Model-Based Perceptual Grouping and Shape Abstraction

Pablo Sala
University of Toronto

Sven J. Dickinson
University of Toronto

Abstract

Contour features are re-emerging in the categorization community as it moves from appearance back to shape. However, the classical assumption of one-to-one correspondence between an extracted image contour and a model contour constrains category models to be highly brittle, offering little abstraction between image and model. Moreover, today's contour-based models are category-specific, offering no mechanism for contour grouping and abstraction in the absence of an object prior. We present a novel framework for recovering a set of abstract parts from a multi-scale contour image. Given a user-specified part vocabulary and an image to be analyzed, the system covers the image with abstract part models drawn from the vocabulary. More importantly, correspondence between image contours and part contours is many-to-one, yielding a powerful shape abstraction mechanism. We illustrate the strengths and weaknesses of this work in progress on a set of anecdotal scenes.

1. Introduction

The object categorization community has seen a recent shift from local, appearance-based features (e.g., [6]) to contour-based features (e.g., [7]), reflecting a growing realization that shape is more generic to a category than appearance. This trend is not at all surprising, for the vast majority of early object recognition systems, in general, and object categorization systems, in particular, were based on contours (shape) for exactly the same reason. Still, there are some important differences between this new generation of contour-based approaches and earlier work, the most pronounced of which is the recent formulation of the problem of object categorization as primarily a detection (i.e., target recognition) problem. As opposed to a more general recognition problem, in which a priori knowledge of scene content (other than the assumption that scene objects are drawn from a large database) is absent, the detection problem tests the existence of an instance of a *particular* category in the image (perhaps at a particular location).

The effect of a strong model prior (in the form of a target model) precludes the need for sophisticated contour grouping, since the target model provides strong constraints on expected contour features and their relations. However, in a more general setting, domain-independent perceptual grouping rules are necessary to form the basis for a set of indexing features (i.e., parts) used to select a small set of promising candidate models present in the scene. This has long been the domain of classical perceptual grouping work in the computer vision community e.g., [10, 8, 1]. Unfortunately, a simple return to these classical techniques will not necessarily allow us to extend detection systems to general recognition systems; contours must not only be grouped, but the resulting contour groups must also be *abstracted* to form the generic shape parts that make up an object.

Figure 1 illustrates the challenge we face, where the image in (a) is processed to yield the contours in (b). Looking at the contour image, it's not difficult to see the emergent shapes of two coarse parts of the rice cooker: its top and its body. Yet while each part defines a subcollection of contours, the individual contours do not necessarily map one-to-one to the model contours that make up the two abstract parts. Only when the contours are grouped does their collective shape emerge as a part abstraction. However, when objects have texture or structural detail, a plethora of contours leads to an intractable grouping task, with most contours not contributing much to the coarse shape of the object. This bottleneck has, in fact, prevented the application of classical perceptual grouping techniques to images of textured objects.

In this paper, we report initial progress on this important problem. We begin by drawing on classical shape modeling techniques that assume that a large vocabulary of objects can be composed of parts drawn from a small vocabulary. However, diverging from classical techniques, we don't assume one-to-one correspondence between extracted image contour features and part model contour features. Instead, we draw on the ability of a detector at the *part* level, instead of at the *object* level, to guide contour abstraction, and introduce a novel indexing mechanism that generates abstract part hypotheses given a collection of contour fragments. The resulting framework takes, as input, a 2-D qualitative part vocabulary and an image, and covers the image

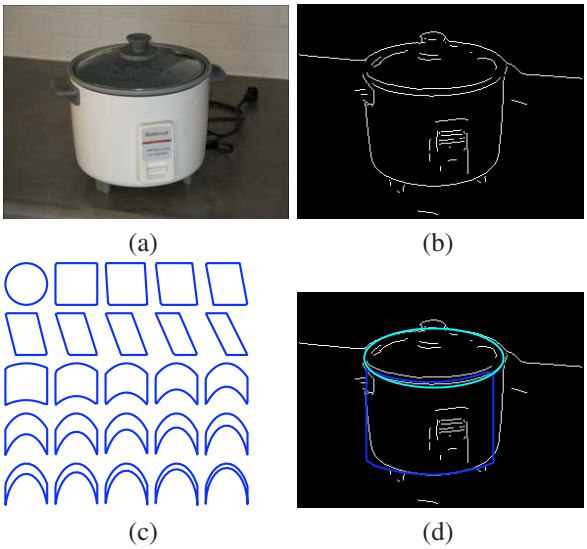


Figure 1. Recovering abstract shape parts from an image: (a) input image; (b) extracted contours; (c) a sample part vocabulary; and (d) the covering of parts (drawn from the vocabulary) produced by our system.

with a set of qualitative parts drawn from the vocabulary. Figure 1(c) shows an example input part vocabulary and the covering of the image (d) computed by our approach using parts drawn from the vocabulary. The abstract parts of the rice cooker reflect its coarse cylindrical structure.

