MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

# Fast Directional Chamfer Matching

Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Rama Chellappa

TR2010-045 July 2010

## Abstract

We study the object localization problem in images given a single hand-drawn example or a gallery of shapes as the object model. Although many shape matching algorithms have been proposed for the problem over the decades, chamfer matching remains to be the preferred method when speed and robustness are considered. In this paper, we significantly improve the accuracy of chamfer matching while reducing the computational time from linear to sublinear (shown empirically). Specifically, we incorporate edge orientation information in the matching algorithm such that the resulting cost function is piecewise smooth and the cost variation is tightly bounded. Moreover, we present a sublinear time algorithm for exact computation of the directional chamfer matching score using techniques from 3D distance transforms and directional integral images. In addition, the smooth cost function allows to be bound the cost distribution of large neighborhoods and skip the bad hypotheses within. Experiments show that the proposed approach improves the speed of the original chamfer matching upto an order of 45x, and it is much faster than many state of art techniques while the accuracy is comparable.

*CVPR 2010*

# Fast Directional Chamfer Matching*

Ming-Yu Liu[†] Oncel Tuzel[‡] Ashok Veeraraghavan[‡] Rama Chellappa[†]
[†]University of Maryland College Park [‡]Mitsubishi Electric Research Labs
{mingyliu,rama}@umiacs.umd.edu {oncel,veerarag}@merl.com

## Abstract

*We study the object localization problem in images given a single hand-drawn example or a gallery of shapes as the object model. Although many shape matching algorithms have been proposed for the problem over the decades, chamfer matching remains to be the preferred method when speed and robustness are considered. In this paper, we significantly improve the accuracy of chamfer matching while reducing the computational time from linear to sublinear (shown empirically). Specifically, we incorporate edge orientation information in the matching algorithm such that the resulting cost function is piecewise smooth and the cost variation is tightly bounded. Moreover, we present a sublinear time algorithm for exact computation of the directional chamfer matching score using techniques from 3D distance transforms and directional integral images. In addition, the smooth cost function allows to bound the cost distribution of large neighborhoods and skip the bad hypotheses within. Experiments show that the proposed approach improves the speed of the original chamfer matching upto an order of 45x, and it is much faster than many state of art techniques while the accuracy is comparable.*

## 1. Introduction

Humans utilize extensive prior information in the form of multiple visual cues including color, texture and shape to recognize objects. Machine vision algorithms attempt to imitate human perception system by learning similar priors based on exemplars. However, collecting the required training data is a tedious task and significantly limits the effectiveness of these algorithms in many cases. For instance, it is much more appealing to search an image collection using a single exemplar supplied by a user than learning every possible object class beforehand. Yet, recognition of objects in images using only a few exemplars remains to be a very challenging problem.

The common approach to tackle this problem is to utilize features that exhibit the least variability within object classes and across imaging conditions together with a similarity measure that models the maximum invariances. Among the aforementioned visual cues, shape information largely satisfies invariances; also in many cases only a single hand-drawn contour is discriminative enough to recognize and localize an object in a cluttered image. Fast and accurate shape matching has numerous applications including object recognition and localization, image retrieval, pose estimation, and tracking.

### 1.1. Related Work

An extensive literature exists for shape matching and here we overview only a few. Several researchers proposed shape representations and similarity measures invariant to deformation or articulation of objects [3, 12]. They successfully handle intra-class variations and achieved good performance in object recognition. However, these methods typically require clean segmentation of shapes which renders them unsuitable while dealing with cluttered images.

More recent studies focus on recognition and localization of object shapes in cluttered images. In [4], the shape matching problem is posed as finding the optimal correspondences between feature points which then leads to an integer quadratic programming problem. In [10], a contour segment network framework is described where shape matching is formulated as finding paths on the network similar to model outlines. In [8], Ferrari et. al. proposed a family of scale invariant local shape descriptors (pair-of-adjacent-segment feature) which are formed by k-connected near straight contour fragments. These descriptors are later utilized in a shape matching framework [9] through a voting scheme on a Hough space. The solution is then iteratively refined using a point registration algorithm. Zhu et. al. [20] formulated shape detection as subset selection on a set of salient contours. The selection problem is approximately solved using a two-stage linear programming. In [7], a hierarchical object contour representation was proposed together with a dynamic programming approach for matching. In [15], a multi-stage approach is employed where coarse detections via matching subsets of contour segments are pruned by building the entire contour using dynamic

[*] Partially supported by MURI from the Office of Naval Research under the Grant $N00014-08-I-0638$.

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

programming.

These algorithms provided impressive results for matching shapes in cluttered images. However, they share a common drawback, high computational complexity, that makes them unsuitable for time critical applications. Although proposed decades ago, chamfer matching [1] remains to be the preferred method when speed and accuracy are considered as discussed in [18]. There exist several new variants of chamfer matching mainly to improve the cost function using orientation information [11, 16]. We discuss more about these approaches in comparison to our formulation in the following sections.

### 1.2. Contributions

In this paper, we greatly improve the accuracy of chamfer matching while significantly reducing its computational cost. We propose an alternative approach for incorporating edge orientation and solve the matching problem in an orientation augmented space. This formulation plays an active role in defining edge correspondences which result in more robust matching in highly cluttered environments. The resulting cost function is smooth and the cost variation is tightly bounded.

