Shape Context and Chamfer Matching in Cluttered Scenes

A. Thayananthan * B. Stenger * P. H. S. Torr † R. Cipolla *

* University of Cambridge
Department of Engineering
Cambridge, CB2 1PZ, UK
{at315|bdrs2|cipolla}@eng.cam.ac.uk

† Microsoft Research Ltd.
7 JJ Thompson Avenue
Cambridge, CB3 OFB, UK
philtorr@microsoft.com

Abstract

This paper compares two methods for object localization from contours: shape context and chamfer matching of templates. In the light of our experiments, we suggest improvements to the shape context: Shape contexts are used to find corresponding features between model and image. In real images it is shown that the shape context is highly influenced by clutter, furthermore even when the object is correctly localized, the feature correspondence may be poor. We show that the robustness of shape matching can be increased by including a figural continuity constraint. The combined shape and continuity cost is minimized using the Viterbi algorithm on features sequentially around the contour, resulting in improved localization and correspondence. Our algorithm can be generally applied to any feature based shape matching method.

Chamfer matching correlates model templates with the distance transform of the edge image. This can be done efficiently using a coarse-to-fine search over the transformation parameters. The method is robust in clutter, however multiple templates are needed to handle scale, rotation and shape variation. We compare both methods for locating hand shapes in cluttered images, and applied to word recognition in EZ-Gimpy *images.*

1. Introduction

People use multiple visual cues to recognize objects, such as object color, texture and shape. In the absence of color and texture information, we can mostly still recognize objects by their geometry alone, for example in line drawings. Grouping low level features to segment the object is by itself a hard problem. A common approach is, therefore, to use a prototype shape, and search for it in the image. This leads to the task of shape matching, which has numerous applications, such as object localization, image retrieval, model registration, and tracking. One way to represent a shape is by a set number of feature points, for example Canny edges. In order to match two shapes, point correspondences on the two shapes have to be established. Subsequently a transformation which aligns the two shapes can be found. The type of transformation depends on the particular setting. Two examples are 2D affine transforms, and non-rigid thin-plate spline transformations. The two problems of finding correspondences and estimating the transformation are tightly coupled: The better the correspondences are known, the better the transformation can be estimated, and vice versa. Therefore, many methods are based on an iterated two-step algorithm, alternating estimation of correspondence and transformation.

In the next section, we review existing work on shape based and chamfer matching. The two methods are explained briefly in section 2, and we outline some of the problems that arise when applied to scenes with cluttered background in section 3. In section 4 we show how shape context matching can be significantly improved by using a continuity constraint. The dynamic programming algorithm used for optimization readily generalizes to any other type of feature. Section 5 shows experimental results on two types of data, images of hands, and words on textured background.

1.1. Previous Work

Belongie *et al.* [3] have introduced the *shape context* descriptor, which characterizes a particular point location on the shape. This descriptor is the histogram of the relative polar coordinates of all other points. Corresponding points on two different shapes have a similar relative position in each shape, and will ideally have a similar shape context. Shape context matching has been applied to a variety of object recognition problems [3, 13]. The background clutter in these applications was usually limited.

Sullivan and Carlsson [17] use a topology-based shape descriptor to find correspondences. The *topological type* of all combinations of four points is recorded in a voting matrix, and one-to-one correspondences are found using a greedy algorithm. The examples shown did not contain significant clutter. While their topological descriptor has higher discriminative power than the shape context, computing the descriptor for all combinations of four points is of complexity $O(n^4)$ (n number of points), and is significantly slower than computing shape contexts, which is of complexity $O(n^2)$. Both methods use shape descriptors without enforcing any continuity constraint, resulting in a number of incorrect correspondences. This shortcoming may sometimes be compensated by iterative alignment and recomputation of the shape descriptor. However, this is computationally expensive, and it would be desirable to obtain good correspondences in the first step.

Chamfer matching was first proposed by Barrow *et al.* [2] and improved versions have been used for object recognition and contour alignment. Borgefors [5] introduced hierarchical chamfer matching, in which a coarse-to-fine search is performed using a resolution pyramid of the image. Olson and Huttenlocher [15] use a template hierarchy to recognize three dimensional objects from different views. They also demonstrate the importance of using oriented edge information for Hausdorff matching, which is closely related to chamfer matching. Gavrila [9] uses chamfer matching to detect pedestrian shapes in real time. In this case a template hierarchy is used to handle shape variation.