2. Related Work

The problem of decomposing an object into a set of qualitative parts encompasses a vast array of related work. Due to space constraints, we will focus only on those approaches most closely related to our own, excluding approaches that: 1) operate on 3-D data; 2) assume figure/ground segmentation, i.e., take, as input, a single silhouette or region; and 3) assume knowledge of what object is in the scene.

A number of approaches have evolved to extract very generic shape regularities, such as compactness, elongatedness, or symmetry. For example, multi-scale blobs, including the work of Crowley [4], Lindeberg [9], and Shokoufandeh *et al*. [16] all employ ridge and/or blob models as symmetry-based mid-level part constraints. While such models offer excellent shape abstraction and have led to powerful hierarchical shape representations for recognition, they have not been successfully recovered from textured objects. Moreover, it's not clear whether they provide a rich enough shape description, for the parts cannot bend or taper.

In contrast to the above approaches, which are based on examining a dense set of filter responses at different positions, orientations, and scales, model-based 2-D part recovery can also be formulated as a fitting problem in which an algebraic model is fit to contour data. For example, the work of Rosin [15] proposed various methods to recover superellipses from contour data, while Osian *et al*. [12] considered the case for superellipses with affine deformations. Since the models were hypothesized from (i.e., fit to) explicit contour sections, scenes containing textured objects were avoided. More importantly, the bottom-up nature of the approaches prevented effective shape abstraction.

Working our way up the spectrum of prior shape knowledge, a number of approaches have assumed knowledge of a small part vocabulary without assuming any object-level knowledge. Pentland [13] proposed a method for decomposing a binary image into 2-D parts corresponding to the view-based projections of a vocabulary of volumetric primitives. Following a MDL principle, the problem was formulated as finding the simplest "covering" of the image in terms of the view-based "templates" that model the 3-D part vocabulary. A related approach was proposed by Dickinson *et al*. [5], in which part-based aspects (representing the possible views of a vocabulary of volumetric parts) were used to cover the contours in an image. Other approaches have attempted to decompose a contour image into a set of part-based views, including the work of Pilu and Fisher [14]. However, all of these approaches made the limiting assumption that image contours mapped one-to-one to model part contours. Little, if any, true abstraction was achieved, and the systems were rarely, if ever, applied to textured objects.

While the idea of a mid-level shape prior to group contour data has received a lot of attention, early approaches assumed an overly strong correspondence between extracted image contours and model contours. This meant that there was little variability between the abstract part model and the observed part, yielding very little shape abstraction and precluding the complex, textured objects addressed by today's categorization (detection) systems (which are based on object-level priors). The solution lies in drawing on the concept of a part-level (as opposed to object-level) shape prior, yet relaxing the highly restrictive one-to-one feature correspondence assumption. Only then can detailed image contour structure be abstracted into a set of qualitative parts suitable for indexing into a large database of categories.

3. Overview of the Approach

Our approach to qualitative, part-based shape abstraction takes, as input, a 2-D image and a vocabulary of 2-D part models. The input image is processed to yield a hierarchy of edge maps at different scales. The part models can be seen as closed contour templates, and are meant to coarsely sample the projections of the surfaces that comprise a vocabulary of qualitative 3-D parts. Our 2-D vocabulary must sample the variability of projected 2-D shape due to a 3-D part's rotation in depth as well as any within-class part deformations, such as bending, tapering, shearing, etc. Rotation and image scale of the part vocabulary do not need to

be sampled, for they are handled by rotation and scaling of the input image. For the experiments reported in this paper, our 2-D shape vocabulary consists of 25 part shapes (shown in Figure 1(c)), sampled from the family of superellipses with bending and shearing deformations.

Our approach simultaneously proceeds both top-down and bottom-up. In a classical top-down step, a fixed-size search window will be placed at all scales, locations, and rotations in an attempt to detect the presence of an abstract 2-D part drawn from the vocabulary. However, unlike today's object detectors, which are object-based, we introduce a bottom-up step, in which local edge evidence falling in the search window is used to index into the space of part templates consistent with the local evidence. In an even more critical departure from today's contour-based detectors, which assume that extracted image contours map one-to-one to model contours, we introduce a critical shape abstraction mechanism that allows a collection of disconnected contour fragments to map many-to-one to an abstract model contour. This is essential to support abstract part decomposition of images of real objects, in which the granularity of abstract model contours may lie far above the granularity of the extracted image contours.

