

Spatiotemporal Contour Grouping using Abstract Part Models

Pablo Sala¹, Diego Macrini², and Sven Dickinson¹

¹ University of Toronto, ² Queen's University

Abstract. In recent work [1], we introduced a framework for model-based perceptual grouping and shape abstraction using a vocabulary of simple part shapes. Given a user-defined vocabulary of simple abstract parts, the framework grouped image contours whose abstract shape was consistent with one of the part models. While the results showed promise, the representational gap between the actual image contours that make up an exemplar shape and the contours that make up an abstract part model is significant, and an abstraction of a group of image contours may be consistent with more than one part model; therefore, while recall of ground-truth parts was good, precision was poor. In this paper, we address the precision problem by moving the camera and exploiting spatiotemporal constraints in the grouping process. We introduce a novel probabilistic, graph-theoretic formulation of the problem, in which the spatiotemporal consistency of a perceptual group under camera motion is learned from a set of training sequences. In a set of comprehensive experiments, we demonstrate (not surprisingly) how a spatiotemporal framework for part-based perceptual grouping significantly outperforms a static image version.

1 Introduction

Interest in the perceptual grouping of image contours peaked in the late 1990's, when the mainstream object recognition community was primarily shape-based and the bottom-up recovery of distinctive indexing structures was critical in identifying a small number of candidate objects (from a large database) present in the scene. However, the advent of appearance-based recognition (and a corresponding movement away from shape), combined with the reformulation of the recognition problem as a detection problem (in which the image is searched for a single target object), diminished the role of perceptual grouping. Even with the re-emergence of image contours as the basis for categorical models (e.g., [2]), the continuing focus on object detection means that the stronger shape prior offered by a detector subsumes the domain-independent shape priors that make up the non-accidental properties that define perceptual grouping. In other words, the process of domain-independent, bottom-up perceptual grouping to extract a meaningful indexing structure in order to select promising candidates is unnecessary, since in a detection task we know what, i.e., which candidate, we're looking for.

There are clear signs that the community is moving back toward unexpected object recognition, i.e., identifying an image of an unknown object from a large database. Since a linear search through a space of object detectors clearly does not scale to large databases, we must drastically prune the space of candidate detectors to apply to the image. This, in turn, means recovering distinctive image structures that can effect such pruning – a return to perceptual grouping. Yet a simple return to classical grouping techniques is insufficient, for while non-accidentally related contours in an image may be grouped, there is still a semantic gap between the resulting contour groups and the shape structures that comprise a categorical shape model. Only when the contour groups are *abstracted* can they be matched to categorical models.

In recent work [1], we developed a framework in which a small vocabulary of abstract part shape models were used to both group and abstract image contours, yielding a covering of the image with a set of 2-D abstract parts which model the projections of the surfaces of a set of abstract volumetric parts that describe the coarse shape of the object. Thus, rather than invoking an object-level shape prior (detector), which we don't have since we don't know what we're looking at, we instead invoke a small, finite set of intermediate-level, domain-independent shape priors to drive the grouping and abstraction processes (we assume only that the parts can be assembled to describe a significant portion of any object in the database). While the method shows clear promise, there is a fundamental trade-off between abstraction and ambiguity; as a greater degree of abstraction of a set of image contours is allowed, the more ambiguous the abstraction, i.e., the abstraction is consistent with an increasing number of shape models.

In this paper, we exploit the dimension of temporal coherence to help cope with the ambiguity of a shape abstraction inherent in a single static image. Like in [1], we rely on a small, user-defined, abstract shape vocabulary to drive the process of perceptual grouping in a single frame. However, unlike [1], which restricts its analysis to a single image, we assume access to a video sequence in which there is relative motion between the camera and the object, and exploit the spatiotemporal coherence of a perceptual group to reduce false positives that are abundant in a single image. If a perceptual group of contours is consistent with an abstract part model, and is stable over time in terms of its shape (continues to match the same part model) and pose, then we consider the perceptual group to be non-accidental. We introduce a novel probabilistic, graph-theoretic formulation of the problem, in which the spatiotemporal consistency of a perceptual group under camera motion is learned from a set of training sequences. In a set of comprehensive experiments, we demonstrate (not surprisingly) how a spatiotemporal framework for part-based perceptual grouping significantly outperforms a static image version.

2 Related Work

The problem of using simple shape models to group and regularize 2-D contour data has been extensively studied in the past. Many have approached this prob-

lem assuming figure-ground segmentation, i.e., they take as input a silhouette, while others have assumed knowledge of the object present in the scene, i.e., object-level shape priors. In our approach, we assume neither; rather, we adopt the classical perceptual grouping position and assume only mid-level shape priors. In the relevant work on this topic, such priors can range from simple smoothness to compactness to convexity to symmetry to more elaborate part models, but stop short of object models.

