

# Multi-feature Hierarchical Template Matching Using Distance Transforms

D.M. Gavrilu  
Daimler-Benz AG, Research and Technology  
Wilhelm Runge St. 11  
89081 Ulm, Germany  
gavrila@dbag.ulm.DaimlerBenz.com

## Abstract

*We describe a multi-feature hierarchical algorithm to efficiently match  $N$  objects (templates) with an image using distance transforms (DTs). The matching is under translation, but it can cover more general transformations by generating the various transformed templates explicitly. The novel part of the algorithm is that, in addition to a coarse-to-fine search over the translation parameters, the  $N$  templates are grouped off-line into a template hierarchy based on their similarity. This way, multiple templates can be matched simultaneously at the coarse levels of the search, resulting in various speed-up factors. Furthermore, in matching, features are distinguished by type and separate DT's are computed for each type (e.g. based on edge orientations). These concepts are illustrated in the application of traffic sign detection.*

## 1 Introduction

Matching is a central problem in pattern recognition and computer vision. A common application is object detection and tracking. The various matching methods that have been proposed can be distinguished by what type of features are used [12]. At the one end there are pixel-based methods, which fit models directly to (filtered) image pixels. At the other end there are symbolic matching methods which operate on a few high-level features (e.g. parts of objects and their relations) and apply graph matching methods to establish correspondence.

In this paper, we consider methods for image matching using distance transforms (DTs). Matching using DTs involves intermediate-level features [2] which are extracted locally at various image locations, e.g. edge points. A DT converts the binary image, which consists of feature and non-feature pixels, into a DT image where each pixel denotes the distance to the nearest feature pixel. Similarly, the object of interest is represented by a binary template using the same fea-

ture representation. Matching proceeds by correlating the template against the DT image; the correlation value is a measure of similarity in image space.

Previous work on DT-based matching [1] [2] [7] [3] [11] [5] [10] [6] has dealt with the case of matching one template against an image, allowing certain geometrical transformations (e.g. translation, rotation, affine). Here we consider a more general case of matching  $N$  templates with an image under translation. Matching of one template under more general transformations can be seen as a special case when all the transformed templates are generated explicitly. In addition to a coarse-to-fine search over the translation parameters, the  $N$  templates are grouped off-line into a template hierarchy based on their similarity. Multiple templates can be matched simultaneously at the coarse levels of the search, resulting in various speed-up factors.

The outline of the paper is as follows. Section 2 reviews previous work on distance transforms, distance measures and matching strategies. Section 3 discusses the proposed extensions to the DT matching scheme, which involve the use of multiple features and an efficient match strategy by means of a template hierarchy. Section 4 lists experiments in the application of traffic sign detection. Finally, we conclude in Section 5.

## 2 Previous Work

### 2.1 Distance Transforms

A distance transform (DT) converts a binary image, which consists of feature and non-feature pixels, into an image where each pixel value denotes the distance to the nearest feature pixel. DTs approximate global distances by propagating local distances at image pixels. Particular DT algorithms depend on a variety of factors. One factor is whether they result in a Euclidean distance metric or not (EDTs vs. WDT) [8] [13]. Figure 1 illustrates a EDT. WDTs define vari-

ous approximations of the “true” Euclidean distance measure. One such approximation is the *chamfer-2-3* metric [1] [2] [13], used in our experiments. Another factor is how the distances are

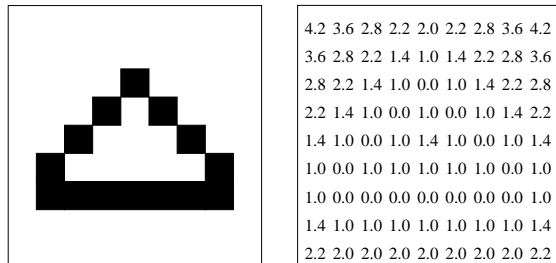


Figure 1: A binary pattern and its Euclidean Distance Transform

propagated over the image, whether in a raster scan or a contour scan fashion. Most algorithms use a raster scan fashion where the propagation of distances is in a manner independent of the feature locations in the image, with a mask of fixed size and shape. Contour scan algorithms propagate the distances from the feature locations. Some DT approaches also weigh the distances from features by their salience, where salient features (e.g. edge strength, length, curvature) result in comparably lower “distance” values [10]. Finally, there are sequential and parallel DT algorithms [4].

## 2.2 Match Measures and Strategies

Matching with DT is illustrated schematically in Figure 2. It involves two binary images, a segmented template  $T$  and a segmented image  $I$ , which we’ll call “feature template” and “feature image”. The “on” pixels denote the presence of a feature and the “off” pixels the absence of a feature in these binary images. What the actual features are, does not matter for the matching method. Typically, one uses edge- and corner-points. The feature template is given off-line for a particular application, and the feature image is derived from the image of interest by feature extraction.