The best computational complexity for existing chamfer matching algorithms is linear in the number of template edge points, even without the orientation term. We optimize the directional matching cost in three stages: (1) We present a linear representation of the template edges. (2) We then describe a three dimensional distance transform representation. (3) Finally, we present a directional integral image representation over distance transforms. Using these intermediate structures, exact computation of the matching score can be performed in sublinear time in the number of edge points. In the presence of many shape templates, the memory requirement also reduces drastically. In addition, the smooth cost function allows to bound the cost distribution of large neighborhoods and skip the bad hypotheses within.

## 2. Chamfer Matching

Chamfer matching (CM) [1] is a popular technique to find the best alignment between two edge maps. Let $U = \{\mathbf{u}_i\}$ and $V = \{\mathbf{v}_j\}$ be the sets of template and query image edge maps respectively. The chamfer distance between $U$ and $V$ is given by the average of distances between each point $\mathbf{u}_i \in U$ and its nearest edge in $V$

$$d_{CM}(U,V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j|. \tag{1}$$

where $n = |U|$.

Let $\mathbf{W}$ be a warping function defined on the image plane parameterized with $\mathbf{s}$. For instance, if it is a 2D Euclidean transformation, $\mathbf{s} \in SE(2)$, $\mathbf{s} = (\theta,\ t_x,\ t_y)$, where $t_x$ and $t_y$ are the translations along $x$ and $y$ axis respectively and $\theta$ is the in-plane rotation angle. Its action on image points is given via the transformation

$$\mathbf{W}(\mathbf{x};\mathbf{s}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \mathbf{x} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \tag{2}$$

The best alignment parameter $\hat{\mathbf{s}} \in SE(2)$ between the two edge maps is then given by

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in SE(2)} d_{CM}(\mathbf{W}(U;\mathbf{s}), V) \tag{3}$$

where $\mathbf{W}(U;\mathbf{s}) = \{\mathbf{W}(\mathbf{u}_i, \mathbf{s})\}$.

Chamfer matching provides a fairly smooth measure of fitness, and can tolerate small rotations, misalignments, occlusions, and deformations. The matching cost can be computed efficiently via a distance transform image $DT_V(\mathbf{x}) = \min_{\mathbf{v}_j \in V} |\mathbf{x} - \mathbf{v}_j|$ which specifies the distance from each pixel to the nearest edge pixel in $V$. The distance transform can be computed in two passes over the image [6] and using which the cost function (1) can be evaluated in linear time $O(n)$ via $d_{CM}(U,V) = \frac{1}{n}\sum_{\mathbf{u}_i \in U} DT_V(\mathbf{u}_i)$.

## 3. Directional Chamfer Matching

Chamfer matching becomes less reliable in the presence of background clutter. To improve robustness, several variants of chamfer matching have been introduced by incorporating edge orientation information into the matching cost. In [5, 11], the template and query image edges are quantized into discrete orientation channels and individual matching scores across channels are summed. Although this method alleviates the problem of cluttered scenes, the cost function is very sensitive to the number of orientation channels and becomes discontinuous in channel boundaries. In [16], the chamfer distance is augmented with an additional cost for orientation mismatch which is given by the average difference in orientations between template edges and their nearest edge points in the query image. The method is called oriented chamfer matching and throughout the paper we use the abbreviation OCM.

Instead of an explicit formulation of the orientation mismatch, we generalize the chamfer distance to points in $\mathbb{R}^3$ for matching directional edge pixels. Each edge point $\mathbf{x}$ is augmented with a direction term $\phi(\mathbf{x})$ and the directional chamfer matching (DCM) score is given by

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j| + \lambda |\phi(\mathbf{u}_i) - \phi(\mathbf{v}_j)| \tag{4}$$

where $\lambda$ is a weighting factor between location and orientation terms. Note that the directions $\phi(\mathbf{x})$ are computed at modulo $\pi$, and the orientation error gives the minimum circular difference between the two directions $min\{|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)|, ||\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)| - \pi|\}$.

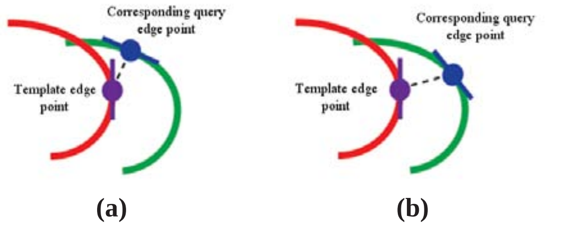
Corresponding query edge point
Template edge point
Corresponding query edge point
Template edge point
(a)
(b)

Figure 1. Matching costs per edge point for (a) Oriented chamfer matching [16]; (b) Directional chamfer matching. DCM jointly minimizes location and orientation errors whereas in [16] the location error is augmented with the orientation error of the nearest edge point.