The non-accidental regularity of convexity to group contours into convex parts has been explored by Jacobs [3] and by Estrada and Jepson [4], to name just two examples. Stahl and Wang [5] explored the non-accidental regularity of symmetry to group contours into symmetric parts, while Lindeberg [6] has explored symmetry to extract symmetric blobs and ridges directly from image data. Although a particular non-accidental shape regularity is exploited by each of these models, they also restrict the image domain. Furthermore, there is little to unify the approaches, since each mid-level shape prior comes with its own computational model.

The early recognition-by-parts paradigm yielded more powerful part models. Pentland [7] partitioned a binary image into 2-D parts corresponding to the projections of a vocabulary of 3-D deformable superquadrics. His method was never applied to contours, since its main focus was more on the problem of part selection (from a large set of part hypotheses) than the grouping of features into parts. Dickinson et al. [8] used part-based aspects (representing the possible views of a vocabulary of volumetric parts) to cover the contours in an image. Pilu and Fisher [9], sought to recover 2-D deformable part models from image contours. Nonetheless, all these approaches were restricted to scenes containing very simple objects, since they assumed a one-to-one correspondence between image and model contours. These systems achieved little, if any, true abstraction and were rarely, if ever, applied to textured objects.

Fitting part models to regions is the dual problem of fitting part models to contours. A method to find instances of a 2-D shape (possibly a part model) in an image was proposed by Liu and Sclaroff ([10]). Taking as input a bottom-up image region segmentation, they explore the space of region merges and splits, searching for region groups whose shapes are similar to a 2-D statistical template model. Also starting with a bottom-up region segmentation, Wang et al.'s approach [11] searches for region groups having a particular shape via a stochastic framework that explores the space of region merges and splits. These approaches, however, not only admit a single model shape, but their grouping process is heavily driven by appearance homogeneity. Furthermore, Wang et al.'s method does not attempt shape abstraction, employing a very detailed model of the shape.

Although we know of no approaches dealing with the problem of finding spatiotemporally coherent perceptual groups, this can be considered, in a sense, to be similar to the tracking problem. Tracking approaches often require some type of initialization to indicate the location, in an initial frame, of the region or object of interest that is to be tracked. Moreover, if during the tracking

process the tracker’s focus of attention drifts away from the objects of interest, some recovery mechanism needs to be in place to recover from such errors. Our method, however, requires neither an initialization nor a drift-recovery step, since the hypothesis detection process applied at each frame acts as an interest operator, yielding the set of image regions of interest in each frame.

The solution proposed in this paper to the problem of determining multiple sequences (i.e., trajectories) of closed contours, each corresponding to the boundary of a particular object surface across frames, is formulated in graph-theoretical and probabilistic terms, and solved efficiently using the Viterbi algorithm. Quach and Farooq [12] have applied Viterbi to solve the data association problem for single-target tracking in a maximum likelihood fashion, assuming that object motion is a Markov process. More recently, Yan et al. [13] have used Viterbi for single-target tracking of a tennis ball in video. These approaches only admit a single-target, and require both an initialization step and a step to identify the object of interest at the end of the sequence. Our method, however, is not only multi-target, modeling both shape and appearance to disambiguate surface correspondences across frames, but also does not require any type of initialization or recovery mechanism. Moreover, our formulation models second-order relationships between the position, orientation and scale of the surface contours across frames rather than simply modeling first-order smoothness of the tracked feature’s location across frames.

3 Overview of the Approach

The input to our perceptual grouping framework is a video sequence and a vocabulary of shape primitives. First, hypotheses are independently recovered from each frame using the method proposed in [1]. Specifically, we begin by computing a region oversegmentation (Figure 1(b)) of the frame (Figure 1(a)). The resulting region boundaries yield a *region boundary graph* (Figure 1(c)), in which nodes represent region boundary junctions where three or more regions meet, and edges represent the region boundaries between nodes; the region boundary graph is a multigraph, since there may be multiple edges between two nodes. We cast the problem of grouping regions into perceptually coherent shapes as finding simple cycles in the region boundary graph whose shape is “consistent” with one of the model shapes in the input vocabulary (Figure 1(d)); these are called *consistent cycles*. Since the number of simple cycles in a planar graph [14] is exponential, simply enumerating all cycles (e.g., [15]) and comparing their shapes to the model shapes is intractable. Instead, we start from an initial set of single-edge paths and extend these paths (see Section 4.1), called *consistent paths*, as long as their shapes are consistent with a part of *some* model. To determine whether a certain path is consistent (and therefore extendable), the path is approximated at multiple scales with a set of polylines (piecewise linear approximations), and each polyline is classified using a one-class classifier trained on the set of training shapes (Figure 1(e)). When a consistent path is also a simple cycle, it is added to the set of output consistent cycles (Figure 1(f)).