When a single template is used, chamfer matching cannot handle large shape variations. Either multiple templates have to be used, or, if the initial localization is good, the shapes can subsequently be aligned using point registration. A standard method for point registration is the Iterated Closest Point (ICP) algorithm [4, 6], where correspondences are found using a nearest-neighbor assignment, and the transformation is estimated by minimizing the geometric error between point pairs. ICP is fast and converges to a local minimum. However, it requires a good initial alignment of model and image. A number of improved point registration methods have been developed recently [7, 8, 11]. Fitzgibbon [8] introduced a version of the ICP algorithm which combines the correspondence and the alignment steps within the structure of the Levenberg-Marquardt algorithm.

2. Methods

In this section we explain the two methods of shape context matching and chamfer matching.

2.1. Shape Context Matching

The shape context descriptor for a point on the shape is a histogram of the relative polar coordinates of all other points on the shape [3]. Point correspondences between two shapes are found by minimizing the point matching costs, which is the χ^2 test statistic for histograms. Globally optimal correspondences are found by minimizing the sum of the individual matching costs. This is solved with a

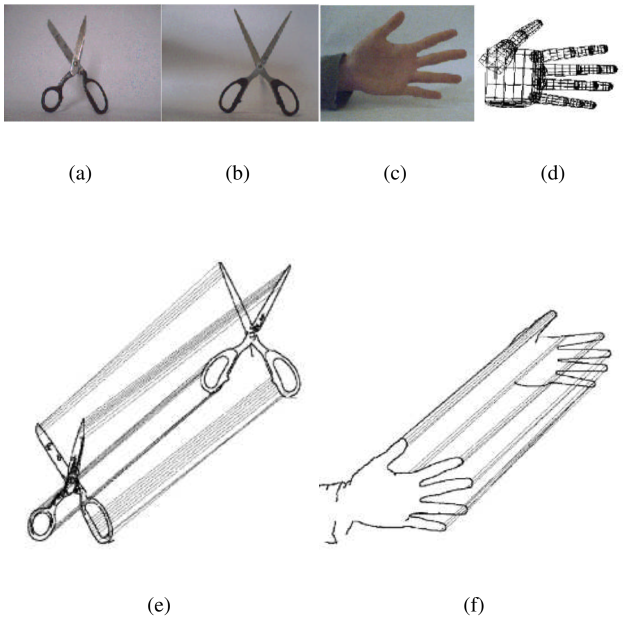


Figure 1: **Point correspondences found with shape contexts.** *Shape contexts can be used to find corresponding points on similar shapes in uncluttered scenes. (a,b) Images of two pairs of scissors. (e) Connections between corresponding points. (c,d) Images of a hand and a 3D hand model. (f) Corresponding points between edge map of (c) and projected contours of (d). For visual clarity not all correspondences are shown.*

bi-partite graph matching algorithm, enforcing one-to-one point matching. igure 1 shows point correspondences between different shapes which were found using the shape context descriptor. The shape context descriptor has the following invariance properties.

1. Translation: The shape context descriptor is inherently translation invariant as it is based on relative point locations.

2. Scale: For clutter-free images the descriptor can be made scale invariant by normalizing the radial distances by the mean (or median) distance between all point pairs.

3. Rotation: It can be made rotation invariant by rotating the coordinate system at each point so that the positive x-axis is aligned with the tangent vector. However, this reduces the discriminative power of the descriptor significantly, and is therefore not used here.

4. Shape variation: The shape context is robust towards slight shape variations. When points in the shape vary a lot, the discrete binning effect will lead to larger matching costs, and wrong matches.

5. Few outliers: Points with a final matching cost larger than a threshold value ϵ are classified as outliers. Additional

'dummy' points with the cost ϵ are introduced to make the number of points on the two shapes equal, and the points matched to these dummy points are also classified as outliers. A common way to increase the robustness towards outliers is to use knowledge from the model and only use those bins for computing the matching cost which are non-empty for the model point.