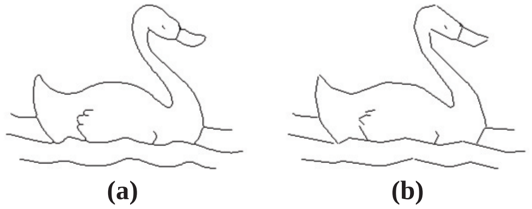

Figure 2. Linear representation. (a) Edge image. The image contains 1242 edge points. (b) Linear representation of the edge image. The image contains 60 line segments.

In Figure 1, we present a comparison of the proposed cost function with [16]. It can be easily verified that the proposed matching cost is a piecewise smooth function of both the translation $t_x$, $t_y$ and the rotation $\theta$ of the template edges. Therefore, matching is more robust against clutter, missing edges and small misalignments.

# 4. Search Optimization

To the best of our knowledge, the lowest computational complexity for the existing chamfer matching algorithms is linear in the number of template edge points, even without the directional term. In this section, we present a sublinear time algorithm for exact computation of the 3D chamfer matching score (4).

## 4.1. Linear Representation

The edge map of a scene does not follow an unstructured binary pattern. Instead, the object contours comply with certain continuity constraints which can be retained by concatenating line segments of various lengths, orientations and translations. Here, we represent an edge image with a collection of $m$-line segments. Compared with a set of points which has cardinality $n$, its linear representation is more concise. It requires only $O(m)$ memory to store an edge map where $m << n$.

We use a variant of RANSAC algorithm to compute the linear representation of an edge map. The outline of the algorithm is as follows. The algorithm initially hypothesizes a variety of lines by selecting a small subset of points and their directions. The support of a line is given by the set of points which satisfy the line equation within a small residual and form a continuous structure. The line segment with the largest support is retained and the procedure is iterated with the reduced set until the support becomes smaller than a few points.

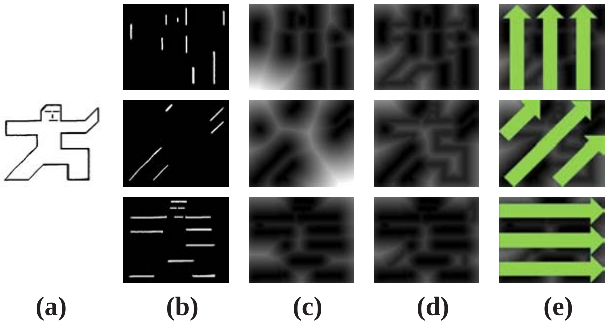

Figure 3. Computation of the integral distance transform tensor. (a) The input edge map. (b) Edges are quantized into discrete orientation channels. (c) Two dimensional distance transform of each orientation channel. (d) The three dimensional distance transform $DT3$ is updated based on the orientation cost. (e) $DT3$ tensor is integrated along the discrete edge orientations and integral distance transform tensor, $IDT3$, is computed.

The algorithm only retains points with certain structure and support, therefore the noise is filtered. In addition, the directions recovered through the line fitting procedure are more precise compared to those computed using local operators such as image gradients. An example of linear representation is given in Figure 2 where a set of 1242 points is modeled with 60 line segments.

## 4.2. Three-Dimensional Distance Transform

The matching score given in (4) requires finding the minimum cost match over location and orientation terms for each template edge point. Therefore the computational complexity of the brute-force algorithm is quadratic in the number of template and query image edge points. Here we present a three-dimensional distance transform representation ($DT3$) to compute the matching cost in linear time. We note that, a similar structure was also used in [14] for fast evaluation of Hausdorff distances.

This representation is a three dimensional image tensor where the first two dimensions are the locations on the image plane and the third dimension is the quantized edge orientations. The orientations are quantized into $q$ discrete channels $\hat{\Phi} = \{\hat{\phi}_i\}$ evenly in $[0 \quad \pi)$ range. Each element of the tensor encodes the minimum distance to an edge point in the joint location and orientation space:

$$DT3_V(\mathbf{x}, \phi(\mathbf{x})) = \min_{\mathbf{v}_j \in V} |\mathbf{x} - \mathbf{v}_j| + \lambda |\hat{\phi}(\mathbf{x}) - \hat{\phi}(\mathbf{v}_j)|. \quad (5)$$

where $\hat{\phi}(\mathbf{x})$ is the nearest quantization level in orientation space to $\phi(\mathbf{x})$ in $\hat{\Phi}$.

We present an algorithm to compute $DT3$ tensor in $O(q)$ passes over the image by solving two dynamic programs consecutively. Equation (5) can be rewritten as

$$DT3_V(\mathbf{x}, \phi(\mathbf{x})) = \min_{\hat{\phi}_i \in \hat{\Phi}} \left( DT_{V\{\hat{\phi}_i\}} + \lambda|\hat{\phi}(\mathbf{x}) - \phi_i| \right) \quad (6)$$

where $DT_{V\{\hat{\phi}_i\}}$ is the two dimensional distance transform of the edge points in $V$ having orientation $\hat{\phi}_i$.

