Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Gradient Networks: Explicit Shape Matching Without Extracting Edges

Edward Hsiao and **Martial Hebert**
Robotics Institute, Carnegie Mellon University, USA
{*ehsiao,hebert*}*@cs.cmu.edu*

Abstract

We present a novel framework for shape-based template matching in images. While previous approaches required brittle contour extraction, considered only local information, or used coarse statistics, we propose to match the shape explicitly on low-level gradients by formulating the problem as traversing paths in a *gradient network*. We evaluate our algorithm on a challenging dataset of objects in cluttered environments and demonstrate significant improvement over state-of-the-art methods for shape matching and object detection.

Figure 1: Example of shape matching under heavy occlusion. (left) Image window, (center) normalized gradient magnitudes, and (right) probability that each pixel matches the shape of the pitcher.

Introduction

Recognizing specific object instances in images of natural scenes is crucial for many applications ranging from robotic manipulation to visual image search and augmented reality. While distinctive point-based features such as SIFT (Lowe 2004) have been shown to work well for recognizing texture-rich objects (e.g., books and paintings) even under severe occlusions, these methods fail when presented with objects that have large uniform regions. These texture-less objects (e.g., pitcher in Figure 1) are primarily defined by their contour structure, which are often just simple collections of curves and junctions. Even though many shape matching approaches work well when objects are un-occluded, their performance decrease rapidly in natural scenes where occlusions are common. This sensitivity to occlusions arises because these methods are either heavily dependent on repeatable contour extraction or only consider information very locally. The main contribution of this paper is to increase the robustness of shape matching under occlusions by formulating it as traversing paths in a low-level *gradient network*.

In the past, significant research has been dedicated to representing and matching shape for object detection. A common representation is to use lines (Ferrari, Tuytelaars, and Van Gool 2006) and contour fragments (Shotton, Blake, and Cipolla 2008). In the simplest form, contours are represented by a set of points and Chamfer matching (Barrow et al. 1977) is used to find locations that align well in an edgemap. Local edge orientation is often incorporated (Shotton, Blake, and Cipolla 2008) in the matching cost to increase robustness to clutter. These methods, however, consider each point independently and do not use edge connectivity.

To incorporate connectivity, some methods enforce the constraint that matched points are close together (Thayananthan et al. 2003), but this still does not ensure that the matches belong to the same image contour. Other approaches capture connectivity by approximating curves as sequences of line segments or splines (Zhao and Chen 1997) instead of points. A common issue with these approaches, however, is the difficulty of breaking contours at repeatable locations due to noise in the edgemaps and object occlusions. To address this issue, many-to-one contour matching (Srinivasan, Zhu, and Shi 2010) pieces together image contours to match the object shape using Shape Context (Belongie, Malik, and Puzicha 2002) features. The Contour Segment Network (Ferrari, Tuytelaars, and Van Gool 2006) method finds paths that match the shape through a network of extracted line segments. A major limitation of these approaches is their reliance on stable edge detection, which still remains an open area of research (Arbeláez et al. 2011).

To bypass edge extraction, some methods represent the shape by using coarse gradient statistics. Histogram of Oriented Gradients (HOG) (Dalal 2006) bins gradient magnitudes into nine orientation bins. These methods, however, only provide a coarse match of shape, losing many fine-grained details needed for instance detection. For example, a HOG cell with a single line and a HOG cell with multiple parallel lines have exactly the same descriptor. In addition, HOG cells on the object boundary are easily corrupted by strong background gradients and by object occlusions.

To capture the shape more explicitly without extracting edges, the LINE2D (Hinterstoisser et al. 2012) method scans

a template of sparse edge points across a gradient map. The rLINE2D (Hsiao and Hebert 2012) method increases the robustness of LINE2D by only considering points where the quantized edge orientation matches exactly. These approaches, however, do not account for edge connectivity, resulting in high-scoring false positives in cluttered regions.

In a parallel line of research, there has been work on classifying edges which belong to a specific object category. The Boosted Edge Learning (BEL) detector (Dollar, Tu, and Belongie 2006) extends the Probabilistic Boosting Tree (Tu 2005) algorithm to classify whether each location in the image belongs to the edge of the object. To speed up the classification, some approaches train local classifiers only at Canny edge points (Prasad et al. 2006). Sparse coding (Mairal et al. 2008) has also been used to learn class-specific edges. However in all of these cases, the classification is done independently at each location, effectively losing connectivity and the global shape. They also require a large amount of labeled training data and for the background in the test images to be very similar to the training set.

In this paper, we propose a shape matching approach which captures contour connectivity directly on low-level image gradients. For each image pixel, our algorithm estimates the probability (Figure 1) that it matches a template shape. The problem is formulated as traversing paths in a *gradient network* and is inspired by the edge extraction method of GradientShop (Bhat et al. 2010). Our results show significant improvement in shape matching and object detection on a difficult dataset of texture-less objects in natural scenes with severe clutter and occlusions.