2.2. Chamfer Matching

The similarity between two shapes can be measured using their chamfer distance. Given the two point sets $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^n$ and $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^m$, the chamfer distance function is the mean of the distances between each point, $u_i \in \mathcal{U}$ and its closest point in $\mathcal{V}$:

$$d_{cham}(\mathcal{U}, \mathcal{V}) = \frac{1}{n} \sum_{u_i \in \mathcal{U}} \min_{v_j \in \mathcal{V}} ||u_i - v_j||. \tag{1}$$

The symmetric chamfer distance is obtained by adding $d_{cham}(\mathcal{V}, \mathcal{U})$. The chamfer distance between two shapes can be efficiently computed using a distance transform (DT). This transformation takes a binary feature image as input, and assigns to each pixel in the image the distance to its nearest feature. The distance between a template and an edge map can then be computed as the mean of the DT values at the template point coordinates. The matching can be made more robust by using the mean of the thresholded distance

$$d_{cham,\tau}(\mathcal{U}, \mathcal{V}) = \frac{1}{n} \sum_{u_i \in \mathcal{U}} \max \left(\min_{v_j \in \mathcal{V}} ||u_i - v_j||, \tau \right) \tag{2}$$

where τ is the threshold value. This reduces the effect of outliers and missing edges.

Chamfer matching as proposed by Barrow *et al.* [2] requires a good initialization of the template. In the hierarchical chamfer matching algorithm [5], candidate template locations are found using by hierarchical search using a resolution pyramid of the image. Subsequently an aligning transform for these candidate matches is estimated. Multiple templates are used to find three dimensional objects in an image [9, 15]. In our experiments we use templates which are generated by projecting a 3D hand model.

After the detection step, the best matching model is aligned by estimating the intrinsic parameters of this 3D model. Levenberg-Marquardt optimization is used for alignment, as described in [8], using the same chamfer cost function in the transformation step as in the search step of the algorithm.

3. Problems With Methods in Clutter

There are, however, problems with the techniques in the presence of background clutter, which are described in the following section.

3.1. Shape Context

It turns out that using the shape context in cluttered scenes is unreliable. It is difficult to recover the scale parameter, since normalizing the radial distances by the mean or median point distances will no longer work. Object and non-object points close to the object are hard to distinguish on the basis of their shape context alone. Points which are close to each other on the model shape are often matched to points which are far away from each other in the image. The iterative nature of the algorithm may sometimes be able to compensate for this shortcoming, improving the point correspondences in each step. Another approach could be, if some of the correspondences are correct, to identify outliers in the alignment phase of the iteration process using a robust estimation scheme, e.g. RANSAC. Outliers can then be excluded from the next shape context computation. However, shape deformations cannot be handled easily this way.

3.2. Chamfer Matching

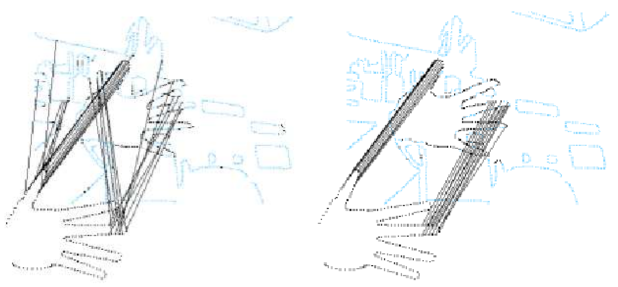
When using a single template, chamfer matching cannot handle large shape variations. The chamfer distance is not invariant towards translation, rotation or scale. Furthermore, the number of templates needed increases with object complexity. Each of these cases has to be handled by matching with different templates. In order to match a large number of templates efficiently, tree-based search methods have been suggested, where a large number of hypotheses can be eliminated at an early stage [9]. In scenes with cluttered background the chamfer cost function (2) will typically have several local minima. In order to make a decision about the object location, orientation and scale, it may be necessary to use a subsequent verification stage [9].

4. Proposed Improvements for Shape Context Matching

This section describes two methods of improving the robustness of point matching using shape contexts.

4.1. Using Edge Orientation

According to [9, 15] multiple feature images can be used, by dividing edge points into discrete sets based on the edge orientation. The same idea can be applied to the shape context by only matching points with similar gradient orientation. Figure 2 shows an example of estimating point correspondences when using single versus multiple features. Using multiple edge features increases the discrimination power of the shape context, and generally leads to improved results. However, as can be seen in figure 2, also with multiple features, incorrect matches can occur (points on middle finger are mapped to ring finger). Note that in cases where



(a) (b)

Figure 2: **Shape context matching is improved by using edge orientation.** *Finding correspondences by (a) matching all edge points (original method), (b) matching edge points with similar gradient direction only. The lighter colored points are matched to 'dummy' points and are classified as outliers.*

the gradient direction can switch, e.g. darker/lighter background behind the object, the sign of the gradient vector should not matter when grouping the edges.