Initially, we compute $q$ two dimensional distance transforms $DT_{V\{\hat{\phi}_i\}}$ which requires $O(q)$ passes over the image using the standard distance transform algorithm [6]. Subsequently, the $DT3_V$ tensor (6) is computed by solving a second dynamic program for each location separately. The tensor is initialized with the two dimensional distance transforms $DT3_V(\mathbf{x}, \hat{\phi}_i) = DT_{V\{\hat{\phi}_i\}}(\mathbf{x})$ and is updated with a forward recursion

$$\begin{aligned} DT3_V(\mathbf{x}, \hat{\phi}_i) = min \quad & \{DT3_V(\mathbf{x}, \hat{\phi}_i), \quad (7) \\ & DT3_V(\mathbf{x}, \hat{\phi}_{i-1}) + \lambda|\hat{\phi}_{i-1} - \hat{\phi}_i|\} \end{aligned}$$

and a backward recursion

$$\begin{aligned} DT3_V(\mathbf{x}, \hat{\phi}_i) = min \quad & \{DT3_V(\mathbf{x}, \hat{\phi}_i), \quad (8) \\ & DT3_V(\mathbf{x}, \hat{\phi}_{i+1}) + \lambda|\hat{\phi}_{i+1} - \hat{\phi}_i|\} \end{aligned}$$

for each location $\mathbf{x}$. Unlike the standard distance transform algorithm, a special condition is needed for handling the circular orientation distance. The forward and backward recursions do not terminate after a full cycle, $i = 1 \dots q$ or $i = q \dots 1$ respectively, but the costs are continued to be updated in a circular form until the cost for a tensor entry is not changed. Note that, at most one and a half cycles are needed for each of the forward and backward recursions, therefore the worst computational cost is $O(q)$ passes over the image. Using the three dimensional distance transform representation $DT3_V$ the directional chamfer matching score of any template $U$ can be computed in linear time via

$$d_{DCM}(U, V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} DT3_V(\mathbf{u}_i, \hat{\phi}(\mathbf{u}_i)). \quad (9)$$

### 4.3. Distance Transform Integral

Let $L_U = \{l_{[\mathbf{s}_j, \mathbf{e}_j]}\}_{j=1 \dots m}$ be the linear representation of template edge points $U$ where $\mathbf{s}_j$ and $\mathbf{e}_j$ are the start and end locations of the $j$-th line respectively. For ease in notation, we sometimes refer to a line with only its index $l_j = l_{[\mathbf{s}_j, \mathbf{e}_j]}$. We assume that the line segments only have directions among the $q$ discrete channels $\hat{\Phi}$, which is enforced while computing the linear representation. Although it might be argued that the discreet line directions introduce quantization artifacts, in fact the linear representation given in Figure 2b is generated using only $q = 60$ directions and it is difficult to observe the difference from the original edge image (Figure 2a).

All the points on a line segment are associated with the same orientation which is the direction of the line $\hat{\phi}(l_j)$. Hence the directional chamfer matching score (9) can be rearranged as

$$d_{DCM}(U, V) = \frac{1}{n} \sum_{l_j \in L_U} \sum_{\mathbf{u}_i \in l_j} DT3_V(\mathbf{u}_i, \hat{\phi}(l_j)). \quad (10)$$

In this formulation, the $i$-th orientation channel of the $DT3_V$ tensor, $DT3_V(\mathbf{x}, \hat{\phi}_i)$, is only evaluated for summing over the points of line segments having direction $\hat{\phi}_i$.

Integral images are intermediate image representations used for fast calculation of region sums [19] and linear sums [2]. Here we present a tensor of integral distance transform representation ($IDT3_V$) to evaluate the summation of costs over any line segment in $O(1)$ operations. For each orientation channel $i$, we compute the one-directional integral along $\hat{\phi}_i$ (Figure 3).

Let $\mathbf{x}_0$ be the intersection of an image boundary with the line passing through $\mathbf{x}$ and having the direction $\hat{\phi}_i$. Each entry of $IDT3_V$ tensor is given by

$$IDT3_V(\mathbf{x}, \hat{\phi}_i) = \sum_{\mathbf{x}_j \in l_{[\mathbf{x}_0, \mathbf{x}]}} DT3_V(\mathbf{x}_j, \hat{\phi}_i). \quad (11)$$

The $IDT3_V$ tensor can be computed in one pass over the $DT3_V$ tensor. Using this representation, the directional chamfer matching score of any template $U$ can be computed in $O(m)$ operations via

$$\begin{aligned} d_{DCM}(U, V) = \quad & \frac{1}{n} \sum_{l_{[\mathbf{s}_j, \mathbf{e}_j]} \in L_U} [IDT3_V(\mathbf{e}_j, \hat{\phi}(l_{[\mathbf{s}_j, \mathbf{e}_j]})) - \\ & IDT3_V(\mathbf{s}_j, \hat{\phi}(l_{[\mathbf{s}_j, \mathbf{e}_j]}))]. \quad (12) \end{aligned}$$

The $O(m)$ complexity is only an upper bound on the number of computations. For object detection or localization we would like to retain only the hypotheses where the matching costs are less than the detection threshold or equivalently for localization less than the best hypothesis. We order the template lines with respect to their supports and start the summation from the line with the largest support. A hypothesis is eliminated during the summation if the cost is larger than the detection threshold or current best hypothesis. The supports of the line segments show exponential decay, therefore for majority of the hypotheses only a few arithmetic operations are performed. We **empirically** show that the number of evaluated line segments is **sublinear** in the number of template points $n$.