When similar hypotheses compete for the same image evidence (in highly overlapping search windows), a non-maximum suppression step discards all but the model achieving the best fit. A final model selection step completes the procedure, in which from the set of redundant (and even contradicting) hypotheses, the subset achieving the *best* covering is selected. The result is a segmentation of an image into a set of 2-D part abstractions drawn from a vocabulary whose size is fixed and independent of the (possibly infinite) number of objects whose projections can be abstractly modeled as configurations of 2-D parts drawn from the vocabulary. Since different object domains may require different vocabularies of shape models, it is important to note that the vocabulary is an input; the method does not depend on a particular vocabulary.

4. Part Hypothesis Generation

Figure 2 presents our algorithm (pseudocode) for generating abstract part hypotheses from extracted image contours. The algorithm takes, as input, a multi-scale edge map H (computed from the input image) and a set V of qualitative 2-D shapes, and outputs a set R of model hypotheses supported by sufficient evidence from the input edge data. In an off-line step, we first compute the *model grid* G, a spatial data structure used to measure distances between observed image edgels and abstract model contours, facilitating the bottom-up indexing step that hypothesizes part models in a search window, and supporting the many-to-one mapping of contours essential for part abstraction.

Input: an edge map hierarchy $\{H_1, \ldots, H_L\}$, a qualitative part vocabulary V
Output: a set R containing the recovered models

Generate model grid G containing all models in V {(Section 4.1)}
Compute set $E_{l,\theta}$ of all rotations of H_l, $l = 1, \ldots, L$ {(Section 4.2)}
$R = \emptyset$
$s_x = \mathsf{min\_s}$ {(Section 4.3)}
while $s_x \leq \mathsf{max\_s}$ **do**
  $s_y = \mathsf{min\_s}$
  while $s_y \leq \mathsf{max\_s}$ **do**
    Compute the relevant level l of the edge map hierarchy
    Compute cell_size
    for $\theta = 0$ to 2π step $d\theta(s_x, s_y)$ **do**
      $e = \mathsf{resample}(E_{l,\theta}, \mathsf{cell\_size})$ {(Section 4.4)}
      $I = \mathsf{integrate}(e)$
      for all t_x, step pixels_per_translation_step **do**
        for all t_y, step pixels_per_translation_step **do**
          model_present = $\mathsf{screen}(I, t_x, t_y)$ {(Section 4.5)}
          if model_present **then**
            $M = \mathsf{index\_models}(e, t_x, t_y, G)$ {(Section 4.6)}
            $R = R \cup \{m \in M : \mathsf{percent\_covered(m)} \geq \tau\}$ {(Section 4.7)}
          end if
        end for
      end for
    end for
    $s_y = \mathsf{s\_ratio} * s_y$
  end while
  $s_x = \mathsf{s\_ratio} * s_x$
end while

Figure 2. Part Hypothesis Generation (components of the algorithm are detailed in the designated subsections)

The on-line procedure effectively places the model grid in the image at all positions, scales (in x and y), and orientations, and votes for models at each grid placement. A discrete set of rotations of the multi-scale edge map H is computed, allowing the orientation of the model grid to be fixed. A loop over all model sizes s_x and s_y follows, in which the relevant level l of the multi-scale edge map is selected as a function of model size. By anisotropically subsampling this level (l), we can effectively vary the length and width of the model grid while keeping its size fixed. This helps manage search complexity by ensuring that search windows never become large, i.e., a large window at a finer granularity can be approximated by a small window at a coarser granularity. The model grid cell size, cell_size, in terms of pixels at the original image resolution, is a function of both the level l as well as the degree of anisotropy.

For each model size, we iterate over all rotation angles θ, and for each angle, a resampled version e of the edge map at level l and orientation θ is computed. Before the area of support corresponding to each translated (by (t_x, t_y) pixels) model grid is used to generate bottom-up part hypotheses, it is first screened to see if enough edge activity lies in the window. If enough edge activity is found, by examining the integral image I in the window, the model grid G is overlaid on the resampled edge map e so that image edgels can vote for part models encoded in G. Finally, all model hypotheses having a minimum percentage τ of their contour accounted for by image evidence are added to the output set R. In the subsections below, we explore these steps in more detail.

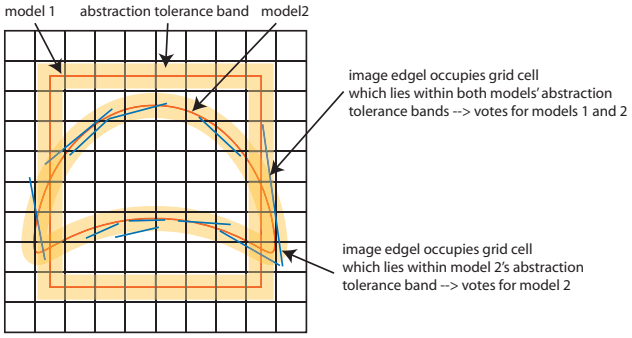


Figure 3. The model grid is a data structure that maps a grid cell to those models whose abstract contour, including a tolerance band around the contour, intersects the cell. An image edgel will vote for all models whose abstraction tolerance bands intersect the cell containing the edgel. Votes are weighted according to relative distance and orientation with respect to the nearest model contour fragment. Two models populate the model grid shown in this example.