Gradient Networks

In this section, we describe our algorithm for explicit shape matching using low-level gradients. For a template shape placed at a particular image location, our method returns for each image pixel, the probability that it matches the template. We begin by defining the *gradient network* of an image. Then, we formulate shape matching as finding paths in the network which have high local shape similarity. We describe the local shape potential for each node in the network, followed by the algorithm used for shape matching.

Formulation

For each pixel p in an image, let $\nu(p)$ be the gradient magnitude and $\theta(p)$ be the gradient orientation computed using oriented filters (Freeman and Adelson 1991). Let Q_0^p be the set of four pixels at integer coordinates closest to the floating point coordinate calculated by translating the pixel p a distance of $\sqrt{2}$ in the direction of the tangent $\theta(p) + \pi/2$. Similarly, let Q_1^p be the set of four pixels in the direction of the tangent $\theta(p) - \pi/2$. A *gradient network* is then defined as a graph where each pixel p in the image is a node that is connected to the eight pixels $q \in \{Q_0^p, Q_1^p\}$ as shown in Figure 2. We define $\phi_\beta(p, q)$ to be the bilinear interpolation weight for each q with respect to its ideal floating point coordinate.

In addition, let $\mathbb{S}(\mathbf{z})$ be a template shape $\mathbb{S}$ placed at position $\mathbf{z}$ in the image. Initially, for simplicity of explanation, we define $\mathbb{S}(\mathbf{z})$ by N edge points $\mathcal{Y} = \{y_1, ..., y_N\}$, each with a gradient orientation ψ_i. Later, we extend the formulation to directly operate on model edge strengths. For conciseness of notation, superscript $\mathbb{S}$ in the following derivation is implicitly $\mathbb{S}(\mathbf{z})$. Let $d^{\mathbb{S}}(p)$ be the distance from p to the nearest model edge point $y^* \in \mathcal{Y}$ and $\theta^{\mathbb{S}}(p) = \psi^*$ be the orientation of that edge point. Both values can be computed simultaneously using a Distance Transform (Breu et al. 1995). The goal is then to find long connected paths in the gradient network which match the template $\mathbb{S}(\mathbf{z})$ well.

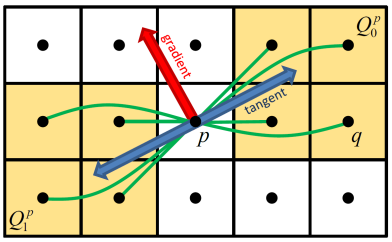

Figure 2: Gradient network. Each node is a pixel in the image. We create a network (green) by connecting each pixel p with its 8 neighbors in the direction of the local tangent.

Local Shape Potential

We begin by defining the local shape potential, $\Phi^{\mathbb{S}}(p)$, which measures how well each node p in the gradient network matches $\mathbb{S}(\mathbf{z})$ locally. This potential is composed of three terms: 1) the region of influence $\phi^{\mathbb{S}}_{roi}$, 2) the local appearance $\phi^{\mathbb{S}}_{\mathcal{A}}$, and 3) the edge potential ϕ_E. It is given by:

$$\Phi^{\mathbb{S}}(p) = \phi^{\mathbb{S}}_{roi}(p) \cdot \phi^{\mathbb{S}}_{\mathcal{A}}(p) \cdot \phi_E(p). \tag{1}$$

Region of Influence Given $\mathbb{S}(\mathbf{z})$, we only want to consider pixels which are sufficiently close as candidates for matching while simultaneously allowing slight deformations of the template (Bai et al. 2009). We employ a linear weighting scheme to define the region of influence as:

$$\phi^{\mathbb{S}}_{roi}(p) = \max\left[1 - \frac{d^{\mathbb{S}}(p)}{\tau_d}, 0\right], \tag{2}$$

where τ_d is the farthest distance from the shape that we want to consider. We set $\tau_d = 15$ to be the same as in Oriented Chamfer Matching (Shotton, Blake, and Cipolla 2008).

Local Appearance This term describes how well each pixel matches the local appearance of $\mathbb{S}(\mathbf{z})$. Many types of information can be used, ranging from local gradient orientation to interior appearance of the object, such as color and texture. For illustration of our approach, we consider the effects of gradient orientation and color. The local appearance potential is then defined as:

$$\phi^{\mathbb{S}}_{\mathcal{A}}(p) = \phi^{\mathbb{S}}_{\theta}(p) \cdot \phi^{\mathbb{S}}_{\mathcal{C}}(p), \tag{3}$$

where $\phi^{\mathbb{S}}_{\theta}$ is the orientation potential and $\phi^{\mathbb{S}}_{\mathcal{C}}$ is the color potential. We define the local orientation potential as:

$$\phi^{\mathbb{S}}_{\theta}(p) = \exp\left(-\frac{\left[\theta(p) - \theta^{\mathbb{S}}(p)\right]^2}{2\sigma_\theta^2}\right), \tag{4}$$