4.2. Shape Context With Figural Continuity

The shape context descriptor alone is not powerful enough to yield reliable point correspondences in cluttered scenes. We propose incorporating a continuity constraint in the correspondence estimation. The idea is that neighboring points on the model shape $\mathcal{U}$, $\mathbf{u}_i$ and $\mathbf{u}_j$, should map to points $\mathbf{v}_{\phi(i)}$ and $\mathbf{v}_{\phi(j)}$ on the target shape $\mathcal{V}$ which are also close to each other. The correspondences are denoted by a function ϕ which maps each model point index to the corresponding image point index. The cost function for ϕ is given by

$$C_\phi(\mathcal{U},\mathcal{V}) = C_{sc}(\mathcal{U},\mathcal{V}) + \lambda\, C_{cont}(\mathcal{U},\mathcal{V}) + \mu\, C_{curv}(\mathcal{U},\mathcal{V}) \tag{3}$$

where C_{sc} are the shape context costs, C_{cont} is a continuity cost term, C_{curv} is a curvature cost term and λ and μ are weighting parameters. The shape context costs are, as before, the sum of all individual point matching costs

$$C_{sc}(\mathcal{U},\mathcal{V}) = \sum_{i=1}^{n} C_{sc}(\mathbf{u}_i, \mathbf{v}_{\phi(i)}). \tag{4}$$

The continuity cost term should ensure that two points which are close on the model shape are close in the image. Assuming that $\mathbf{u}_i$ and $\mathbf{u}_{i-1}$ are neighboring points:

$$C_{cont}(\mathcal{U},\mathcal{V}) = \sum_{i=2}^{n} ||\mathbf{v}_{\phi(i)} - \mathbf{v}_{\phi(i-1)}|| \tag{5}$$

Algorithm 1 Viterbi Algorithm for Point Correspondences

1: Compute the shape context costs $C_{sc}(i,j) \quad \forall i,j$

2: **Initialization**
$C_\phi(1,j) = C_{sc}(1,j) \quad j = 1,...,m$

3: **Propagation**
For each model point $\mathbf{u}_i \quad i = 2,...,n$
For each feature point $\mathbf{v}_j \quad j = 1,...,m$
For $k = 1,...,m$

$$C^k_\phi(i,j) = C_\phi(i-1,k) + C_{sc}(i,j) + \lambda\,||\mathbf{v}_k - \mathbf{v}_j|| + \mu\,||\kappa(\mathbf{v}_k) - \kappa(\mathbf{u}_{i-1})|| \tag{7}$$

Compute the costs of assigning $\mathbf{u}_i$ to $\mathbf{v}_j$ as

$$C_\phi(i,j) = \min_k C^k_\phi(i,j)$$

Store a pointer to the previous correspondence index

$$P(i,j) = \operatorname*{argmin}_k C^k_\phi(i,j)$$

4: **Termination**
Assign the point with optimal costs to $\mathbf{u}_n$
$\phi(\mathbf{u}_n) = \min_j C_\phi(n,j) \quad j = 1,...,m$

5: **Optimal Path Backtracking**
Find the other correspondences by
$\phi(\mathbf{u}_i) = P(\phi(i, \mathbf{u}_{i+1})) \quad i = n-1,...,1$

The curvature cost term will have low costs if the corresponding points have similar curvature energy

$$C_{curv}(\mathcal{U},\mathcal{V}) = \sum_{i=2}^{n-1} ||\kappa(\mathbf{u}_i) - \kappa(\mathbf{v}_{\phi(i)})|| \tag{6}$$

where κ is the curvature energy at the point, $\kappa(\mathbf{u}_i) = ||\mathbf{u}_{i-1} - 2\mathbf{u}_i + \mathbf{u}_{i+1}||$. Finding the minimum of this cost function is generally expensive. However, in the case when an ordering of the model point is given, this function can be optimized using dynamic programming. Writing the possible point assignments into a matrix, we use the Viterbi algorithm to find a path through this matrix which minimizes the total cost for its correspondences (algorithm 1). Figure 3 shows the results of the proposed method compared to the original version (using only shape context and bi-partite matching). The figure shows matches after the first correspondence step (no aligning transformation has been applied). The matches found by the original method do not obey the continuity constraint, whereas correspondences found by Viterbi are clearly better. We use the same scale, obtained from the model shape, to compute all shape contexts. The method is therefore not scale invariant, but in practice can handle some degree of scale variation. The Viterbi approach does depend on there being contours that can be followed in edge images, which not always the case. In order to deal with discontinuous edges, a 'dummy' point is added to which model points are matched to, as long as there are no good edge point candidates. Other optimization methods could be used to minimize the cost function in equation 3, which do not rely on sequential contour following. The algorithm, as described here, is designed for the case when an ordering of the model points is given. If this is not the case, for example, when edges branch off or merge, the continuity term has to be modified, still ensuring that two points that are close on the model are also nearby in the image.