4.1. The Model Grid

The model grid G has fixed size and resolution, and represents a unit square (i.e., $[-0.5, 0.5] \times [-0.5, 0.5]$) containing axis-aligned instances of all models in the part vocabulary, scaled to fit in the square. Each model is represented as a discrete set of roughly similar length, linear *model contour fragments*. Each model also defines a scale-independent *abstraction tolerance band* along its model contour fragments, such that any image edgel falling within this band is considered to be consistent with the model (note that the grid is extended in size to encompass the band). Scale independence of the band is achieved by making its size a function of the lengths of the model axes. Finally, each cell in the grid contains a list of model contour fragments belonging to any model whose abstraction tolerance band encompasses the center of the cell. Associated with each model contour fragment in this list is its normal orientation and the minimum Euclidean distance between the fragment and the center of the cell. Figure 3 illustrates the model grid and the abstraction tolerance bands for two models.

4.2. Rotation Invariance

Rotation invariance is achieved by fixing the model grid orientation and rotating the image. Given a finite set of angles, a rotation $E_{l,\theta}$ of each level l in the edge map hierarchy is computed (once, at initialization) for each angle θ in the set. If a given model is rotated through θ, a normalized alignment error can be computed as the ratio of the integral of the Euclidean distance between corresponding model points, before and after the rotation, to the model's perimeter. In general, this error increases as the shape's aspect ratio deviates from unity. Therefore, to guarantee that this (normalized) error remains bounded and independent of the model size and aspect ratio, the rotation step size is dependent on the model's aspect ratio.

Specifically, the rotation angle step can be computed as a function $d\theta(s_x, s_y)$ of a parameter ψ, which is the ratio of the maximum error between two corresponding model points to the length of the smallest axis. Formally, let $s_{\mathsf{max}} = max(s_x, s_y)$ and $s_{\mathsf{min}} = \min(s_x, s_y)$. ψ specifies the distance between the point $(s_{\mathsf{max}}, 0)$ and its position following rotation by an angle $d\theta$, normalized by s_{min}, i.e., $\psi = \|(\cos(d\theta) - 1, \sin(d\theta))\| \frac{s_{\mathsf{max}}}{s_{\mathsf{min}}}$. It follows that $d\theta = \arccos(1 - (\psi \frac{s_{\mathsf{min}}}{s_{\mathsf{max}}})^2/2)$.

4.3. Scale and Translation Invariance

We achieve scale invariance by fixing the model grid size and moving it through different scales of the image, with the ratio (s_ratio) between adjacent scales held constant. The number of scales that are generated is a function of the minimum and maximum model sizes (in terms of pixels at the original image resolution) min_s and max_s. Translation invariance is achieved by translating the model grid, where the translation step size is a function of the model size which, in turn, implies that it is constant for all scales. Edge maps are resampled so as to ensure that this constant translation step size is an integer number of pixels.

Specifically, let t be the fraction of the model size that corresponds to a translation step, and let a be the fraction of the model size that corresponds to the abstraction tolerance. If $\frac{t}{(1+2a)}$ is a rational number, then given a minimum dimension of K cells for the model grid, it is always possible to resample the edge maps such that: 1) both the model grid size (dimension) $\widetilde{K}$ and model translation step correspond to an integer number of pixels; and 2) $\widetilde{K} \geq K$. Let n and d be integers such that $\frac{n}{d} = \frac{t}{(1+2a)}$ and $gcd(n, d) = 1$. Then $\widetilde{K} = \lceil K/d \rceil \cdot d$ is the minimum value for the model grid size satisfying the above two constraints. The corresponding number of pixels per translation step is pixels_per_translation_step $= \lceil K/d \rceil \cdot n$.

4.4. Edge Map Resampling

To detect anisotropic scalings of a model, we anisotropically scale the isotropic multi-scale edge map. For each rotation angle in the set of angles corresponding to a particular model size, the corresponding rotated edge map at a certain level of the hierarchy is anisotropically resampled, with the level chosen based on the size (s_x, s_y) of the model. The level whose pixel size (in terms of input image pixels) is closest to $\min(\sigma_x, \sigma_y)$ is chosen, since that is the edge map whose resolution matches that of the smaller (finer granularity) dimension of the scaled model. In our experiments, we use a hierarchy of edge maps with as many levels as sampled model sizes.