### 4.4. Optimized Region Search

The proposed DCM cost function (4) is smooth. Moreover, the variation of the matching cost is bounded and only changes smoothly in the spatial domain. We utilize this fact to significantly reduce the amount of hypotheses evaluated from an image. Let $\delta \in \mathbb{R}^2$ be the translation of the model $U$ in the image plane. The DCM cost variation due to translation then becomes

$$\begin{aligned} & d_{DCM}(U+\delta,V) \\ & = \tfrac{1}{n}\textstyle\sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i + \delta - \mathbf{v}_j| + \lambda|\phi(\mathbf{u}_i) - \phi(\mathbf{v}_j)| \\ & \leq \tfrac{1}{n}\textstyle\sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j| + |\delta| + \lambda|\phi(\mathbf{u}_i) - \phi(\mathbf{v}_j)| \\ & \qquad\qquad = |\delta| + d_{DCM}(U,V). \end{aligned} \tag{13}$$

Therefore, the variation of the DCM cost is bounded by the spatial translation $|d_{DCM}(U+\delta,V) - d_{DCM}(U,V)| \leq |\delta|$. If the targeting matching cost is $\epsilon$ and the cost of the current hypothesis is $\psi$, i.e. $\epsilon < \psi$, there can not be a detection within the $|\psi - \epsilon|$ pixel range of the current hypothesis and we can skip the evaluation of the hypotheses within this region.

## 5. Experiments

We conducted three sets of experiments on challenging synthetic and real datasets. Note that, in all our experiments we emphasize the speed and the improved accuracy of our approach compared to chamfer matching. Although the performance of proposed approach is comparable with state of art methods, it can also be utilized to quickly retrieve accurate initial hypotheses which can then be refined using more expensive point registration algorithms. In all our experiments we used $q = 60$ orientation channels and 6 degrees error in orientation corresponds to 1 pixel distance.

### 5.1. Object Detection and Localization

In the first experiment, we performed object detection and localization on the ETHZ shape class dataset [10]. The dataset contains $255$ images where each image contains one or more objects from five different classes: apple logos, bottles, giraffes, mugs, and swans. The objects in the dataset have large variations in appearance, viewpoint, size, and non-rigid deformation. We followed the experimental setup proposed in [10, 9] in which a single hand-drawn shape for each class was used to detect and localize its instances in the dataset.

Our detection system is based on scanning using a sliding window, i.e., we retain all the hypotheses where the cost function is less than the detection threshold. To illustrate the speed of the algorithm, we densely sampled the hypothesis space and searched the images at 8 different scales and 3 different aspect ratios. The ratio between two consecutive scales was $1.2$ and the aspect ratio was $1.1$. We performed non-maxima suppression by retaining only the lowest cost hypothesis among the ones which had significant overlap.

In Figure 4, we report false positive per image vs. detection rate. The curve is generated via altering the detection threshold for the matching cost. We compared our approach with the oriented chamfer matching [16] and two recent studies proposed by Ferrari et. al. [10, 9]. Our approach is significantly superior to the oriented chamfer matching at all the false positive rates and comparable to [9] where our results are better for two classes (giraffes and bottles) and slightly worse for the swans class while for two other classes, the numbers are almost identical. As shown in the detection examples (Figure 4), object localization is extremely accurate. We note that in [20] and [15], slightly better performances were reported on this dataset. As these results were presented in different formats we could not include them in our graphs.

| Algorithm | DCM | OCM | CM |
|---|---|---|---|
| Time ($\mu s$) | 0.40 | 51.50 | 17.59 |

Table 1. Hypothesis evaluation time comparison. The evaluation time is averaged over the $5$ hand-drawing shapes used to detect object in the ETHZ dataset

In Table 1, we present the mean evaluation time of matching costs per hypothesis. The average number of points in the shape templates were $1610$, computed over five classes. Similarly, on average our linear representation included $39$ lines per class. The number of lines per class is only an upper bound on the number of computations. Since the algorithm retrieves only the hypotheses having a smaller cost than the detection threshold, the summation was terminated for a hypothesis as the cost reached this value. On average only $14$ lines were evaluated per hypothesis. The results indicate that the proposed method improves the speed of chamfer matching by $43$x and of the oriented chamfer matching by $127$x. Note that, the speed up is more significant for larger size templates since our cost computation is insensitive to the template size whereas the cost of the original chamfer matching increases linearly.

On average, we evaluated $1.05$ million hypotheses per image which took $0.42$ seconds. Using the bounding technique presented in Section 4.4, we further reduced the average processing time per image to $0.39$ seconds where approximately $91\%$ of the hypotheses were skipped. Note that, while using the bounding function we needed to compute the full cost function (summation over all lines) for each evaluated hypothesis. Therefore, the speedups is not proportional.