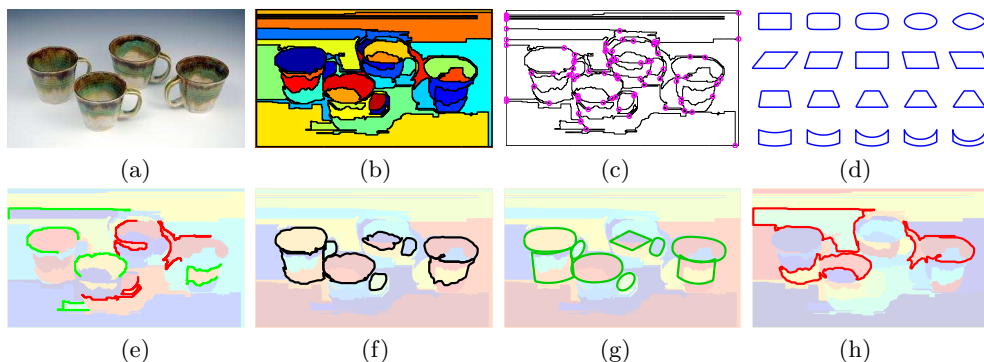


Fig. 1. Problem Formulation: (a) input image; (b) region oversegmentation; (c) region boundary graph; (d) example vocabulary of shape models (used in our experiments); (e) example paths through the region boundary graph that are consistent (green) and inconsistent (red); (f) example detected cycles that are consistent with some model in the vocabulary; (g) abstractions of cycles consistent with some model; (h) example cycles inconsistent with all models.

Figure 1(d) shows the input vocabulary used in our experiments: four part classes (superellipses plus sheared, tapered, and bent rectangles, representing the rows) along with a few examples of their many within-class deformations (representing the columns). Each shape model is allowed to anisotropically scale in the x- and y-directions, rotate in the image plane, and vary its deformation parameters (e.g., shearing, tapering, bending).

The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. However, as mentioned in Section 1, the consistent cycle classifier may yield many false positives at reasonable recall rates. Some of the recovered consistent cycles may yield shapes that are qualitatively different from those in the vocabulary, while in other cases the shapes may be consistent but accidental, e.g., a number of the detected consistent cycles might not correspond to actual scene surfaces. By exploiting spatiotemporal consistency of these consistent cycles across a video sequence, we can filter out many of these false positives. That is, we assume that the only cycles that are likely to be caused by the projection of an actual scene surface are those whose shape and internal appearance remain stable or vary smoothly across consecutive frames.

We formulate the problem of finding sequences of consistent cycles with temporally coherent shapes across frames of a video sequence in graph-theoretical and probabilistic terms. We refer to such sequences as *trajectories*. The potential correspondences between consistent cycles detected at different frames are modeled by constructing a graph in which a maximum-weight path corresponds to a trajectory with maximum joint probability of including all and only those consistent cycles in the sequence that correspond to the same scene’s surface boundary. Specifically, nodes in the graph encode pairs of potential matches between consistent cycles in nearby frames, edges connect pairs of nodes that

share a common consistent cycle, and edge weights encode the probability of correctness of the cycle matches connected by the edge conditioned on geometric and photometric properties of the cycles involved. We learn this probability distribution from a few hand-labeled training sequences. The top trajectories of temporally coherent consistent cycles are obtained by iteratively applying the Viterbi algorithm on the graph to find paths with maximum joint probability, and removing from the graph the nodes involved in such paths.

4 Detecting Consistent Cycles

In the following subsections, we review the steps of our algorithm, described in [1], for finding consistent cycles in a single frame, i.e., cycles whose shape is consistent with one of the model shapes. The two main steps of the algorithm are *path initialization* and *path extension*. In Section 5, we introduce the temporal coherence constraint to our grouping framework.

4.1 Path Initialization

The first step in the algorithm generates an initial set of single-edge paths that will be iteratively extended into cycles by repeated executions of the path extension step. This set of edges should be as least redundant as possible, to avoid generating the same cycle more than once (from different edges in the same cycle). Moreover, all possible graph cycles should be realizable by path extensions starting from edges in this set. Such an optimal set corresponds to the *feedback edge set*, which is the smallest set of edges whose deletion results in an acyclic graph. This initial set of single-edge paths are added to the queue of paths to be extended.

4.2 Path Extension

At each algorithm iteration, one of the paths is taken off the queue. If the path is a cycle and it is consistent with at least one of the shapes in the vocabulary of model shapes, the cycle is added to the output list of consistent cycles. If, however, the path is not a cycle, its consistency is also checked. If the path is consistent with a portion of the boundary of at least one shape in the vocabulary, then the path’s possible extensions by a single edge are added to the queue. The algorithm continues until the queue is empty, and then outputs the consistent cycles.