Specifically, the function resample(η, σ) resamples an

edge map η for a model grid cell size $\sigma = (\sigma_x, \sigma_y)$. An edgel p in the resampled edge map is considered active if there is at least one active pixel in η that is resampled to p. The resampled location of a pixel (q_x, q_y) from η is computed as $(\lfloor q_x/\sigma_x \rfloor, \lfloor q_y/\sigma_y \rfloor)$. The σ used for resampling is computed as a function of model size, the ratio between the resolution of the original image and that of the edge map at the current level of the hierarchy, and the value of the translation step. Cell size is computed as $(\sigma_x, \sigma_y) = (dt_x, dt_y)/\mathsf{pixels\_per\_translation\_step}$, where the translation steps dt_x and dt_y are computed as $(dt_x, dt_y) = t \cdot (s'_x, s'_y)$. s'_x and s'_y are the model sizes s_x and s_y after rescaling them according to the ratio of the resolution of the edge map e to that of the original image.

4.5. Model Screening

The process of hypothesizing model parts contained in the search window is computationally expensive, and should only be applied in windows where a part may exist. Thus, a fast screening operation is performed on the contents of the search window to identify search windows which are highly unlikely to contain a part. This fast screening process is based on edge activity and how it is spatially distributed in the window. Since abstract part boundaries map to edgel evidence, and since part compactness implies a non-local distribution of this evidence, we check that there is a minimum amount of edge activity that is distributed in the four quadrants of the window.

The process can be efficiently computed using integral images (or summed-area tables [3]). The integral I of an edge map e contains at position (i, j) the sum of active pixels with height i and width j, i.e.,

$$I(i,j) = \sum_{i'<=i} \sum_{j'<=j} e(i', j'). \quad (1)$$

I can be computed recursively by the formula:

$$I(i,j) = \begin{cases} I(i,j-1) + I(i-1,j) + & i,j \geq 0 \\ e(i,j) - I(i-1,j-1), & \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The number of active pixels in a rectangular window (at position (i, j) with height and width (h, w)) in edge map e can be efficiently computed using four references to I as

$$I(i,j) + I(i+h, j+w) - I(i, j+w) - I(i+h, j). \quad (3)$$

Each quadrant can be computed in this manner. A count of the number of quadrants whose edge activity exceeds a threshold reflects the degree to which there is coherent edge activity distributed about the center.

4.6. Model Indexing

Hypothesizing models in a search window $w \subset e$ at a particular position (t_x, t_y) of the resampled edge map e is performed by a process equivalent to overlaying the model grid G on the window and having each active edge pixel $p \in w$ vote for (or index) those models in G having a model contour fragment whose abstraction tolerance band encompasses p (see Figure 3). A voting table T is used to keep track of the evidence supporting each model contour fragment. Specifically, entry (m, f) in T accumulates evidence in support of model contour fragment f belonging to model m according to the edgel activity falling within f's abstraction tolerance band. The voting process proceeds as follows. An edgel at position (i, j) in the search window lies within the abstraction tolerance bands of all model contour fragments in $G(i, j)$. For each of these candidate model contour fragments, their corresponding entry in T is updated according to the scoring function below. After all edgels in the window have been processed, those models (i.e., rows in T) with an insufficient number of model contour fragments receiving non-zero evidence are discarded.

4.7. Model Scoring

A model score is a representation of how well the image evidence accounts for a particular model at a certain position, orientation and scale. Abstraction is implemented in this calculation by taking into consideration all image edgels falling within the model's abstraction tolerance band. The contribution to the model score by each such image edgel depends on three intuitive components common to classical work in chamfer- or Hausdorff-based target recognition (e.g., [11, 2]):

1. **distance** As the distance between an edgel and the nearest model contour fragment increases, the edgel's contribution to the model score decreases.

2. **orientation** As the difference in orientation between the edgel's normal and that of the nearest model contour fragment increases, the edgel's contribution to the model score decreases.

3. **continuity** Edgels that form longer, more salient contours should contribute more to the model score than spurious, disconnected edgels that may be due to noise.

For continuity, rather than performing a full connected components analysis to determine the length of the contour that subsumes each edgel, we instead compute a local measure of continuity of all image edgel evidence in support of a model contour fragment. Specifically, let C_f be the set of image edgels within the abstraction tolerance band that are closer to f than to any other model contour fragment, let N_f be the total number of edgels in the edge map (at original

resolution) that were resampled to edgels in C_f, and let E_f be the number of these edgels that are endpoints. For model contour fragment f, the ratio $\frac{E_f}{N_f}$ reflects how disconnected the edgel support for the model contour fragment is.

Combining these measures yields the following equation to compute the score of a model contour fragment f:

$$S_f = \frac{E_f}{N_f} \frac{1}{|C_f|} \sum_{c \in C_f} g(d_c) h(\gamma_c), \tag{4}$$

where d_c is the distance from edgel c to the fragment f, γ_c is the difference in normal orientation between c and f, and g and h are symmetric weight functions achieving a maximum in 0, and decaying away from it; in our implementation g is a Gaussian and $h(x) = cos^2(x)$. The division by C_f makes the measure independent of the number of image edgels contributing to f. If C_f is empty, S_f is defined as 0.