### 5.2. Human Pose Estimation

In the second experiment, we utilized the derived shape matching framework for human pose estimation which is a very challenging task due to large variations in appearances of human poses. As proposed in [13], we matched a gallery of human shapes with known poses to the given observation. Due to articulation, the size of the pose gallery needed for accurate pose estimation is quite large. Hence, it becomes increasingly important to have an efficient matching

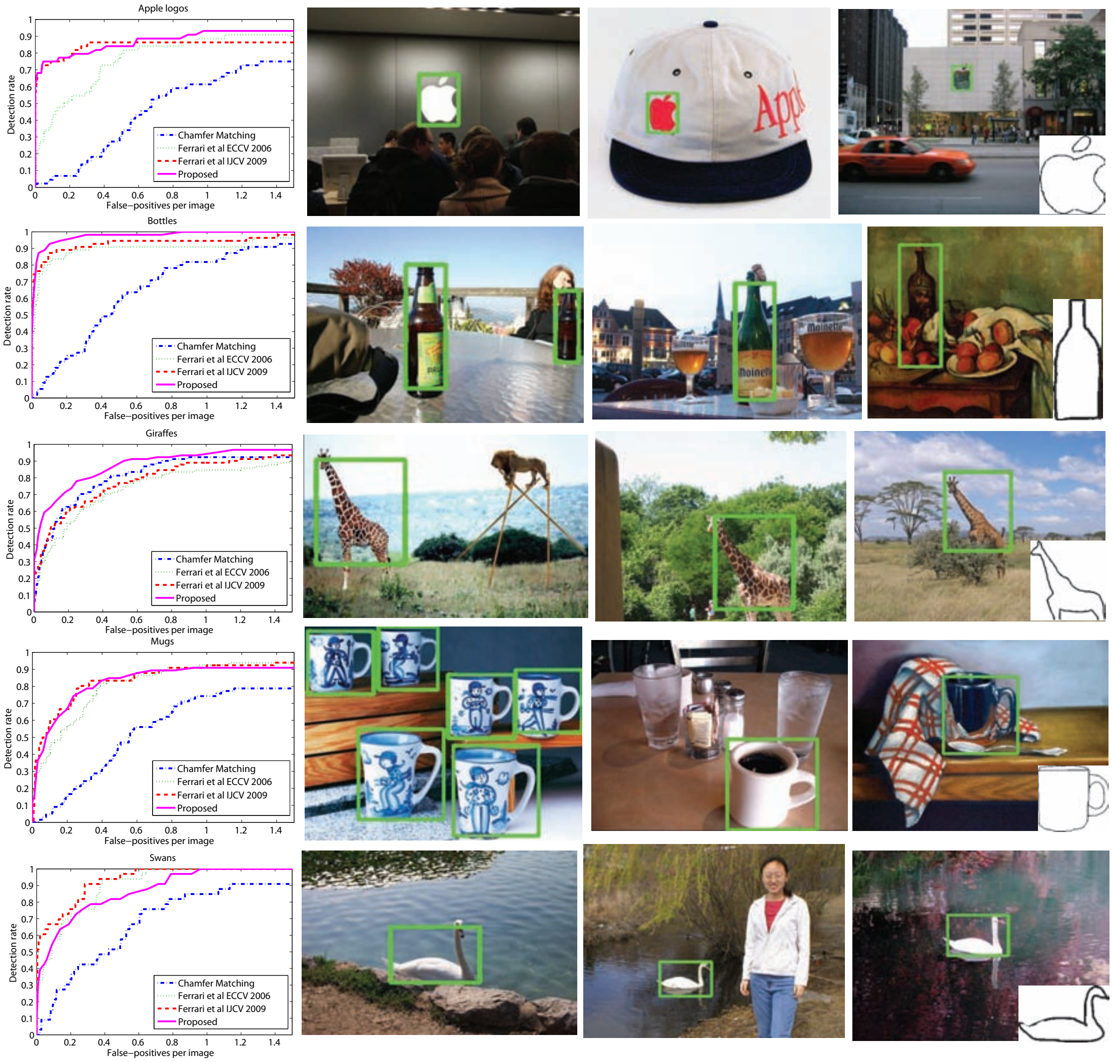

Figure 4. ROC curve and several localization results on the ETHZ shape dataset. The images are searched using a single hand-drawn shape shown on the side. The proposed approach achieved performance comparable to [9].

algorithm which can cope with background clutter.

The experiments were also performed on the HumanEva dataset [17] which contains video sequences of multiple human subjects performing various activities captured from different viewing directions. The ground truth locations of human joints at each image were extracted using the attached markers. Shape gallery templates were acquired in two steps. First, we computed the human silhouettes via HumanEva background subtraction code. Then, using the Canny edges around the extracted silhouette outlines, we obtained the shape templates. We included all the images from an action sequence (about 1,000 - 2,000) to the shape gallery which is then used to estimate human poses in different sequences. As we extracted Canny edges directly from the test images, they included significant amount of background clutter. The best shape template together with its scale and location is then retrieved via the matching framework. We quantitatively evaluated the mean absolute error between the marker locations of the ground truth and the estimated pose on the image plane. The results are presented in Table 2 where we observe significantly improved accuracy compared to chamfer matching and the oriented chamfer matching. The proposed approach can evaluate more than 1.1 million hypotheses per second whereas chamfer matching and the oriented chamfer matching can evaluate 31000 and 14000 hypotheses per second respectively. Several pose estimation examples are given in Figure 5.