(a) template (b) window (c) $\Phi^{\mathbb{S}}$ (d) $\Omega^{\mathbb{S}}$

(e) $\phi^{\mathbb{S}}_{roi}$ (f) $\phi^{\mathbb{S}}_{\theta}$ (g) $\phi^{\mathbb{S}}_{\mathcal{C}}$ (h) ϕ_E

Figure 3: Illustration of algorithm. Given (a) the template and (b) the image window, we compute (c) the local shape potential and apply the message passing algorithm to produce (d) the shape similarity. The local shape potential is composed of the (e) region of interest, (f) orientation, (g) color, and (h) edge potentials.

with $\sigma_\theta = \pi/8$ (i.e., the orientation bin size of LINE2D (Hinterstoisser et al. 2012)).

Many methods can be used to incorporate color information around an edge. We describe a simple approach to show its efficacy. Unlike BEL (Dollar, Tu, and Belongie 2006) and (Prasad et al. 2006) which consider local patches centered on an edge, we only use information from the object interior to be more robust to background clutter.

Let v_i be the unit-norm gradient vector pointing to the object interior and c_i be the L*u*v color of the object extracted a fixed distance in the direction of v_i for each model edge point y_i. Then let $v^{\mathbb{S}}(p) = v^*$ and $\mathcal{C}^{\mathbb{S}}(p) = c^*$ correspond to the $y^* \in \mathcal{Y}$ closest to p. From the image, we extract the color $\mathcal{C}(p)$ at the same fixed distance from p in the direction of $v^{\mathbb{S}}(p)$. This corresponds to what the object interior would look like from this pixel if it is part of the shape. The local color potential is then defined as:

$$\phi^{\mathbb{S}}_{\mathcal{C}}(p) = \exp\left(-\frac{\left[\mathcal{C}(p) - \mathcal{C}^{\mathbb{S}}(p)\right]^2}{2\sigma^2_{\mathcal{C}}}\right). \quad (5)$$

We set $\sigma^2_{\mathcal{C}} = 1/15$ according to (Luo and Guo 2003) for L*u*v color normalization.

Edge Potential The edge potential, ϕ_E, characterizes how likely a pixel belongs to an edge in the image. Many different metrics can be used. In the simplest form, the edge potential can be the raw gradient magnitude $\nu(p)$. In GradientShop, the authors normalize the magnitude of each pixel with respect to the magnitudes in a 5×5 neighborhood z to be more robust to edge contrast. If μ_z and σ_z are the mean and standard deviation of the magnitudes in z, then the normalized gradient magnitude is:

$$\hat{\nu}(p) = \frac{\nu(p) - \mu_z}{\sigma_z + \epsilon}. \quad (6)$$

More complicated edge potentials, such as the output of edge detectors like the Global Probability of Boundary (gPb) (Arbeláez et al. 2009), can also be used instead. In the evaluation, we explore the effect of different edge potentials.

Shape Matching

While the local shape potential can be used as a measure of shape similarity, it considers only a very limited scope when determining how well each pixel matches $\mathbb{S}(\mathbf{z})$. By itself, it is prone to incorrect similarities from accidental alignments in background clutter and occlusions (Figure 3c). Our key idea for obtaining a more robust shape similarity is to broaden the scope of each pixel, p, by traversing the path in the gradient network on which it is centered. The pixel matches the shape well if this path consists of a long contiguous set of pixels which all have high local shape potential.

We characterize the contiguity from pixel p to each $q \in \{Q^p_0, Q^p_1\}$ by the pairwise potential:

$$\Psi^{\mathbb{S}}(p,q) = \phi_\beta(p,q) \cdot \phi_\theta(p,q) \cdot \phi^{\mathbb{S}}_{\mathcal{A}}(q), \quad (7)$$

where $\phi_\beta(p,q)$ is the bilinear interpolation weight and $\phi_\theta(p,q) = \exp\left(-\frac{[\theta(p)-\theta(q)]^2}{2(\pi/5)^2}\right)$ is the edge smoothness (Bhat et al. 2010). The local appearance potential, $\phi^{\mathbb{S}}_{\mathcal{A}}(q)$, effectively breaks the contiguity when the shape of the neighbor q is improbable. We do not include the region of influence potential as we do not wish to overly penalize an imperfectly aligned template.