Consistency of a cycle or path is checked by first approximating the shape of the cycle or path with a polyline computed at different scales using the Ramer-Douglas-Peucker algorithm [16]. For each resulting polyline, a feature vector is computed, encoding the angles and normalized lengths of the linear segments making up the polyline. As illustrated in Figure 2 (a), a feature vector’s length is a function of the number of linear segments comprising the polyline. A consistency decision for a feature vector is made by a one-class classifier that determines if the feature vector is geometrically close to one of the training feature

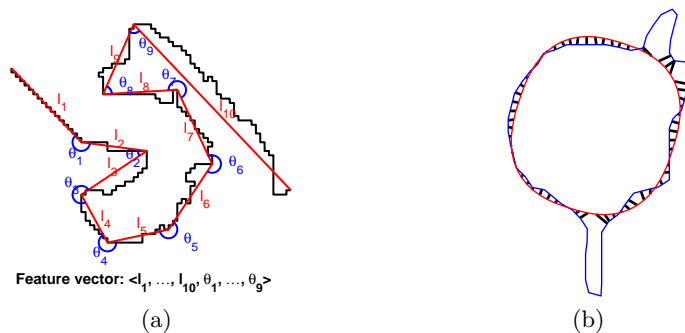


Fig. 2. (a) Feature vector computation for a polyline approximation of a contour; (b) Model-based abstraction (red) of a consistent cycle hypothesis (blue): the black line segments illustrate the distance between equidistantly sampled model points to their closest points along the hypothesis’ contour.

vectors. (Notice that since the feature vectors can have different sizes depending on the lengths of their corresponding polylines¹, there is a classifier for each possible feature vector length.) The scales at which their corresponding polylines are consistent are associated with the path. If a path at a particular scale is not consistent, then no extension of that path can be consistent at that scale. Thus, when a path is initialized, it is associated with all scales, and when it is extended, its associated scales can only remain constant or decrease. If there is no scale at which the path is consistent, the path is discarded.

4.3 Training the Classifiers

We trained the classifiers using feature vectors generated from approximately 4 million contour fragments of noisy instances of within-class deformations of each model. Feature vectors are generated from the polyline approximations (computed using a tolerance proportional to model size) of each sampled contour fragment and their dimensionality is reduced via PCA. Classification is performed on the reduced dimensionality vectors. For the model vocabulary employed in our experiments, 99% of the feature vector variance is, in general, captured by the top N PCA components for the case of feature vectors of dimension $2N - 1$, corresponding to polylines with N linear segments. We obtained very fast classification and good accuracy using as classifiers a Nearest Neighbor Data Description approach [17].

¹ The number K of linear segments comprising the longest polyline approximating a model’s contour is determined by the shapes in the vocabulary and the “level of abstraction” (i.e., tolerance, proportional to model size), used to compute the polyline approximations of training model fragments. In our implementation, $K = 13$.

5 Temporal Coherence of Consistent Cycles

We formulate the problem of finding temporally coherent consistent cycles in a video sequence in graph-theoretical terms as the search for maximum-weight paths between the source and sink nodes of a particular directed graph $G = (V, E)$. In order to obtain a more robust model of consistent cycle correspondence across frames, we not only model the first derivative of a cycle’s pose function (i.e., the cycle’s frame-to-frame change in position, scale and orientation), but we also model its second derivative, i.e., the change in the pose transformation function between corresponding consistent cycles. For this reason, instead of modeling the problem via a trellis graph, in which nodes represent consistent cycles and edges model potential cycle correspondences across frames, we actually define G as the dual of such a graph. Namely, nodes represent potential matches between consistent cycles detected in close spatial and temporal proximity, and there is an edge between each pair of nodes that share a common consistent cycle. Two special nodes, a source and a sink, also exist, which are connected to every other node in G . Edge weights correspond to a log-probability conditional to various attributes of the cycles involved in the edge, such that a maximum-weight path from source to sink corresponds to the trajectory with the highest joint probability of containing the densest sequence of correct consistent cycle matches.

5.1 Retrieving Consistent Cycle Trajectories

The construction of graph G is as follows. The set V contains two special nodes, s and t , called *source* and *sink*, respectively. All other nodes in V correspond to potential matches between consistent cycles detected at different frames and are referred to as *internal*. Formally, if \mathcal{C}_i is the set of all consistent cycles detected at frame i , the set of nodes V is defined as $V = V^{\text{internal}} \cup \{s, t\}$, where $V^{\text{internal}} \subset \bigcup_{i < j} \mathcal{C}_i \times \mathcal{C}_j$. An internal node involving consistent cycles x and y is noted by $\langle x, y \rangle$. There is an edge connecting every pair of nodes that share a common consistent cycle. The direction of these edges, referred to as *internal* edges, is determined by the frame numbers at which the non-common consistent cycles in the pair were detected. Namely, edges leave from the nodes whose non-common consistent cycles are detected at earlier frames. There is also a directed edge from s to every internal node, as well as directed edges from all internal nodes to t . The former edges called *initial*, while the latter are called *final*. Formally, $E = E^{\text{initial}} \cup E^{\text{internal}} \cup E^{\text{final}}$, where $E^{\text{initial}} = \{(s, \langle x, y \rangle) : \langle x, y \rangle \in V\}$, $E^{\text{internal}} = \{(\langle x, y \rangle, \langle y, z \rangle) : \langle x, y \rangle, \langle y, z \rangle \in V \text{ and } n(x) < n(z)\}$, and $E^{\text{final}} = \{(\langle x, y \rangle, t) : \langle x, y \rangle \in V\}$, where $n(x)$ denotes the frame at which cycle x was detected.