### 5.3. Synthetic Experiments

In the last experiment, we estimated the 3D pose of several industrial parts using synthetic data. The 3D CAD models of the objects were given in advance. We decomposed the 3D rotation matrix into in-plane and out-of-plane rotations, and generated a gallery of shape templates from

| Algorithm | Walking | Jogging | Boxing | Avg. |
|---|---|---|---|---|
| DCM | 7.3 | 12.5 | 9.7 | 9.8 |
| OCM | 15.0 | 15.3 | 13.6 | 14.6 |
| CM | 9.3 | 13.6 | 10.6 | 11.2 |

Table 2. Pose estimation errors on three action sequences. Errors are measured as the mean absolute pixel distance from the ground truth marker locations.

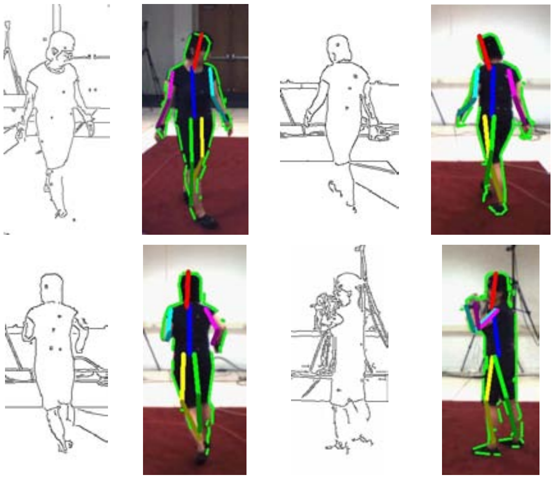

Figure 5. Human pose estimation results. First row: Walking sequence. Second row: Jogging and boxing sequences. Estimated poses and contours are overlayed onto the images.

the uniformly sampled out-of-plane rotations (300 poses for each object) via rendering the 3D CAD model and detecting the depth edges (edges due to depth discontinuities). Two samples among the 300 shape templates for each object are shown in the first row of Figure 6.

We also generated a synthetic test set with a similar procedure. We randomly generated 3D pose parameters for each object and simultaneously inserted all the objects in the scene. A few of the generated test images are shown in the second row of Figure 6. As seen in the images, the objects have large overlaps, therefore a significant amount of the edges are occluded. Moreover, the images were corrupted with noise via adding uniformly sampled line segments and a small fraction of the detected edges were also removed. The test set included 500 images.

We retrieved the best gallery pose together with in-plane rotation and translation parameters via the proposed shape matching algorithm. Several pose estimation results for five different objects are shown in the third row of Figure 6. Full 3D pose of the objects were then recovered for a known depth using the estimated in-plane transformation parameters together with the out-of-plane rotation parameters that generated the gallery templates. An estimate was labeled as correct if the three estimated angles were within 10 degrees and the position was within 5 mm. of the ground truth pose. As seen in Table 3, on average, our approach reduces estimation errors by $75\%$ compared to chamfer matching and $40\%$ compared to orientated chamfer matching. In this experiment, we were able to estimate the 3D pose of an object in $.71$ seconds via the proposed approach whereas the same process took $29.1$ and $65.3$ seconds via chamfer matching and the oriented chamfer matching respectively.

| Algorithm | Circuit | Ellipse | T-Nut | Knob | Wheel | Avg. |
|---|---|---|---|---|---|---|
| DCM | 0.03 | 0.05 | 0.11 | 0.04 | 0.08 | 0.06 |
| OCM | 0.05 | 0.14 | 0.17 | 0.04 | 0.17 | 0.11 |
| CM | 0.11 | 0.26 | 0.34 | 0.26 | 0.22 | 0.24 |

Table 3. Miss rates for synthetic 3D pose estimation experiment.

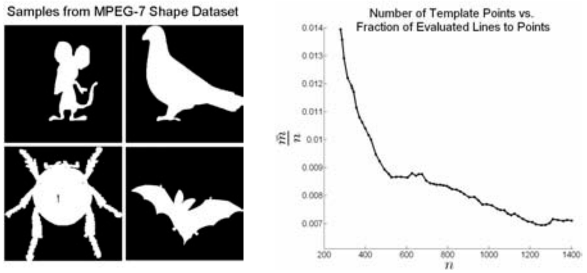

Figure 7. Empirical evidence of sublinearity. See text for details.

## 5.4. Empirical Evidence for Sublinear Complexity

When the template shapes are scaled, sublinear complexity trivially holds. In this case, as the number of edge points, $n$, increases the cardinality of the linear representation, $m$, remains constant which implies constant complexity for matching. We also provide empirical evidence that the matching complexity is a sublinear function of the number of template points in a more general setup. As we discussed before, the $O(m)$ complexity is only an upper bound on the number of evaluations, and on average we need to evaluate only a fraction of those lines to find the minimum cost match. Empirically, we evaluate $\bar{m}$ lines which fit $20\%$ - $30\%$ of the template points and most of the energy is concentrated in only a few lines. In Figure 7, we plot the number of template points, $n$, versus the fraction of evaluated lines to points, $\frac{\bar{m}}{n}$, where $\bar{m}$ is selected as the number of lines that fit $30\%$ of template points. The curve is generated using 1000 shape images from the MPEG-7 dataset. We observe that as the number of template points increases the fraction of evaluated lines decreases, which provides empirical evidence that the algorithm is sublinear in the number of template points ($< O(n)$). Note that, there is no bias factor involved since $n$ vs. $\bar{m}$ curve passes through zero.