This formulation of shape matching is related to the edge extraction approach of GradientShop (Bhat et al. 2010). We adapt their message passing technique to estimate the shape similarity. The problem of estimating the shape similarity at p is broken into two subproblems; one for estimating the similarity in the direction of Q^p_0 and the other for estimating the similarity in the direction of Q^p_1. At each iteration t, the messages are computed as:

$$m^{\mathbb{S},t}_0(p) = \sum_{q \in Q^p_0} \Psi^{\mathbb{S}}(p,q) \cdot \left[\Phi^{\mathbb{S}}(q) + m^{\mathbb{S},t-1}_0(q)\right], \quad (8)$$

$$m^{\mathbb{S},t}_1(p) = \sum_{q \in Q^p_1} \Psi^{\mathbb{S}}(p,q) \cdot \left[\Phi^{\mathbb{S}}(q) + m^{\mathbb{S},t-1}_1(q)\right], \quad (9)$$

and the estimated shape similarity is:

$$\Omega^{\mathbb{S},t}(p) = m^{\mathbb{S},t}_0(p) + m^{\mathbb{S},t}_1(p) + \Phi^{\mathbb{S}}(p). \quad (10)$$

The messages are initialized to $m^{\mathbb{S},0}_0(p) = m^{\mathbb{S},0}_1(p) = 0$, and the message passing is iterated for a fixed number of iterations to produce the final shape similarity estimate $\Omega^{\mathbb{S}}$. Empirically, the message passing converges in 25 iterations and we use this for all of our experiments.

Probability Calibration

The shape similarity $\Omega^{\mathbb{S}}$, computed in Equation 10, depends on the template shape $\mathbb{S}$. This makes it difficult to compare the similarity values of different templates, and thus difficult to choose the highest scoring template for object recognition. A method to calibrate these values is thus needed.

We use the Extreme Value Theory (Scheirer et al. 2012) to calibrate the shape similarity since it only requires the distribution of similarity values on negative data. Unlike category recognition where positive data can easily be mined from the Internet, it is much more difficult to obtain many images

of the same object instance under the same viewpoint. Negative data, on the other hand, is easy to obtain. We sample random locations in background images and use all the $\Omega^{\mathbb{S}}$ within the region of influence as negative data.

Soft Shape Model

The above formulation defines the template $\mathbb{S}(\mathbf{z})$ as a discrete set of edge points. This discrete representation requires either a manually specified template or automatic edge extraction. Manual specification, however, is impractical for a large set of templates, and automatic edge extraction requires time consuming parameter tuning to obtain good edgemaps. We address this limitation by computing a soft shape model using the raw edge strength (e.g., edge potential) of every object pixel (i.e., object mask), instead of discrete edge points. In the following, we define a soft way to compute the distance $d^{\mathbb{S}}$ and orientation $\theta^{\mathbb{S}}$ which fully describe the relationship between $\mathbb{S}(\mathbf{z})$ and the image.

Let $\mathcal{Y} = \{y_1, ..., y_N\}$ be all the pixels representing $\mathbb{S}(\mathbf{z})$, each with an edge strength γ_i and gradient orientation ψ_i. We define the soft distance $d^{\mathbb{S}}$ as:

$$d^{\mathbb{S}}(p) = \min_i \left[D(p, y_i) + 1/\gamma_i - 1/\gamma^{max} \right], \qquad (11)$$

where $D(p, y_i)$ is the Euclidean distance between p and y_i, and γ^{max} is the maximum edge strength. Then, the soft orientation is $\theta^{\mathbb{S}}(p) = \psi^*$ and corresponds to the $y^* \in \mathcal{Y}$ that minimizes Equation 11. Both values can be computed simultaneously using the Generalized Distance Transform (Felzenszwalb and Huttenlocher 2012). If γ_i is binary, then the soft shape model reduces to the discrete case.

Evaluation

In order to validate our method's performance in shape-based object instance detection, we performed two sets of experiments. The first evaluates the algorithm's accuracy in shape matching, while the second evaluates the algorithm's ability to detect objects. We compare the effects of using a hard model versus a soft model, as well as the effects of different edge and local appearance potentials.

Dataset

We evaluate our algorithm on the challenging CMU Kitchen Occlusion (CMU_KO8) dataset (Hsiao and Hebert 2012). Unlike other object instance detection datasets (Sun et al. 2010; Lai et al. 2011), CMU_KO8 contains objects in more realistic household scenes with both severe clutter and occlusions. The dataset contains 1600 images of 8 texture-less household objects under single and multiple viewpoints with groundtruth occlusion labels.

Algorithms

We compare our approach with a number of state-of-the-art methods for template-based shape matching. For fair comparison, we use the same M sampled model edge points x_i for all the methods. These points are specified relative to the template center. We give a brief description of the algorithms in our comparison below.

Gradient Network (GN) Our algorithm returns a shape similarity $\Omega^{\mathbb{S}}$ for each pixel given the template $S(\mathbf{z})$. For fair comparison, we apply a 7×7 max spatial filter (i.e., equivalent to LINE2D) to $\Omega^{\mathbb{S}}$ resulting in $\hat{\Omega}^{\mathbb{S}}$. The template score at $\mathbf{z}$ is then $\sum_{i=1}^{M} \hat{\Omega}^{\mathbb{S}}(x_i + \mathbf{z})$. We use the soft shape model with normalized gradient magnitudes for the edge potential, and both color and orientation for the appearance.

