approach using 42 videos representing 3 different classes: *car* (15 videos), *airplane* (12 videos), and *helicopter* (15 videos). These videos are between 50 and 1000 frames and were collected manually or from the web.

We obtain 3D models from the Princeton Shape Benchmark [15], which contains a variety of different objects. We use 52 classes, 5 models per class. The classes represent a rich variety of object types – vehicles, animals, furniture, architectural elements, etc. In particular, the dataset contains classes of type *car* and *helicopter* and 5 classes of type *airplane* (*passenger plane*, *jet*, *F117*, *biplane*, and *space_shuttle*). Note that we initially extract 500 views per



class	accuracy
car	80.0%
airplane	91.7%
helicopter	80.0%

Figure 6. Left: precision-recall curves. Right: recognition accuracy.

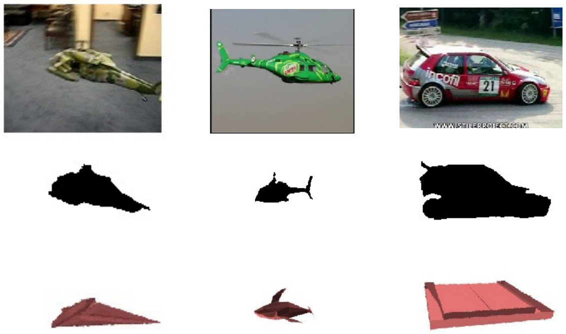


Figure 7. Failure cases for the matching (first row – frame, second row – object mask, third row - best model match).

model, resulting in a total 130000 silhouettes for the 260 models we use, many of which contain unrealistic views, e. g. bottom of a sailboat, which make the matching problem even harder. Moreover, although we test on 3 video classes, we use all 52 model classes in order to test the robustness of our system to shape variation. The goal is to utilize the whole shape dataset without any manual selection.

We match the input videos to the models and determine a model class using the voting procedure of sec. 5. We achieve accuracy of 83% over all videos (see fig. 6 and fig. 8). The results show that we can robustly detect the correct object class using a textureless dataset of models without being confused by the large variety of object types. The few mistakes are result of shape ambiguities. For example, in fig. 7 we can see incorrect matches due to very similar model object outlines. However, due to object motion we always see one or several discriminative object shapes, which decrease the effect of ambiguous shapes. Using the alignment procedure of eq. (5) we can achieve good alignment of the model with the video, and thus estimate the rough pose of the moving object (see fig. 8).

7. Conclusion

Sivic and Zisserman identified 3D object retrieval as one of the three open challenges in video retrieval [17]. We believe that our approach made progress towards this goal by combining two methods: a joint-frame segmentation and motion grouping in video with a view-space of silhouettes as object representation. Our approach is different than related work in being independent of appearance and insensitive to viewpoint-induced shape variation. We achieve this by utilizing large 3D model datasets containing more than 50 different model classes. We empirically show that the presented approach can robustly recognize objects in videos and additionally estimate their rough pose by using only similar but not exact 3D models.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(4):509–522, 2002.

[2] C. Cyr and B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. *IJCV*, 57(1):5–22, 2004.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[4] M. Everingham and et. al. The 2005 pascal visual object classes challenge. In *MLCW*, pages 117–176, 2005.

[5] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *IEEE CVPR*, 2004.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[7] W. Grimson. *Object recognition by computer: The role of geometric constraints*. The MIT Press, Cambridge, MA, 1990.

[8] J. Koenderink and A. Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE TPAMI*, 26:65–81, 2002.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2003.

[11] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D Feature Maps. In *IEEE CVPR*, 2008.

[12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-view Spatial Constraints. In *IEEE CVPR*, 2003.

[13] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *IEEE ICCV*, 2007.

[14] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR*, 1994.

[15] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, Genova, Italy, 2004.

[16] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object Level Grouping for Video Shots. *IJCV*, 67:189–210, 2006.