A match $\langle x, y \rangle$ is said to be *correct* iff consistent cycles x and y correspond to projected boundaries of the same image surface. The cardinality of V (and thus the total running time of the algorithm) can be kept low by not including in V^{internal} cycle correspondences that are highly unlikely to be correct. This can be done by assuming that a cycle undergoes smooth changes in location,

scale, shape, and appearance across frames. Therefore, potential matches can be considered only between cycles whose distance along these dimensions falls within given threshold values proportional to the distance between the frames in which they were detected. Also, consideration can be restricted to matches of cycles detected at frames that are within a specified maximum frame distance W . This maximum frame distance should be chosen such that the likelihood of a consistent cycle being undetected (e.g., due to undersegmentation) for that many consecutive frames is low.

We model the change in appearance between two potentially corresponding cycles by first approximating the shape of one of the cycles by a polygon whose vertices are points sampled at equidistant positions along the cycle. The cycle's internal appearance is then modeled by computing a homogeneous triangulation of the polygon (e.g., a Delaunay triangulation constraining triangle angles and areas to ensure an approximately uniform sampling of the image region inside the cycle at a fine enough resolution). The triangulation is then mapped onto the other cycle by means of the estimated geometrical transformation between the cycles, and their appearance distance is measured in terms of the absolute difference between sampled image color values at the centroids of corresponding triangles.

From all trajectories of consistent cycles corresponding to some particular scene surface, we are interested in finding the trajectory that is the *densest*, i.e., the one that does not miss any frame where a consistent cycle accounting for the specific surface exists. A correct match $\langle x, y \rangle$ is said to be *consecutive* iff no consistent cycle corresponding to the same surface boundary as x and y was detected in a frame $k : n(x) < k < n(y)$. Let $x \sim y$ represent the relation “ $\langle x, y \rangle$ is a correct and consecutive match”, and let $-b(x)$ ($-a(x)$) symbolize the predicate “no consistent cycle that correctly matches x was detected before (after) frame $n(x)$.” If $\langle x_i, x_j \rangle$ is a potential match, then T_{ij} represents the geometric transformation between cycles x_i and x_j . The weight $w(\cdot)$ of an edge is a log conditional probability defined depending on the type of edge:

$$w((s, \langle x_1, x_2 \rangle)) = \log(p(-b(x_1))p(x_1 \sim x_2 | \theta_{12})) \quad (1)$$

$$w(\langle \langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle \rangle) = \log(p(x_2 \sim x_3 | x_1 \sim x_2, \phi_{123})) \quad (2)$$

$$w(\langle \langle x_1, x_2 \rangle, t \rangle) = \log(p(-a(x_2))), \quad (3)$$

where $\theta_{ij} = \langle \mathbf{t}_{ij}, \delta n_{ij}, \delta sh_{ij} \rangle$ and $\phi_{ijk} = \langle \mathbf{t}_{jk}, \delta n_{jk}, \delta sh_{jk}, \delta T_{ijk} \rangle$ are attributes of the consistent cycles involved in the edge. Namely, $\mathbf{t}_{ij} \in \mathbb{R}^2$ is the change in contour position between x_i and x_j , $\delta n_{ij} = |n(x_j) - n(x_i)|$, δsh_{ij} is the shape distance between cycles x_i and x_j , and δT_{ijk} is the difference between the transforms T_{ij} and T_{jk} computed at each consistent cycle correspondence.

With this edge weight specification, a path $(s, \langle x_1, x_2 \rangle, \dots, \langle x_{r-1}, x_r \rangle, t)$ from source to sink achieving maximum weight corresponds to the trajectory of consistent cycles x_1, \dots, x_r maximizing the probability

$$p(-b(x_1))p(-a(x_r))p(x_1 \sim x_2 | \theta_{12}) \prod_{i=2}^{r-1} p(x_i \sim x_{i+1} | x_{i-1} \sim x_i, \phi_{i-1,i,i+1}). \quad (4)$$

Now, under the following natural assumptions:

1. $f \sim g$, $\neg b(f)$, and $\neg a(g)$ are mutually independent,
2. $x_i \sim x_j$ and $\phi_{k,l,m}$ are independent if $i \neq l$ or $j \neq m$, and
3. $x_i \sim x_j$ and $\theta_{l,m}$ are independent if $i \neq l$ or $j \neq m$,

equation 4 is equivalent to the joint probability

$$p\left(\neg b(x_1), x_1 \sim x_2 \sim \dots \sim x_r, \neg a(x_r) | \theta_{12}, \{\phi_{i-1,i,i+1}\}_{i=2}^{r-1}\right), \quad (5)$$

thus yielding x_1, \dots, x_r as the trajectory of consistent cycles most likely to be the longest and densest trajectory of correct consistent cycle correspondences in the video sequence. Trajectories of consistent cycles can thus be efficiently generated in decreasing order of probability by iteratively applying the Viterbi algorithm [18] on G to find the maximum-weight path from s to t , and then removing from V all internal nodes belonging to such a path.