Matching  $T$  and  $I$  involves computing the distance transform of the feature image  $I$ . The template  $T$  is transformed (e.g. translated, rotated and scaled) and positioned over the resulting DT image of  $I$ ; the matching measure  $D(T, I)$  is determined by the pixel values of the DT image which lie under the “on” pixels of the template. These pixel values form a distribution of

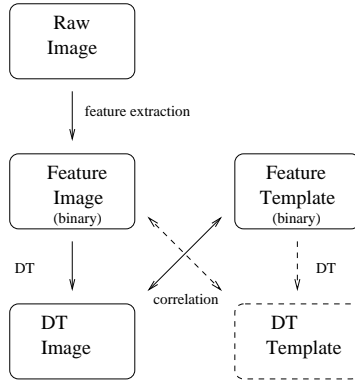


Figure 2: Matching using a DT

distances of the template features to the nearest features in the image. The lower these distances are, the better the match between image and template at this location. There are a number of matching measures that can be defined on the distance distribution. One possibility is to use the average distance to the nearest feature. This is the *chamfer* distance.

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t) \quad (1)$$

where  $|T|$  denotes the number of features in  $T$  and  $d_I(t)$  denotes the distance between feature  $t$  in  $T$  and the closest feature in  $I$ . The chamfer distance consists thus of a correlation between  $T$  and the distance image of  $I$ , followed by a division. Other more robust measures reduce the effect of missing features (i.e. due to occlusion or segmentation errors) by using the average truncated distance or the  $f$ -th quantile value (the *Hausdorff* distance) [7] [11]. In applications, a template is considered matched at locations where the distance measure  $D(T, I)$  is below a user-supplied threshold  $\theta$

$$D(T, I) < \theta \quad (2)$$

Figure 3 illustrates the matching scheme of Figure 2 for the typical case of edge features. Figure 3a-b shows an example image and template. Figure 3c-d shows the edge detection and DT transformation of the edge image. The distances in the DT image are intensity-coded; lighter colors denote increasing distance values.

The advantage of matching a template (Figure 3b) with the DT image (Figure 3d) rather than with the edge image (Figure 3c) is that the resulting similarity measure will be more smooth

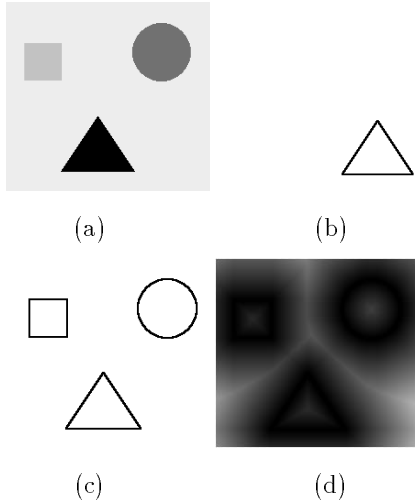


Figure 3: (a) original image (b) template (c) edge image (d) DT image

as a function of the template transformation parameters. This enables the use of various efficient search algorithms to lock onto the correct solution, as will be discussed shortly. It also allows more variability between a template and an object of interest in the image. Matching with the unsegmented (gradient) image, on the other hand, typically provides strong peak responses but rapidly declining off-peak responses.

A number of extensions have been proposed to the basic DT matching scheme. Some deal with hierarchical approaches to improve match efficiency and use multiple image resolutions [2]. Others use a pruning [3] [7] or a coarse-to-fine approach [11] in the parameter space of relevant template transformations. The latter approaches take advantage of the smooth similarity measure associated with DT-based matching; one need not to match a template for each location, rotation or other transformation. Other extensions involve the use of a un-directed (“symmetric”) similarity measure between image and a template [7] [5]. In this case, a DT is applied on both the image and template. Matching takes place with the feature image and feature template, vice versa, as seen in Figure 2.

Here is a summary of various aspects covered in past work on DT-based matching

- **features:** edge points, corner points
- **multi-typing:** none
- **distance metric:** chamfer-2-3, chamfer-3-4, Euclidean

- **computation of DT image:** serial vs. parallel, salience weighing
- **match measures:** Euclidean vs. robust measures, directed vs. undirected measures
- **matching N templates:** none
- **global search algorithms:** exhaustive vs. hierarchical (in transformation space, in image resolution)

### 3 Extensions

#### 3.1 Multiple Feature-Types: Edge Orientation

So far, no distinction has been made regarding the type of features. All features would appear in one feature image (or template), and subsequently, in one DT image. If there are several feature types, and one considers the match of a template at a particular location of the DT image, it is possible that the DT image entries reflect shortest distances to features of non-matching type. The similarity measure would be too optimistic, increasing the number of false positives one can expect from matching.