Our algorithm takes on average 2 ms per location $\mathbf{z}$ on a 3GHz Core2 Duo CPU. In practice, we run our algorithm only at the hypothesis detections of rLINE2D (Hsiao and Hebert 2012), which has been shown to have high recall. The combined computation time is about 1 second per image.

LINE2D (L2D) (Hinterstoisser et al. 2012) This method quantizes all gradient orientations into 8 orientation bins. The similarity for point x_i is the cosine of the smallest quantized orientation difference, $\Delta\theta_i$, between its orientation and the image orientations in a 7×7 neighborhood of $x_i + \mathbf{z}$. The score of a window is $\sum_{i=1}^{M} \cos(\Delta\theta_i)$.

rLINE2D (rL2D) (Hsiao and Hebert 2012) This method binarizes LINE2D by only considering model edge points which have the same quantized orientation as the image. The algorithm is more robust than LINE2D in cluttered scenes with severe occlusions. The score of a window is $\sum_{i=1}^{M} \delta(\Delta\theta_i = 0)$ where $\delta(z) = 1$ if z is true.

Oriented Chamfer Matching (OCM) (Shotton, Blake, and Cipolla 2008) This method extends Chamfer matching to include the cost of the orientation dissimilarity. Let DT be the distance transform of the image edgemap and DT_θ be the orientation of the nearest edge point, then the OCM score at position $\mathbf{z}$ is $\sum_{i=1}^{M} DT(x_i + \mathbf{z}) + \lambda \sum_{i=1}^{M} D_\theta \left[DT_\theta(x_i + \mathbf{z}), \psi_i \right]$. The parameter λ is learned for each shape independently.

Histogram of Oriented Gradients (HOG) (Dalal 2006; Malisiewicz and Efros 2011) This method represents an object as a grid of gradient histograms. An Exemplar SVM (Malisiewicz and Efros 2011) is learned for each shape. We use the one hundred negative images in CMU_KO8, the same parameters as (Malisiewicz and Efros 2011), and three hard negative mining iterations for training. The object is detected by convolving the learned template with the HOG of the image.

rL2D-gPb and GN-gPb To explore the use of more complex edge potentials, we extend rL2D and GN to use the output of gPb (Arbeláez et al. 2009), a state-of-the-art edge detector that uses texture and color segmentations. gPb outputs the probability B that a pixel belongs to an object boundary. The **rL2D-gPb** algorithm applies a 7×7 max spatial filter to B to produce $\hat{B}$ and computes the score at position $\mathbf{z}$ as $\sum_{i=1}^{M} \hat{B}(x_i + \mathbf{z}) \cdot \delta(\Delta\theta_i = 0)$. The **GN-gPb** algorithm uses $\phi_E = B$ as the edge potential.

Shape Matching

We first evaluate the performance in matching accuracy. Each algorithm, besides HOG, returns a similarity measure

	combined	orientation	color
naive (label all visible)	0.78	-	-
L2D	0.83	-	-
rL2D	0.83	-	-
rL2D-gPb	0.79	-	-
OCM	0.79	-	-
GN	0.87	0.85	0.83
GN-hard	0.86	0.85	0.83
GN-gPb	0.85	0.84	0.84

Table 1: F-measure characterizing the shape matching on CMU_KO8. For methods which use GN, we evaluate the effects of using orientation, color, and their combination for the local appearance potential, $\phi^{\mathbb{S}}_{\mathcal{A}}$. We also compare soft shape models (GN) with hard shape models (GN-hard).

Single	L2D	rL2D	rL2D-gPb	OCM	HOG	GN	GN-gPb
baking pan	0.46	0.68	0.41	0.66	0.69	**0.89**	0.86
colander	0.58	0.87	**1.00**	0.74	0.85	0.92	0.97
cup	0.45	0.80	0.93	0.71	0.86	**0.98**	0.96
pitcher	0.45	0.84	0.67	0.76	0.77	0.85	**0.89**
saucepan	0.49	0.82	0.71	0.70	0.69	0.99	**1.00**
scissors	0.29	0.62	0.27	0.53	0.75	**0.87**	0.86
shaker	0.29	0.68	0.91	0.49	0.72	0.84	**0.93**
thermos	0.57	0.80	0.50	0.71	0.80	**0.94**	0.94
Mean	0.45	0.76	0.68	0.66	0.77	0.91	**0.93**
Multiple	L2D	rL2D	rL2D-gPb	OCM	HOG	GN	GN-gPb
baking pan	0.32	0.41	0.19	0.45	0.65	**0.97**	0.90
colander	0.53	0.81	**0.95**	0.31	0.82	0.93	0.94
cup	0.34	0.67	0.78	0.42	0.90	**0.97**	0.97
pitcher	0.43	0.65	0.11	0.28	0.68	**0.86**	0.83
saucepan	0.41	0.76	0.64	0.59	0.82	**0.99**	0.98
scissors	0.37	0.60	0.07	0.17	0.64	**0.93**	0.80
shaker	0.34	0.61	0.50	0.18	0.59	0.84	**0.89**
thermos	0.38	0.75	0.40	0.36	0.85	0.93	**0.95**
Mean	0.39	0.66	0.45	0.35	0.74	**0.93**	0.91