Due to undersegmentation errors in the low-level region segmentation of a frame n , which is the input to the consistent cycle detector, it is possible that no consistent cycle is detected in frame n that corresponds to a surface boundary for which consistent cycles have been indeed detected in nearby frames. In these cases, the retrieved trajectories will be missing the frames in which the undersegmentation occurred. A surface's position and shape can however be interpolated in a missing frame from its known position and shape in nearby trajectory frames. In our approach, we compute an initial guess for the position and shape of the surface boundary in frame n by linearly interpolating the transformation between the corresponding detected consistent cycles in the closest frames around n . This guess is refined by optimizing the normalized cross-correlation between the image data internal to the consistent cycle in a nearby frame where it was detected, and the image data inside a 2-D window around the initial position estimate in frame n . The surface boundary is thus interpolated into frame n , unless the image appearance inside the contour in the estimated position of frame n and the contour appearance in the closest frames differs significantly. In that case, the surface is assumed to be occluded in frame n .

5.2 Probability Density Estimation

In order to compute the edge weights, we need to model the probability distributions involved in Equations 1, 2 and 3. By applying Bayes' rule, the probability function from Equation 1, $p(x_1 \sim x_2 | \theta_{12})$, can be rewritten as

$$\frac{p(\theta_{12} | x_1 \sim x_2) p(x_1 \sim x_2)}{p(\theta_{12} | x_1 \sim x_2) p(x_1 \sim x_2) + p(\theta_{12} | x_1 \not\sim x_2) p(x_1 \not\sim x_2)}, \quad (6)$$

and the probability function $p(x_2 \sim x_3 | x_1 \sim x_2, \phi_{123})$ from Equation 2 as:

$$\frac{p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2) p(x_2 \sim x_3, x_1 \sim x_2)}{p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2) p(x_2 \sim x_3, x_1 \sim x_2) + p(\phi_{123} | x_2 \not\sim x_3, x_1 \sim x_2) p(x_2 \not\sim x_3, x_1 \sim x_2)}. \quad (7)$$

We can thus estimate these probability distributions from training sequences.

Notice that we can factor $p(\theta_{12}|x_1 \bowtie x_2)$ as

$$p(\mathbf{t}_{12}|\delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2)p(\delta n_{12}, \delta sh_{12}|x_1 \bowtie x_2), \quad (8)$$

where $\bowtie \in \{\sim, \approx\}$. In our experiments, we quantized the space of $\langle \delta n_{12}, \delta sh_{12} \rangle$ values, discretely modeling $p(\delta n_{12}, \delta sh_{12}|x_1 \bowtie x_2)$ via a probability table. And $p(\mathbf{t}_{12}|\delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2)$ (for each quantized value of $(\delta n_{12}, \delta sh_{12})$) was modeled by a multivariate Gaussian, which appeared to be a good approximation to this distribution. Analogously, $p(\phi_{123}|x_2 \bowtie x_3, x_1 \sim x_2)$ can be factored as

$$p(\mathbf{t}_{23}, \delta T_{123}|\delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2)p(\delta n_{23}, \delta sh_{23}|x_2 \bowtie x_3, x_1 \sim x_2), \quad (9)$$

and so we modeled $p(\delta n_{23}, \delta sh_{23}|x_2 \bowtie x_3, x_1 \sim x_2)$ by a probability table, and $p(\mathbf{t}_{23}, \delta T_{123}|\delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2)$ by a multivariate Gaussian distribution for each quantized value of $(\delta n_{23}, \delta sh_{23})$. The value of $p(x_2 \bowtie x_3, x_1 \sim x_2)$ is computed directly from the training sequences. Finally, we approximated $p(-b(x))$ by $q^{n(x)-1}$ and $p(-a(x))$ by $q^{F-n(x)}$, where F is the total number of frames in the sequence and q is a tight lower bound of $p(x \sim y|n(y) = n(x) + 1)$ computed from the training sequences.

6 Results

We are not aware of any benchmark dataset for evaluating spatiotemporal contour grouping using abstract part models. Therefore, to evaluate our proposed approach, we generated an annotated dataset consisting of 12 video sequences² (a total of 484 frames), containing object exemplars whose 3-D shape can be qualitatively described by cylinders, bent or tapered cubic prisms, and ellipsoids. The visible surface contours of each object’s 3-D shape that are consistent with 2-D models from our vocabulary were hand-labeled.