It is interesting to note that the continuity and curvature terms are similar to the energy terms used in active contour models [12]. The model used here can therefore be characterized as a snake with integrated shape information. In fact, the algorithm is independent of the particular shape descriptor.

5. Results

To compare the algorithms, we show results on two types of data, images of hands in cluttered scenes, and words on textured background.

5.1. Initializing a Hand Model

We use shape matching to locate a hand in an image and estimate a set of shape parameters of a 3D hand model [16]. This is the initialization step in a model-based hand tracker, where automatic initialization and adaptation to the user is required (see figure 4). The 15 model parameters to be estimated are translation and rotation in the image plane (3), scale (1), the angles between fingers and palm (5), the finger lengths (5) and a width parameter for all fingers (1). The user is required to hold the open hand parallel to the image plane. The problem of parameter estimation is underdetermined when using a single view, however, the method extends to multiple views, similar to [10]. The image feature points are Canny edges, the model points are the projected contours. For the results shown here, we do not use skin color information. Skin color classification with a low detection threshold could be used to remove some background clutter, however, this was not done in these experiments. In the case of shape context matching, correspondences are found only once. The parameters of the aligning 3D transformation are then estimated using Levenberg-Marquardt optimization. In the case of chamfer matching, the hand is first localized using a global coarse-to-fine search. The model is then aligned using a version of the ICP algorithm which employs Levenberg-Marquardt optimization (LM-ICP) [8]. The error function for both global search and alignment are defined using the chamfer distance. The discriminative power of the error function is enhanced by

Figure 4: **Hand localization and model alignment.** *The images show the initialization phase of a model-based tracker. The hand is located using chamfer matching and subsequently aligned by optimization. The right image shows the projected contours of the adapted model,*

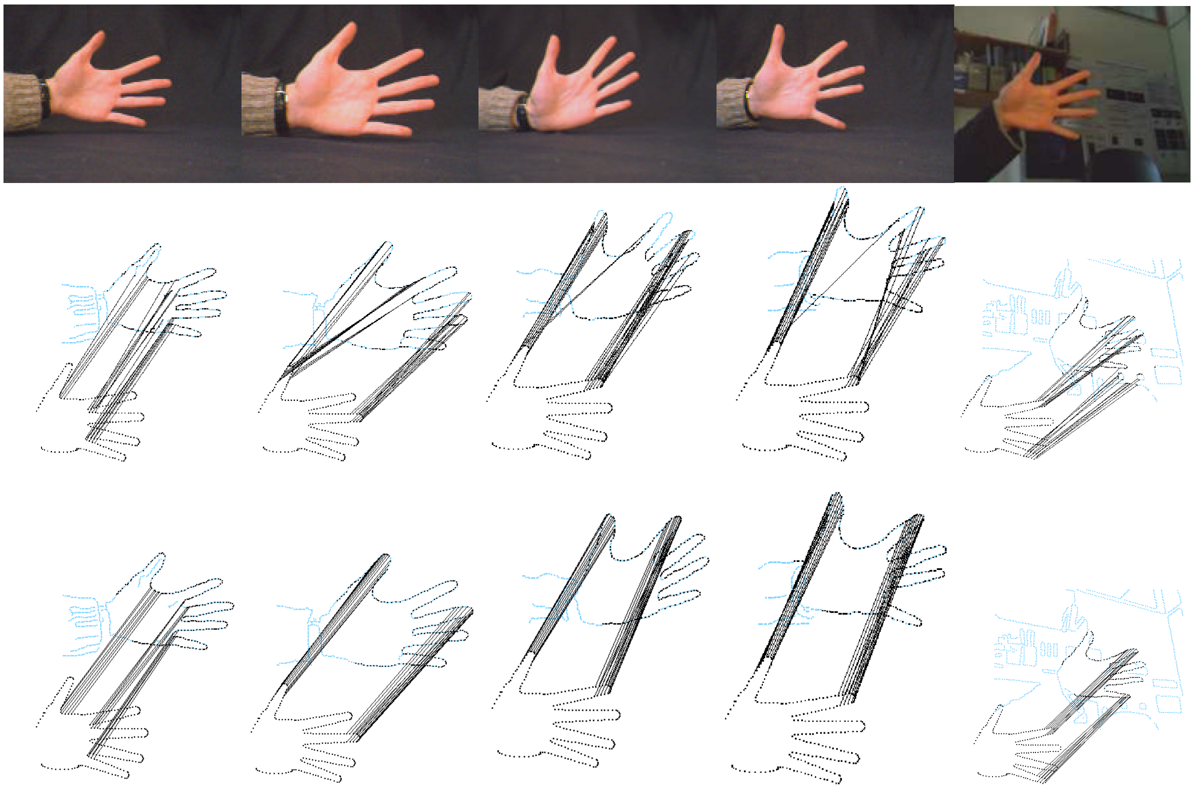
using multiple feature types based on edge orientation (discretized into 8 regions). The global search in image translation and scale space is done in a hierarchical fashion. For the results shown we use 147 templates, using 7 rotation angles in the image plane, 7 different scales and 3 shape variations.