Table 2: Detection rate at 1.0 FPPI on CMU_KO8.

per model point. Ideally for an image window, points corresponding to visible object parts should have higher similarity than those that are occluded. Given the groundtruth occlusion labels for every image, we partition the similarity scores into visible and occluded scores, and report the F-measure (i.e., maximum geometric mean of precision and recall) in Table 1. We do not include HOG in this evaluation because it does not return point confidences, and thus cannot be compared fairly with the other methods. Figure 4 shows some qualitative results.

From the table, GN outperforms all the baseline algorithms. L2D, rL2D and OCM consider information very locally resulting in many incorrect point confidences. rL2D-gPb removes spurious texture responses by using gPb, but performs poorly because its similarity measure is not indicative of how well the shape matches (e.g., high contrast edges have high gPb probabilities irrespective of shape). By considering long connected paths in gPb which match the shape, GN-gPb performs significantly better than rL2D-gPb. However, it performs slightly worse than GN, because gPb often gives low probability to interior object edges, resulting in incorrect confidences in these areas. The table also shows, importantly, that both the orientation and color appearance potentials at edge points are informative for shape matching.

In addition, we evaluate the effect of using a soft shape model (GN) versus a hard shape model (GN-hard). We tuned

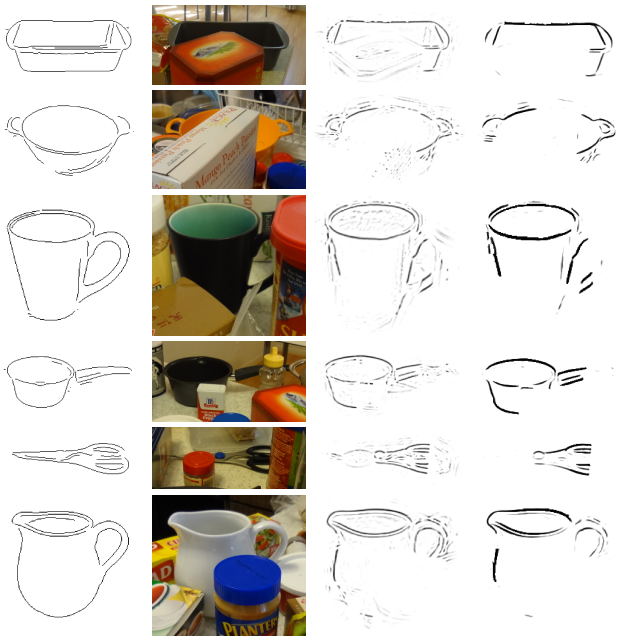

Figure 4: Results of shape matching using GN. From left to right, we show: 1) template, 2) window, 3) $\Phi^{\mathbb{S}}$, and 4) probability that each pixel matches the template.

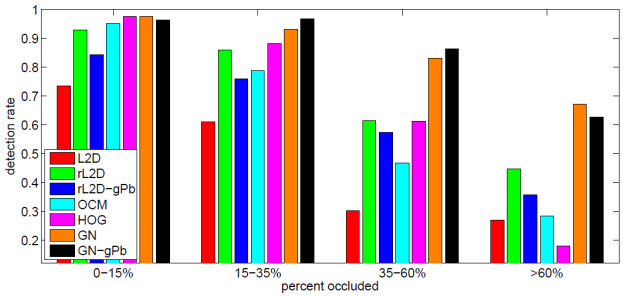

Figure 5: Detection rate under different occlusion levels. GN and GN-gPb are more robust to occlusions.

a Canny edge detector very carefully on the model images to obtain the best possible contours for GN-hard. Our results show that using a soft shape model, which does not require any parameter tuning, actually performs on par and even slightly better than a hard shape model.

Object Detection

Next we evaluate the performance for object detection. An object is correctly detected if the intersection-over-union (IoU) of the predicted bounding box and the groundtruth bounding box is greater than 0.5. The CMU_KO8 dataset is split into two parts: 800 images for single viewpoint and 800 images for multiple viewpoints. Figure 6 and 7 show the false positive per image (FPPI) versus the detection rate (DR) for these parts respectively. Table 2 summarizes the performance with the detection rate at 1.0 FPPI.