Having defined the score for a given model contour fragment, we can now define the score of an entire model. In addition to summing the scores of its component fragments, we would like the overall score to reflect the spatial coherence of the local evidence. We therefore reward spatially coherent sets of consecutive model contour fragments supported by the image evidence. Specifically, we compute the final model score as the sum of *augmented fragment scores*, where the augmented score $\widetilde{S}_f$ of a fragment is 0 if $S_f = 0$; otherwise, it is the convolution with a Gaussian filter of the scores of the fragments in a neighborhood centered on f. Thus, the more fragments in f's neighborhood that are accounted for by image evidence, the larger $\widetilde{S}_f$ is. Formally, let $f_0, \ldots, f_{t-1}$ be the list of model contour fragments making up the model's contour, sorted by their position along the contour, and let S_i be the score of fragment f_i. The total score of the model is computed as:

$$S = \sum_{i=0}^{t} [S_i \neq 0] \sum_{r=-d}^{d} G(r) S_{(i+r) \mod t} \tag{5}$$

5. Part Hypothesis Selection

Part hypothesis generation yields both redundant and competing hypotheses, from which a final selection must be made. Our selection strategy begins with a local, non-maximum suppression step, focusing on redundant hypotheses, i.e., models with similar shape that account for the same image evidence. Specifically, only that model with the best score is kept whenever two or more models from the same shape class are detected in highly overlapping windows.

The second phase of our selection strategy is global, selecting the smallest subset of maximally-sized, highest-scoring part hypotheses that covers the object surfaces visible in the image. We adopt an optimization formulation in the spirit of Pentland [13], whose goal was to maximize the savings in the encoding length of describing a binary image with a subset of binary mask hypotheses. However, unlike [13], in which explicit region information was available, we have only boundary data (contours) to work with. We therefore introduce a hybrid strategy in which the boundary-based hypothesis score is used to weight the area of the hypothesis, favoring the selection of larger hypotheses, but penalizing them if they lack strong boundary support.

Interaction between pairs of hypotheses also needs to be treated differently. Whereas [13] measured both the agreement and disagreement in image region data where two hypotheses intersect, we have no explicit surface data to work with. We therefore compute the cost of encoding the overlap (area of intersection), representing the cost of encoding the fact that two surfaces cannot occupy the same region in the image. This clearly penalizes overlapping part hypotheses which, as we shall see in Section 6, is well-motivated (and effective) only in the absence of significant occlusion or self-occlusion. Finally, we introduce a part compatibility term that reflects the degree to which two part hypotheses fit together well, measured in terms of the area of intersection of their abstraction tolerance bands. While the resulting hybrid objective function is not technically a description length, we will adopt the term "savings" to refer to the benefit of selecting one or more hypotheses.

Let H be our set of part hypotheses that survive the non-maximum suppression (phase 1 selection). The savings in encoding the entire image (in terms of individual pixels) using part hypothesis h_i is:

$$\mathbf{S}(h_i) = k_1 a_{ii} - k_2 e_{ii} - k_3, \tag{6}$$

where a_{ii} is the part hypothesis' score (S_{h_i}) scaled by its area, and $e_{ii} = 0$. k_1, k_2, and k_3 can be interpreted as encoding costs for a pixel's area, for an incorrect pixel's area, and for the hypothesis, respectively.

The global solution is found by searching the power set of H to find the subset of part hypotheses that maximizes savings. If $\mathbf{x}$ is the binary vector whose length is the number of hypotheses, then a subset of hypotheses can be encoded by setting those elements in the subset (and clearing the others). In this case, our objective function becomes:

$$\mathbf{S}(\mathbf{x}) = k_1 \mathbf{A}\mathbf{x}^T - k_2 \mathbf{x}\mathbf{E}\mathbf{x}^T - k_3 \mathbf{x}\mathbf{x}^T \tag{7}$$

where $a_{ij} = a_{ji}, i \neq j$, is a function of the intersection of the abstraction tolerance bands of parts i and j, and $e_{ij} = e_{ji}, i \neq j$ is (half) the area of intersection of parts i and j. We seek the subset of part hypotheses that maximizes this savings, and employ a standard quadratic programming optimization procedure to solve the problem.

6. Demonstration

We demonstrate our work in progress on some anecdotal images to help illuminate both the strengths and weaknesses of our framework. Figure 4 illustrates the results of our framework applied to the six examples shown in the first row. Rows two through five contain the edge maps, the top 200 (scoring) hypotheses prior to model selection, the ground truth solution manually chosen from the generated hypotheses, and finally the solution as selected by our system. The ground truth, representing a subset of the generated hypotheses, clearly reflects the ability of our framework to generate model-based shape abstractions of noisy, disconnected contour data without assuming a one-to-one correspondence between extracted image contours and abstract model contours.