Figure 5 shows results of hand localization experiments under a number of different lighting situations and with significant background clutter. It can be seen that matching using the shape context with continuity constraint (middle column) as well as chamfer matching (right column) give good results in the shown cases, whereas matching using the shape context alone (left column) does not work for other than relatively simple backgrounds. Figure 6 shows a typical failure mode of the shape context matching using Viterbi, while chamfer matching still produces reasonable results. The underlying reason for the failure is that when the shape context information is unreliable due to clutter or variations in scale and shape, the continuity constraint may not be able to compensate for this. In the example the edges in the background have a shape context similar to the model points of the thumb, and hence a wrong path along the contour is chosen.

The number of template points is about 200, the number of sampled edge points in the image is typically 1000-2000. The time until detection is approximately 10s for the original shape context version, 20s for Viterbi and 6s for chamfer matching (on a Pentium III, 1.0 GHz).

5.2. Word Recognition in Cluttered Images

We use chamfer matching for recognizing words in images generated by the *EZ-Gimpy* program [1]. These are word images (from a dictionary containing 561 words) corrupted with different types of image noise, deformations or background texture. Automatic recognition is made difficult specifically for the task to tell humans and computers apart [18]. Mori and Malik [14] have obtained a word recognition rate of 92.1% on 191 EZ-Gimpy images. Their



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

Figure 3: **Shape context matching is improved by using shape continuity.** *Correspondences found between edge points and points on hand template in the cases of (a) correct scale, (b) scale variation, (c) rotation, (d) shape variation, (e) background clutter. Top: input images, middle: results of the original method, using shape context and bi-partite matching, bottom: results of shape context combined with continuity constraint, computed with Viterbi. The results shown are the correspondences found after a single matching step without iteration. Even in relatively simple cases the original shape context method finds many wrong matches, whereas the introduction of the continuity constraint leads to improved correspondences.*

method is based on matching letters using shape contexts and thin plate spline transformations. This method has previously been applied successfully to the clutter free MNIST data set of handwritten digits [3].

We conduct two experiments, using one and two templates per letter. As a preprocessing step, the images are first binarized using simple thresholding. We compute the symmetric chamfer cost between templates and image at each location in an exhaustive manner. Local cost minima are hypothesized letter locations. The symmetric chamfer cost is used because it allows discrimination between two letters, where one letter shape is part of the other one, for example 'o' and 'p'. In contrast to section 5.1, no further optimization is performed. Subsequently we compute the matching cost for each word in the dictionary. We also use the knowledge that the letter distance within a word has low variance. The word matching cost is the average symmetric chamfer distance of the letters and the variance of letter distances in the x and y direction. On the same test set used by Mori and Malik, we obtain a recognition rate of 89.5% when using one template and 93.2% when using two templates per letter (an additional sheared version of each template). Figure 7 shows examples of matching results. Images with an incorrect top match, are mostly distorted by a 'whirl' or 'wave' transformation (last two rows in figure 7). A further optimization step may improve the results.