A simple way to take advantage of possibility to distinguish feature types is to use separate feature-images and DT images, for each type. Thus having  $M$  distinct feature types results in  $M$  feature images and  $M$  DT images. Similarly, the “untyped” feature template is separated in  $M$  “typed” feature templates. Matching proceeds as before, but now the match measure between image and template is the sum of the match measures between template and DT image of the same type.

We now consider the frequent case of the use of edge points as features. For this case, we propose the use of edge orientation as feature type by partitioning the unit circle in  $M$  bins

$$\left\{ \left[ \frac{i}{M}2\pi, \frac{i+1}{M}2\pi \right] \mid i = 0, \dots, M-1 \right\} \quad (3)$$

Thus a template edge point with edge orientation  $\psi$  is assigned to the typed template with index

$$\left\lfloor \frac{\psi}{2\pi}M \right\rfloor \quad (4)$$

We still have to account for measurement error in the edge orientation and the tolerance we’ll allow between the edge orientation of template and image points during matching. Let the absolute measurement error in edge orientation of the

template and image points be  $\Delta\phi_T$  and  $\Delta\phi_I$ , respectively. Let the allowed tolerance on the edge orientation during matching be  $\Delta\phi_{tol}$ . In order to account properly for these quantities, a template edge point is assigned to a range of typed templates, namely those with indices

$$\{ \lfloor \frac{(\psi - \Delta\phi)}{2\pi} M \rfloor, \dots, \lfloor \frac{(\psi + \Delta\phi)}{2\pi} M \rfloor \} \quad (5)$$

mapped cyclically over the interval  $0, \dots, M - 1$ , with

$$\Delta\phi = \Delta\phi_T + \Delta\phi_I + \Delta\phi_{tol} \quad (6)$$

For applications where there is no sign information associated with the edge orientation, a template edge point is also assigned to the typed templates one obtains by substituting  $\psi + \pi$  for  $\psi$  in Equation (5).

### 3.2 Matching $N$ Templates: Template Hierarchy

One often encounters the problem of matching  $N$  templates with an image. If the  $N$  templates bear no relationship to each other, there is little one can do better than match each of the templates separately. If there is some structure in the template distribution, one can do better. The proposed scheme to match the  $N$  related templates involves the use of a template hierarchy, in addition to a coarse-to-fine search over the image. The idea is that at a coarse level of search, when the image grid size of the search is large, it would be inefficient to match each of the  $N$  objects separately, if they are relatively similar to each other. Instead, one would group similar templates together and represent them by a prototype template; matching would be done with this prototype, rather than with the individual templates, resulting in a (potentially significant) speed-up. This grouping of templates is done at various levels, resulting in a hierarchy, where at the leaf levels there are the  $N$  templates one needs to match with, and on intermediate levels there are the prototypes.

To make matters more concrete, consider first the case of a coarse-to-fine search where one matches a single template under translation. Assume there are  $L$  levels of search ( $l = 1, \dots, L$ ), determined by the size  $\sigma_l$  of the underlying uniform grid and the distance threshold  $\theta_l$  which determines when a template matches sufficiently enough to consider matching on a finer grid (in the neighborhood of the promising solution). Let  $\tau_{tol}$  denote the allowed tolerance on the distance

measure between template and image at a ‘‘correct’’ location. Let  $\mu$  denote the distance along the diagonal of a unit grid element. Then by having

$$\theta_l = \tau_{tol} + \frac{1}{2}\mu\sigma_l \quad (7)$$

one has the desirable property that, using untruncated distance measures such as the chamfer distance, one can assure that the coarse-to-fine approach will not miss a solution. The second term accounts for the (worst) case that the solution lies at the center of the 4 enclosing grid points which form a square.

Now consider the case where the above  $L$ -level search is combined with a  $L$ -level template hierarchy. Matching can be seen as traversing the tree structure of templates. Each node corresponds to matching a (prototype) template  $\mathbf{p}$  with the image at node-specific locations. For the locations where the distance measure between template and image is below user-supplied threshold  $\theta_p$ , one computes new interest locations for the children nodes (generated by sampling the local neighborhood with a finer grid) and adds the children nodes to the list of nodes to be processed. The matching process starts at the root, the interest locations lie initially on a uniform grid over relevant regions in the image. The tree can be traversed in breadth-first or depth-first fashion. In the experiments, we use depth-first traversal which has the advantage that one needs to maintain only  $L - 1$  sets of interest locations.