Figures 3 and 4 illustrate the output of our approach on two selected frames (closer to the beginning and end) of six sequences in the dataset: row (a) shows the input frames; row (b) shows the consistent cycles closest to the ground-truth detected at each static frame (obtained by [1]); row (c) shows the temporally coherent detected consistent cycles closest to the ground-truth; and row (d) shows the ground-truth surface contours. Notice that images in rows (c) and (d) also show the boundaries of the region oversegmentation used as input to [1] (computed using the “statistical region merging” approach of Nock and Nielsen [19] with its parameters fixed for all frames from all sequences). The numbers in the top-right corner of each image in rows (b) and (c) correspond to the total number of consistent cycles in each case. The numbers appearing in the centroid of the recovered hypotheses in these rows indicate the rank of the hypothesis among all recovered hypotheses in the frame. In the case of static consistent

² Available at <http://www.cs.toronto.edu/~psala/datasets.html>.

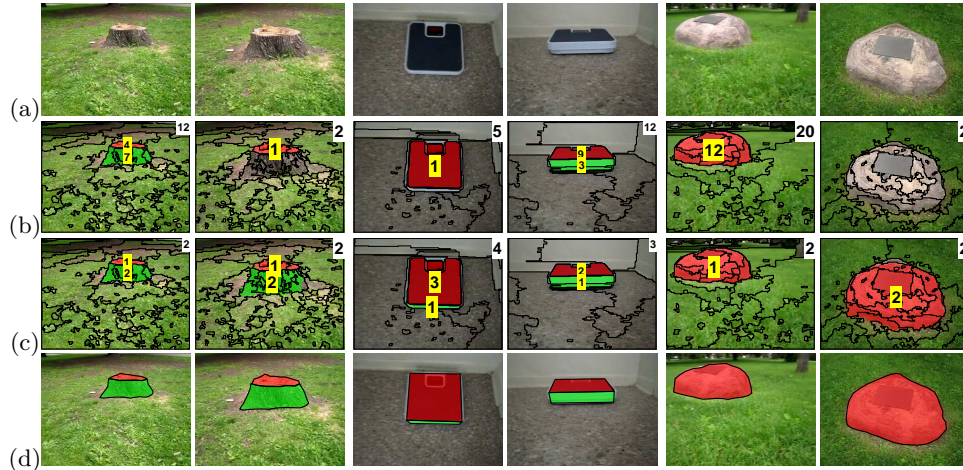


Fig. 3. Part Recovery (see text for discussion)

cycle detection, such ranking is a function of the fitting error between the consistent cycle and the model abstracting the cycle³. In the spatiotemporal case, hypotheses are ranked by the length of the consistent cycle’s temporal flow (i.e., the number of frames in which the cycle is found to be temporally consistent).

These ranking values were obtained after a non-maximum suppression step was applied to eliminate redundant cycle hypotheses in the static and dynamic cases, by discarding all but one of the similar consistent cycles competing for the same image evidence. (The cycle achieving the smallest shape distance to all other competing cycles was kept.) In the static case, as in [1], detected hypotheses with a high fitting error to their abstraction shapes were also discarded. By comparing the rankings of the recovered hypotheses corresponding to ground-truth parts in the static (row (b)) and dynamic (row (c)) cases, we can see that employing temporal coherence outperforms the static version, as the rankings in row (c) are consistently higher than those in row (b). In some cases, even the rankings of ground-truth parts in row (c) correspond to the top ones. Moreover, the total number of candidate hypotheses in the static case is generally higher than in the dynamic version, demonstrating the superior performance of the dynamic approach to prune false positive hypotheses.

A quantitative evaluation of our spatiotemporal grouping framework is shown in the precision-recall curves of Figure 5, where it is compared to [1] as a baseline. There, it can be seen that both precision and recall increase substantially

³ Abstraction of a cycle’s contour by a model in the vocabulary is accomplished via a robust active shape model fitting framework. (See [1] for details.) A hypothesis is ranked based on the average distance from equidistantly sampled points along the abstracting model’s contour to their closest points on the hypothesis’ contour, normalized by the mean distance from the hypothesis’ centroid to its contour. (See Figure 2 (b).) (As in [1], a significant portion of the hypothesis’ contour has to be explained by the model for an abstraction to be considered correct.)