6. Summary and Conclusions

We have presented an empirical study of two different methods for object localization from edges in cluttered scenes – shape context and chamfer matching. The results demonstrate that the original shape context algorithm fails in heavily cluttered scenes, where it is no longer robust towards variations in scale or rotation. By including contour continuity and curvature information, similar to those used in active contour models, it is possible to obtain significantly better correspondences and model alignment results. If a point ordering is given for the model, the joint cost function can be optimized using the Viterbi algorithm.

When using the same number of templates, shape context matching can handle larger shape variations than chamfer matching. However, when shape context matching fails, the incorrect correspondences often lead to bad alignment, and subsequent optimization fails to find the the correct transformation. Failure cases in chamfer matching are mainly due to false positive matches during the global search phase. The results may be improved by including a hypothesis verification step. Our experiments have shown that chamfer matching is more robust in clutter than shape context matching, even with the suggested improvements.

Acknowledgments

The authors would like to thank the Gates Cambridge Trust, the Overseas Research Scholarship Programme, the Gottlieb Daimler–and Karl Benz–Foundation, and the EPSRC for their support.

References

[1] available at http://www.captcha.net.

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intell.*, 24(4):509–522, April 2002.

[4] P. J. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Analysis and Machine Intell.*, 14(2):239–256, March 1992.

[5] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intell.*, 10(6):849–865, November 1988.

[6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.

[7] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 2003. in press.

[8] A. Fitzgibbon. Robust registration of 2D and 3D point sets. In *Proc. British Machine Vision Conference*, volume II, pages 411–420, Manchester, UK, September 2001.

[9] D. M. Gavrila. Pedestrian detection from a moving vehicle. In D. Vernon, editor, *Proc. 6th European Conf. on Computer Vision*, volume II, pages 37–49, Dublin, Ireland, June/July 2000.

[10] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, June 1996.

[11] S. Gold, A. Rangarajan, C.-P. Lu, and E. Mjolsness. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, London, June 1987.

[13] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. 7th European Conf. on Computer Vision*, volume III, pages 666–680, Copenhagen, Denmark, May 2002.

[14] G. Mori and J. Malik. Recognizing objects in adversarial clutter – breaking a visual captcha. In *Proc. Conf. Computer Vision and Pattern Recognition*, Madison, USA, June 2003. in press.

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

[16] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 310–315, Kauai, USA, December 2001.

[17] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *Proc. 7th European Conf. on Computer Vision*, volume I, pages 629–644, Copenhagen, Denmark, May 2002.

[18] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart (automatically) or how lazy cryptographers do AI. Technical Report TR CMU-CS-02-117, Carnegie Mellon University, Pittsburgh, PA, February 2002.