In (a), our search strategy converges on a plausible set of abstract part surfaces, including correct parts for the apple and can surfaces as well as four out of the five surfaces of the slot machine. In (b), the correct part abstraction has been fit to the top of the cup, the cup's body has been slightly oversegmented, and an entire surface has been recovered for the silhouette of the cup's handle. This larger hypothesis is preferred over the two handle hypotheses shown in the ground truth, where the handle has been oversegmented due to the fact that it cannot be completely covered by a single part from the vocabulary.

In (c), the two abstract surfaces of the jar are recovered, but slight problems exist on the oatmeal bag. One of the bag's surfaces has been oversegmented, while the end of the bag has been fit with an ellipitcal part instead of a parallelogram. This is understandable, for the ground truth clearly reflects the human selector's bias toward the block-like regularization despite the elliptical surface evidence in the original image. Both these types of errors (oversegmentation and misidentification) can be easily corrected when the identities of nearby hypotheses are taken into account. Recall that other than generic boundary overlap, explicit part interactions are not modeled.

Clearly, our objective function needs further improvement to strike an optimal balance between hypothesis score, overlap, size, compatibility, and cardinality. This is a challenging task whose goals can be in conflict. For example, in (d-e), the overlap penalty has prevented us from recovering the cup's body and the hat's body, respectively, instead favoring the larger saucer and brim hypotheses. In such cases, we need to better reason about occlusion, and not penalize the overlap when it can be explained by occlusion. In (f), we see that some of the ground truth parts (desk objects) have been undersegmented while others oversegmented. Still, it is important to note that only a small set of correctly recovered parts may be necessary to invoke stronger top-down models with which to disambiguate competing hypotheses and to guide hypothesis selection; perfect, bottom-up part segmentation is not a realizable goal.

We are currently exploring a number of selection strategies and optimization frameworks to bring us closer to the ground truth. In addition, we are constructing a much larger ground truth dataset with which to perform a more comprehensive evaluation of our framework under different conditions, including occlusion, self-occlusion, clutter, extraneous contours (structural noise), and texture. While the hypothesis generation represents a more mature component in our system, demonstrating promising model-based perceptual grouping and shape abstraction, the hypothesis selection is clearly work in progress.

7. Conclusions

Unexpected object recognition requires the recovery of generic parts and their relations to support effective indexing into large databases. While contours may reflect important shape information, a single image contour or fragment may not be generic to a category, and assuming one-to-one correspondence with a model contour can be highly restrictive. However, a collection of local contours may reflect a more abstract regularity that may be shared by many categories. Such abstract parts require not only that a noisy, broken collection is grouped, but also abstracted.