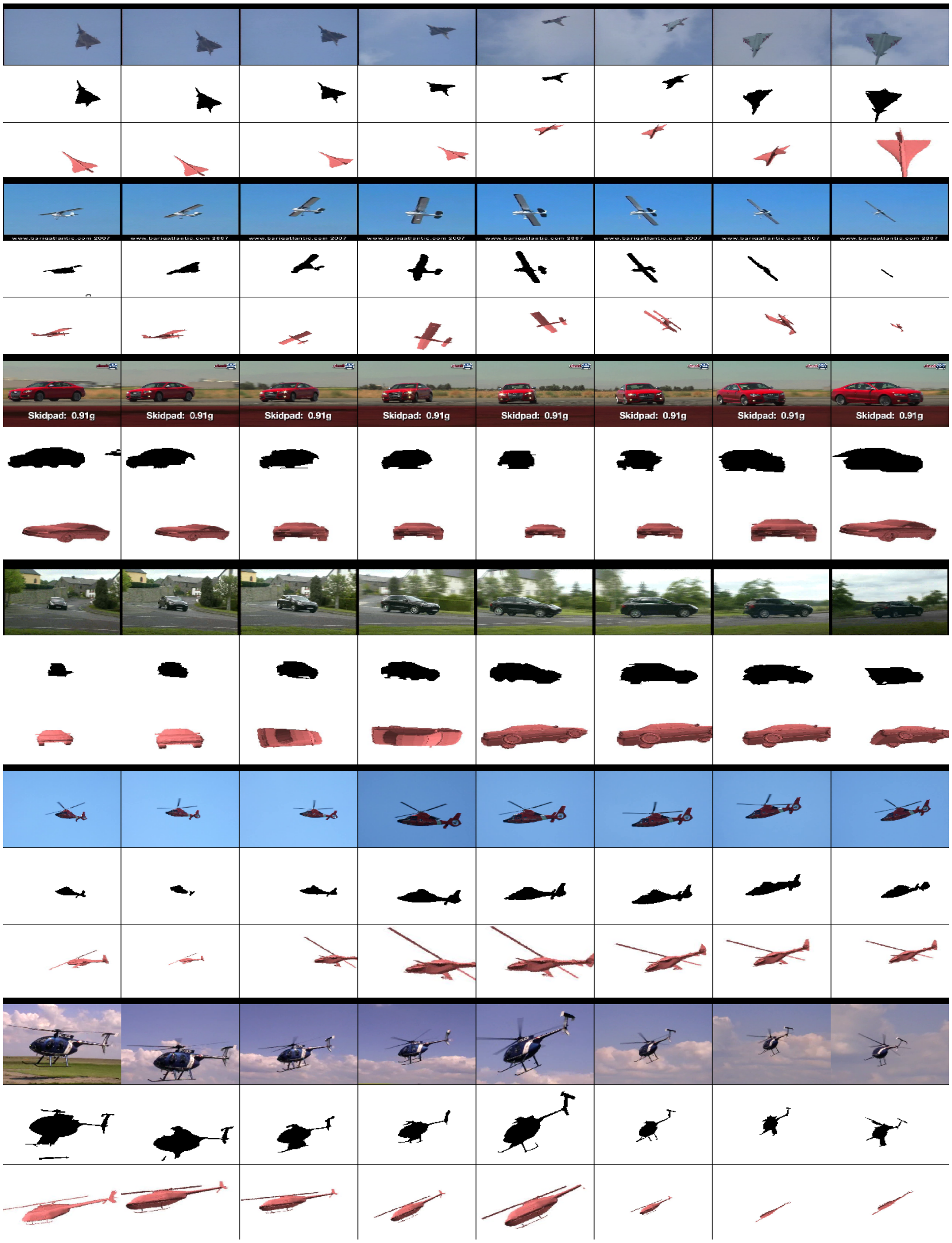


Figure 8. Matching results and model alignment for 6 videos. For each example, we show in three rows the input video, the detected silhouette sequence, and the aligned matched model (we sample 8 equally spaced in time frames from each of the displayed videos).

[17] J. Sivic and A. Zisserman. Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566, 2008.

[18] A. Toshev, J. Shi, and K. Daniilidis. Image matching via saliency region correspondences. In *IEEE CVPR*, 2007.

[19] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE TPAMI*, 13:992–1006, 1991.

[20] J. Wills, S. Agarwal, and S. Belongie. What went where. In *IEEE CVPR*, 2003.

[21] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE ICCV*, 2003.