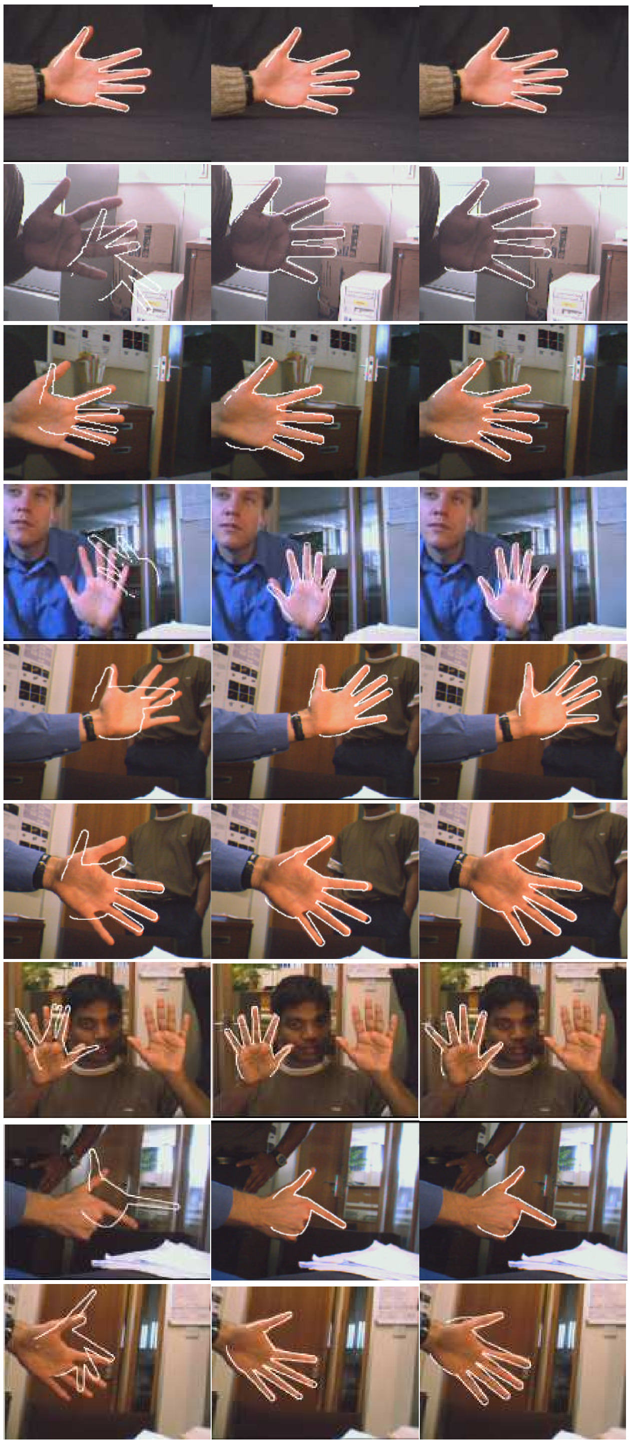
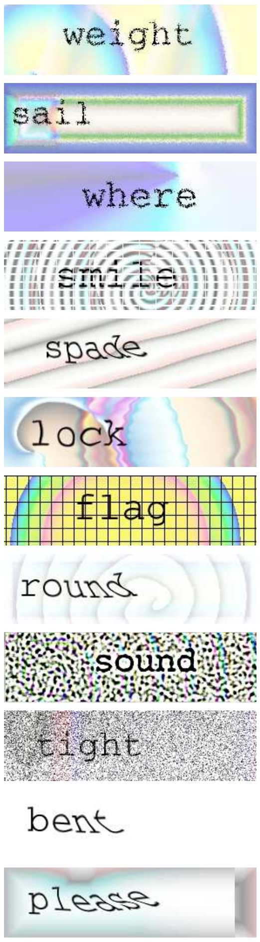


Figure 5: **Results of hand localization.** *Left column: hand localization using shape context information only (original algorithm), middle column: shape context with continuity constraint, right column: chamfer matching and LM-ICP.*

(a) (b) (c)

Figure 6: **Failure case for shape context matching.** *(a) Edge points and model points. Edge points matched using Viterbi are black, (b) alignment using Viterbi, (c) using chamfer matching.*



1. weight	(0.53)	4. sign	(0.94)
2. night	(0.74)	5. fight	(1.00)
3. tight	(0.89)	6. high	(1.05)
1. sail	(0.59)	4. tail	(0.88)
2. nail	(0.77)	5. wall	(1.00)
3. rail	(0.86)	6. tall	(1.01)
1. where	(0.75)	4. waste	(1.12)
2. here	(0.87)	5. when	(1.13)
3. horn	(1.10)	6. hate	(1.17)
1. smile	(1.00)	4. solid	(1.74)
2. sail	(1.64)	5. knife	(1.82)
3. while	(1.65)	6. nail	(1.86)
1. spade	(1.41)	4. send	(1.97)
2. shade	(1.50)	5. sand	(1.99)
3. trade	(1.83)	6. road	(2.05)
1. lock	(1.07)	4. foot	(1.87)
2. look	(1.18)	5. fork	(1.93)
3. loss	(1.74)	6. book	(2.02)
1. flag	(1.62)	4. fish	(1.86)
2. flat	(1.72)	5. true	(1.89)
3. free	(1.81)	6. from	(1.89)
1. round	(1.30)	4. roof	(1.82)
2. sound	(1.59)	5. moon	(2.05)
3. wound	(1.64)	6. road	(2.07)
1. sound	(0.95)	4. young	(1.35)
2. wound	(1.12)	5. soup	(1.35)
3. round	(1.20)	6. south	(1.37)
1. fight	(1.59)	4. light	(1.70)
2. right	(1.64)	5. fish	(1.89)
3. tight	(1.68)	6. debt	(1.90)
1. bank	(1.71)	4. bent	(1.94)
2. bath	(1.90)	5. bone	(1.95)
3. back	(1.92)	6. book	(1.96)
1. blood	(2.23)	4. clear	(2.36)
2. slope	(2.33)	5. clean	(2.37)
3. clock	(2.36)	6. tired	(2.43)

Figure 7: **Word recognition results.** *Examples of recognized words in* EZ-Gimpy *images. The top six matches for each word and their cost are shown.*