From the tables, GN significantly outperforms the other algorithms. The relative performance of the algorithms is similar to the shape matching evaluation. For objects with

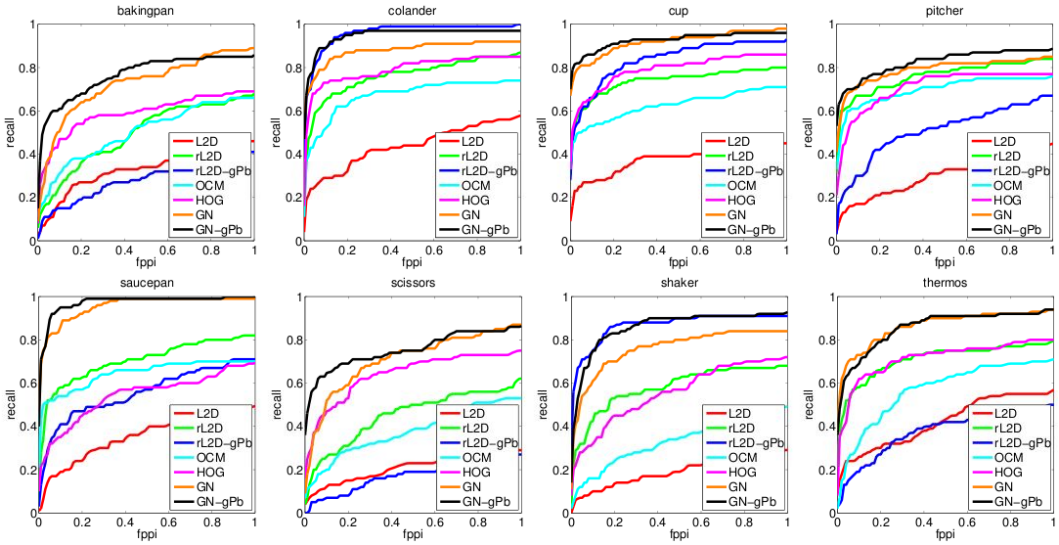

Figure 6: FPPI/DR results for single view on CMU_KO8.

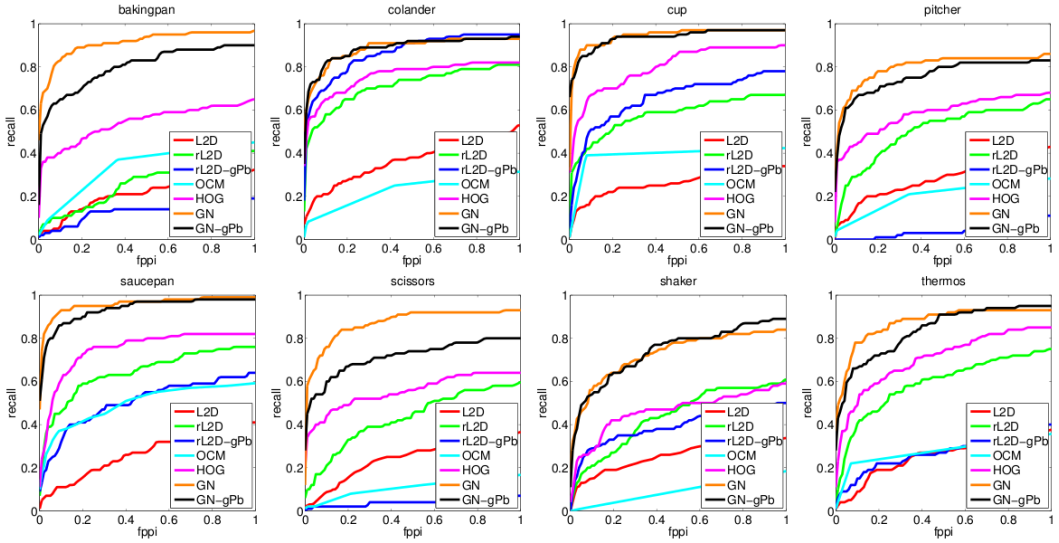

Figure 7: FPPI/DR results for multiple view on CMU_KO8.

Single	combined	orientation	color
GN	0.91	0.88	0.76
GN-hard	0.92	0.87	0.77
GN-gPb	0.93	0.85	0.86
Multiple	combined	orientation	color
GN	0.93	0.74	0.88
GN-hard	0.93	0.73	0.87
GN-gPb	0.91	0.65	0.66

Table 3: Average detection rate at 1.0 FPPI on CMU_KO8.

vibrant colors, such as the shaker (red) and colander (orange), GN-gPb performs slightly better than GN because these objects receive high gPb edge potentials. However for typical, un-colorful objects, gPb gives less confident edge potentials and this results in worse overall performance of GN-gPb for these objects. In addition, HOG performs worse than GN because it only captures the shape very coarsely and the cells covering the object boundary are easily corrupted by background clutter and occlusions.

Figure 5 shows the performance under different levels of occlusions. While many of the systems perform fairly well at low occlusion levels (0-15%), they perform significantly worse at high occlusion levels (>35%). L2D, rL2D and OCM often incorrectly have high point confidences in background clutter which result in false positives with higher score than true positives under heavy occlusion. HOG performs especially poorly because occlusions severely corrupt the descriptors of HOG cells. Our GN and GN-gPb algorithms are more robust to object occlusions, since they predict better shape similarities.