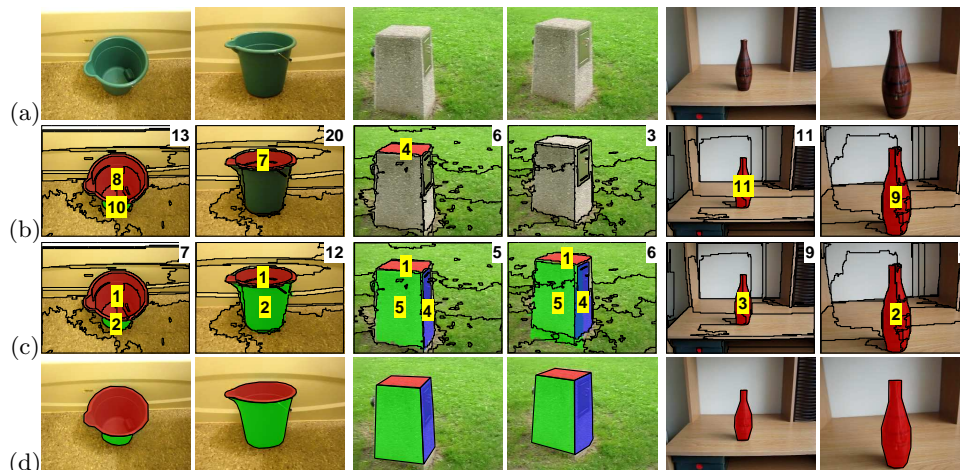


Fig. 4. Part Recovery (cont'd - see text for discussion)

when temporal coherence is taken into account. The increase in precision can be explained as the result of the pruning ability of our temporal coherence framework on false positive consistent cycles. Since such hypotheses are produced by accidental arrangements of texture or image structure in a single frame, they are unlikely to be temporally stable. Moreover, in the spatiotemporal case, hypotheses are ranked by their persistence, which proves to be a better measure of hypothesis relevance than ranking by the fitting error between a consistent cycle’s contour and its model abstraction contour, as employed in the static case. The improved recall is the result of interpolating hypotheses when gaps of false negatives (mostly due to undersegmentation) have a length not greater than the maximum frame distance W used in the construction of graph G . (In our experiments, $W = 6$.) In terms of running time, the entire process of searching for consistent cycle trajectories in a video sequence takes an average time of less than 5 seconds per frame, in our MATLAB implementation running on a laptop.

7 Conclusions

The semantic gap between real scene contours and the abstract parts that make up categorical shape models can be bridged with the help of a small vocabulary of part models. Yet as the degree of abstraction between image contours and abstract parts increases, so too does the ambiguity of a perceptual group of image contours – if abstraction is viewed as a process of “controlled hallucination”, the more you hallucinate, the greater the possible mappings to different parts. By imposing spatiotemporal constraints on the grouping process, we can significantly reduce such ambiguity, ensuring greater precision of the recovered abstract parts which, in turn, facilitates the indexing and recognition of categorical shape models.

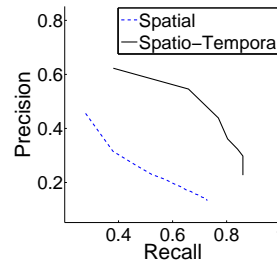


Fig. 5. Quantitative Evaluation: Precision-recall curve (see text for discussion).

References

1. Sala, P., Dickinson, S.: Contour grouping and abstraction using simple part models. In: ECCV. LNCS **6315**, Crete, Greece (2010) 603–616
2. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI **30** (2008) 36–51
3. Jacobs, D.W.: Robust and efficient detection of salient convex groups. PAMI **18** (1996) 23–37
4. Estrada, F., Jepson, A.: Perceptual grouping for contour extraction. In: ICPR. (2004)
5. Stahl, J., Wang, S.: Globally optimal grouping for symmetric boundaries. In: CVPR. (2006)
6. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. IJCV **11** (1993) 283–318
7. Pentland, A.P.: Automatic extraction of deformable part models. IJCV **4** (1990) 107–126
8. Dickinson, S.J., Pentland, A.P., Rosenfeld, A.: 3-d shape recovery using distributed aspect matching. PAMI **14** (1992) 174–198
9. Pilu, M., Fisher, R.: Model-driven grouping and recognition of generic object parts from single images. In: ISIRS, Lisbon, Portugal (1996)
10. Liu, L., Sclaroff, S.: Deformable model-guided region split and merge of image regions. IVC **22** (2004) 343–354
11. Wang, J., Gu, E., Betke, M.: Mosaicshape: Stochastic region grouping with shape prior. In: CVPR. (2005)
12. Quach, T., Farooq, M.: Maximum likelihood track formation with the viterbi algorithm. In: CDC, Lake Buena Vista, FL (1994) 271 – 276
13. Yan, F., Christmas, W., Kittler, J.: A maximum a posteriori probability viterbi data association algorithm for ball tracking in sports video. In: ICPR, Hong Kong (2006) 279 – 282
14. Buchin, K., Knauer, C., Kriegel, K., Schulz, A., Seidel, R.: On the number of cycles in planar graphs. In: COCOON, LNCS **4598**, Springer (2007) 97–107
15. Tiernan, J.C.: An efficient search algorithm to find the elementary circuits of a graph. Commun. ACM **13** (1970) 722–726
16. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. CC **10** (1973) 112–122
17. Tax, D., Duin, R.: Data description in subspaces. In: ICPR **2**. (2000) 672–675
18. Forney, G.D.: The viterbi algorithm. Proceedings of the IEEE **61** (1973) 268–278
19. Nock, R., Nielsen, F.: Statistical region merging. PAMI **26** (2004) 1452–1458