## 6. Conclusion

We presented a novel approach for improving the accuracy of chamfer matching while significant reducing its computational cost. We proposed an alternative approach for incorporating the edge orientation in the cost function and solved the matching problem in the orientation augmented space. The novel cost function is smooth and can be computed in sublinear time in the size of the shape

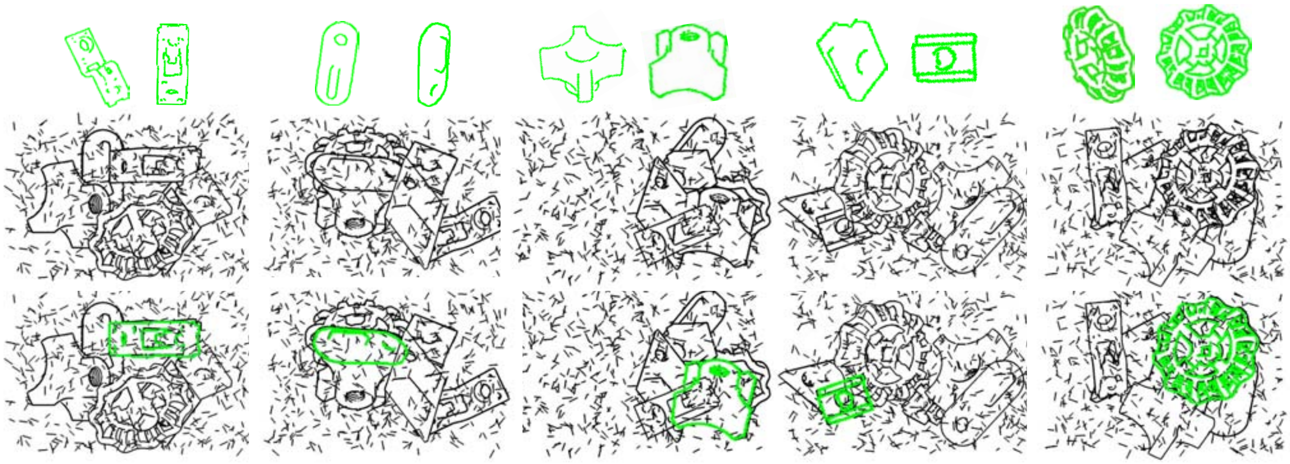

Figure 6. 3D pose estimation. First row: Samples from the shape gallery. Second row: Query images. Third row: Pose estimation results.

template. We demonstrated the superior performance of the algorithm on three challenging applications where we achieved speedups up to an order of 45x while drastically reducing the matching errors.

## References

[1] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Int'l Joint Conf. of Artif. Intel.*, pages 659–663, 1977.

[2] C. Beleznai and H. Bischof. Fast human detection in crowded scenes by contour integration and local shape estimation. In *IEEE Conf. on Comp. Vis. and Pattern Rec.*, pages 2246–2253, 2009.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Anal. and Mac. Intel.*, 24(4):509–522, 2002.

[4] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Conf. on Comp. Vis. and Pattern Rec.*, pages 26–33, 2005.

[5] O. Danielsson, S. Carlsson, and J. Sullivan. Automatic learning and extraction of multi-local features. In *Int'l Conf. on Comp. Vis.*, 2009.

[6] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.

[7] P. Felzenszwalb and J. Schwartz. Hierarchical matching of deformable shapes. In *Proc. IEEE Conf. on Comp. Vis. and Pattern Rec.* IEEE Computer Society, 2007.

[8] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. on Pattern Anal. and Mac. Intel.*, 30(1):36–51, 2008.

[9] V. Ferrari, F. Jurie, , and C. Schmid. From images to shape models for object detection. *Int'l Journal of Comp. Vis.*, 2009.

[10] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *Proc. European Conf. on Comp. Vis.*, volume 3953 of *LNCS*, pages 14–28. Elsevier, June 2006.

[11] D. M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Int'l Conf. on Pattern Rec.*, pages 439–444, 1998.

[12] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. on Pattern Anal. and Mac. Intel.*, 29(2):286–299, 2007.

[13] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. European Conf. on Comp. Vis.*, volume 3, pages 666–680, 2002.

[14] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Trans. on Pattern Anal. and Mac. Intel.*, 6(1):103–113, 1997.

[15] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. In *Proc. European Conf. on Comp. Vis.*, pages 483–496, 2008.

[16] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *IEEE Trans. on Pattern Anal. and Mac. Intel.*, 30(7):1270–1281, 2008.

[17] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, 2006.

[18] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proc. IEEE Conf. on Comp. Vis. and Pattern Rec.*, pages 127–133, 2003.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Comp. Vis. and Pattern Rec.*, volume 1, pages 511–518, 2001.

[20] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *Proc. European Conf. on Comp. Vis.*, pages 774–787, 2008.