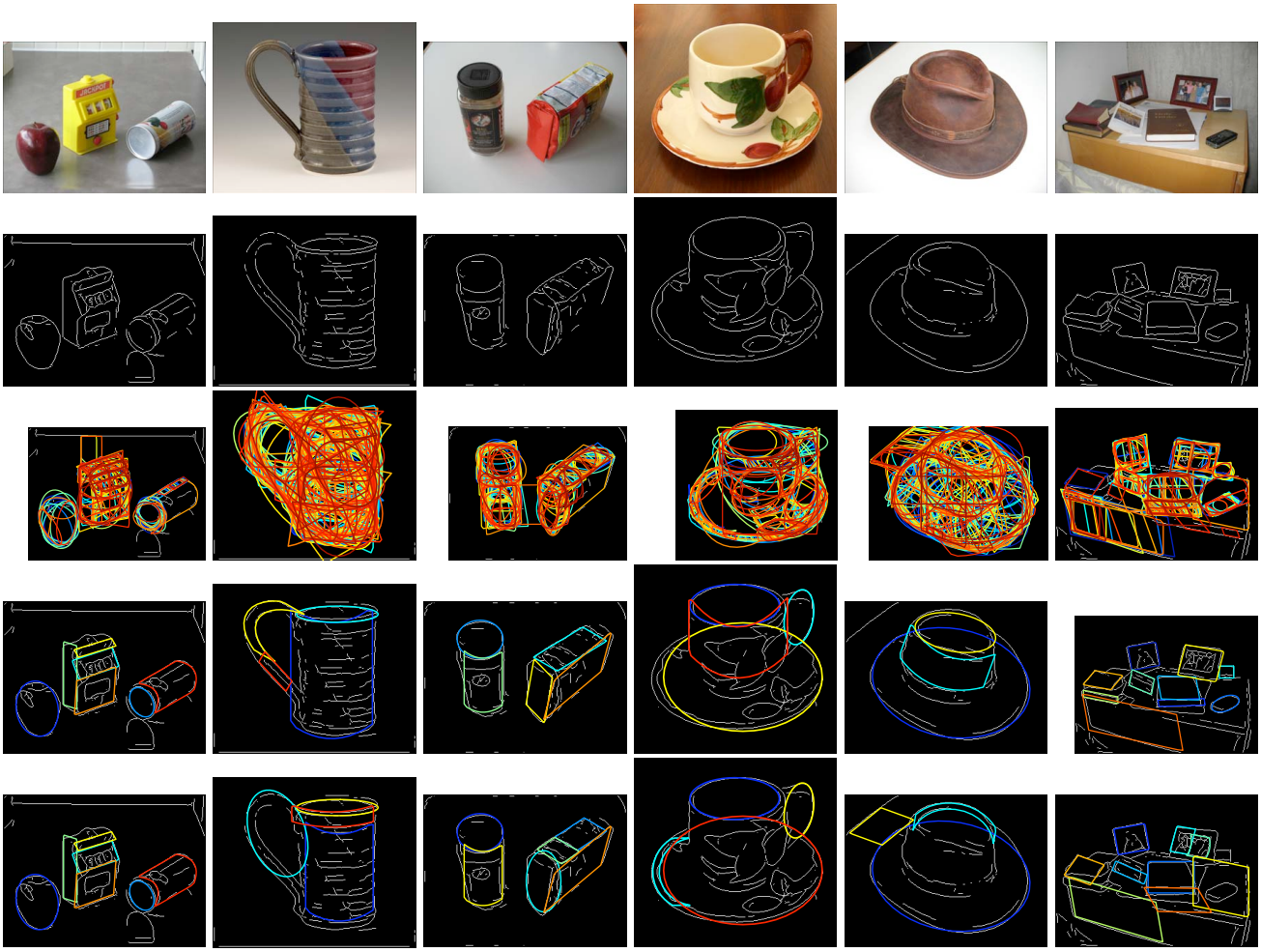
We have described a model-based framework for such grouping/abstraction that combines a mid-level shape prior in the form of a small (arbitrary) input vocabulary of part models (independent of the number of objects that can be constructed from the vocabulary) with a bottom-up part indexing framework that maps contour collections to abstract part models. Our preliminary results are promising and indicate the potential to recover abstract part structure from images of real objects. If a few correct parts can be identified, they may be sufficient to form a powerful index into a collection of object candidates which, in turn, can be used in a top-down manner to guide part selection.

The selection model is very simple and serves only to reflect the availability of good hypotheses. Our future work will focus on two major issues: 1) strengthening the role of part relations in the selection process, including both generic relations (e.g., junction information) as well as specific relations (e.g., hypothesis co-occurrence); and 2) the inclusion of 3-D constraints, (e.g., [17]), with the ultimate goal of recovering a set of abstract volumetric parts from an image.

References

[1] K. Boyer and S. Sarkar. Perceptual organization in computer vision: status, challenges, and potential. *CVIU*, 76(1), 1999.

[2] Y. Boykov and D. P. Huttenlocher. A new bayesian framework for object recognition. In *CVPR*, pages 2517–2523, 1999.



(a) (b) (c) (d) (e) (f)

Figure 4. Abstract Part Recovery (see text for discussion)

[3] F. Crow. Summed-area tables for texture mapping. In *ACM Siggraph*, pages 207–212, 1984.

[4] J. Crowley and A. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE PAMI*, 6(2):156–169, March 1984.

[5] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-d shape recovery using distributed aspect matching. *IEEE PAMI*, 14(2):174–198, 1992.

[6] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[7] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007.

[8] D. W. Jacobs. Robust and efficient detection of salient convex groups. *IEEE PAMI*, 18(1):23–37, 1996.

[9] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *IJCV*, 11:283–318, 1993.

[10] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *AI*, 31:355–395, 1987.

[11] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.

[12] M. Osian, T. Tuytelaars, and L. V. Gool. Fitting superellipses to incomplete contours. In *CVPRW*, page 49, Washington, DC, USA, 2004. IEEE Computer Society.

[13] A. P. Pentland. Automatic extraction of deformable part models. *IJCV*, 4(2):107–126, 1990.

[14] M. Pilu and R. Fisher. Recovery of generic solid parts by parametrically deformable aspects. In *ECCV*, Cambridge, England, April 1996.

[15] P. L. Rosin. Fitting superellipses. *IEEE PAMI*, 22(7):726–732, 2000.

[16] A. Shokoufandeh, I. Marsic, and S. Dickinson. View-based object recognition using saliency maps. *IVC*, 17(5-6):445–460, 1999.

[17] F. Ulupinar and R. Nevatia. Perception of 3-D surfa¡ces from 2-D contours. *IEEE PAMI*, 15:3–18, 1993.