Table 3 analyzes the performance of soft versus hard shape model, different edge potentials and the effects of gradient orientation and color. Again, a soft shape model per-

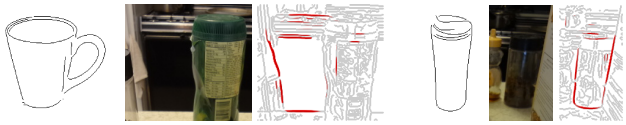

Figure 8: False positives of GN. Each triplet shows (1) template, (2) false positive window and (3) predicted match in red overlaid on the Canny edgemap.

forms equivalently to the hard model, and both the orientation and color contribute to the detection accuracy.

Figure 8 shows typical false positives of GN. These detections have long contours which align well to the image. Additional information such as occlusion reasoning (Hsiao and Hebert 2012) or interior appearance of the object is needed to filter these false positives.

Conclusion

The main contribution of this paper is to demonstrate that shape matching can incorporate edge connectivity directly on low-level gradients without extracting contours. We create a gradient network where each pixel is connected with its neighbors in the local tangent direction. Long paths which match the template shape are found using a message passing algorithm. Our results on a challenging dataset of textureless objects in realistic environments with severe occlusions demonstrate significant improvement over state-of-the-art methods for shape matching and object instance detection.

Acknowledgments

This work was supported in part by the National Science Foundation under ERC Grant No. EEEC-0540865.

References

Arbeláez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2009. From contours to regions: An empirical evaluation. In *CVPR*.

Arbeláez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2011. Contour detection and hierarchical image segmentation. *PAMI* 33(5):898–916.

Bai, X.; Li, Q.; Latecki, L.; Liu, W.; and Tu, Z. 2009. Shape band: A deformable object detection approach. In *CVPR*.

Barrow, H.; Tenenbaum, J.; Bolles, R.; and Wolf, H. 1977. Parametric correspondence and chamfer matching: two new techniques for image matching. In *IJCAI*.

Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *PAMI* 24(4):509–522.

Bhat, P.; Zitnick, C.; Cohen, M.; and Curless, B. 2010. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics* 29(2):10.

Breu, H.; Gil, J.; Kirkpatrick, D.; and Werman, M. 1995. Linear time euclidean distance transform algorithms. *PAMI* 17(5):529–533.

Dalal, N. 2006. *Finding People in Images and Videos*. Ph.D. Dissertation, Institut National Polytechnique de Grenoble / INRIA Grenoble.

Dollar, P.; Tu, Z.; and Belongie, S. 2006. Supervised learning of edges and object boundaries. In *CVPR*.

Felzenszwalb, P., and Huttenlocher, D. 2012. Distance transforms of sampled functions. *Theory of Computing* 8.

Ferrari, V.; Tuytelaars, T.; and Van Gool, L. 2006. Object detection by contour segment networks. In *ECCV*.

Freeman, W., and Adelson, E. 1991. The design and use of steerable filters. *PAMI* 13(9):891–906.

Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; and Lepetit, V. 2012. Gradient response maps for real-time detection of textureless objects. *PAMI* 34(5):876–888.

Hsiao, E., and Hebert, M. 2012. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*.

Lai, K.; Bo, L.; Ren, X.; and Fox, D. 2011. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*.

Lowe, D. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60(2):91–110.

Luo, J., and Guo, C. 2003. Perceptual grouping of segmented regions in color images. *Pattern Recognition* 36(12):2781–2792.

Mairal, J.; Leordeanu, M.; Bach, F.; Hebert, M.; and Ponce, J. 2008. Discriminative sparse image models for class-specific edge detection and image interpretation. In *ECCV*.

Malisiewicz, T., and Efros, A. A. 2011. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*.

Prasad, M.; Zisserman, A.; Fitzgibbon, A.; Kumar, M.; and Torr, P. 2006. Learning class-specific edges for object detection and segmentation. In *CVGIP*.

Scheirer, W. J.; Kumar, N.; Belhumeur, P. N.; and Boult, T. E. 2012. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*.

Shotton, J.; Blake, A.; and Cipolla, R. 2008. Multiscale categorical object recognition using contour fragments. *PAMI* 30(7):1270–1281.

Srinivasan, P.; Zhu, Q.; and Shi, J. 2010. Many-to-one contour matching for describing and discriminating object shape. In *CVPR*.

Sun, M.; Bradski, G.; Xu, B.; and Savarese, S. 2010. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*.

Thayananthan, A.; Stenger, B.; Torr, P.; and Cipolla, R. 2003. Shape context and chamfer matching in cluttered scenes. In *CVPR*.

Tu, Z. 2005. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*.

Zhao, D., and Chen, J. 1997. Affine curve moment invariants for shape recognition. *Pattern Recognition* 30(6):895–901.