Let  $\mathbf{p}$  be the template corresponding to the node currently processed during the traversal and let  $C = \{\mathbf{t}_1, \dots, \mathbf{t}_c\}$  be the set of templates corresponding to its children nodes. Let  $\delta_p$  be the maximum distance between  $\mathbf{p}$  and the elements of  $C$ .

$$\delta_p = \max_{\mathbf{t}_i \in C} D(\mathbf{p}, \mathbf{t}_i) \quad (8)$$

Then by having

$$\theta_p = \tau_{tol} + \delta_p + \frac{1}{2}\mu\sigma_l \quad (9)$$

one has the desirable property that, using untruncated distance measures such as the chamfer distance, one can assure that the coarse-to-fine approach using the template hierarchy will not miss a solution. The thresholds one obtains by Equation (9) are quite conservative, in practice one can use lower thresholds to speed up matching, at the cost of possibly missing a solution (see Experiments).

## 4 Experiments

To illustrate the proposed matching method we apply it to the detection of circular and triangular (up/down) signs, as seen on highways and secondary roads. For the moment, we do not consider traffic signs which appear tilted and/or skewed in the image; the only shape parameter considered is scale. Edge points are used as features, further differentiated by their edge orientation. The edge orientations are discretized in 8 values. We use templates for circles and triangles with radii in the range of 7-18 pixels (the images are of size 360 by 288 pixels). This leads to a total of 36 templates, for which a template tree is specified “manually” as in Figure 4. The tree has three levels (not counting the root level, which contains no template). The root node has six children corresponding to two prototypes for each of the three main shapes to be matched: circle, triangle-up, triangle-down. The prototypes at the first level of the hierarchy are simply the templates with radii equal to the median value of intervals [7-12] and [13-18], namely 9 and 15. The prototypes at the second level are the templates with radii equal to the median value of intervals [7-9], [10-12], [13-15] and [16-18]. Each template (or prototype) is partitioned into 8 typed templates based on edge orientation. The direction of the edge orientation is specified, we only search for circles and triangles with a “light-inside-dark-outside” contour characteristic.

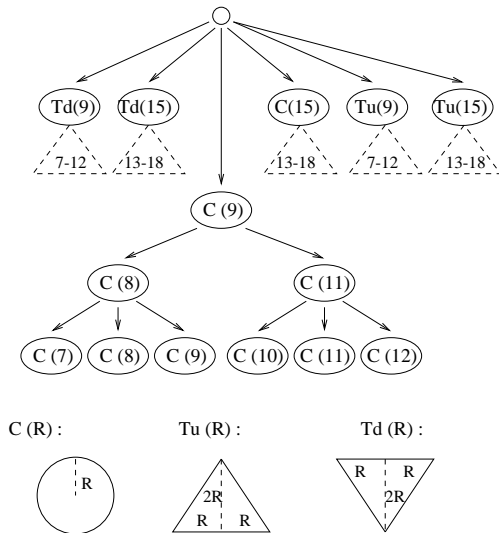


Figure 4: Template hierarchy

Matching uses a depth-order traversal over the template tree, in the manner described by

Subsection 3.2. Coarse-to-fine sampling uses a grid size of  $\sigma = 8, 4, 1$  for the three levels of the template tree. We used distance thresholds  $\theta_i = 3.5, 1.35, 0.6$  pixels for the three levels, respectively.

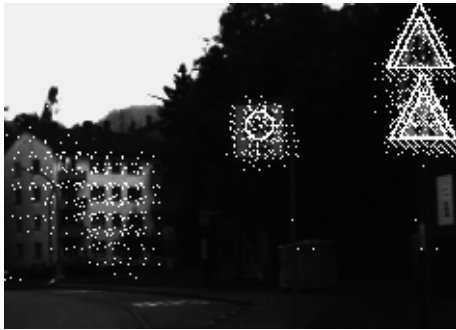
The experiments involved both off- and online tests. Off-line, we used a database of 1000 images, taken during day-time (sunny, rainy) and night-time. We obtained single-image detection rates of about 90%, when allowing solutions to deviate by 2 pixels and by radius 1 from the values obtained by a human. Typically, there were 4-6 false positives per image (in a later pictograph classification phase, more than 95% of these were rejected using a RBF network). Figure 5 illustrates the followed hierarchical approach. The white dots indicate locations where the match between image and a (prototype) template of the template tree was good enough to consider matching with more specific templates (the children), on a finer grid. The final detection result is also shown. More detection results are given in Figure 6, including some false positives. The traffic signs of the database that were not detected were had low contrast, were tilted or skewed. Improvement of the detection rate can thus be achieved in a relative straightforward manner, by lowering the edge threshold and by adding more templates.

Given image width  $W$ , image height  $H$ , and  $K$  templates, a non-hierarchical matching algorithm would require  $W \times H \times K$  correlations between template and image. In the presented hierarchical approach both factors  $W \times H$  and  $K$  are pruned (by a coarse-to-fine approach in image space and in template space). It is not possible to provide an analytical expression for the speed-up, because it depends on the actual image data and template distribution. Typically, we have observed speed-up factors in the range of 200-400.

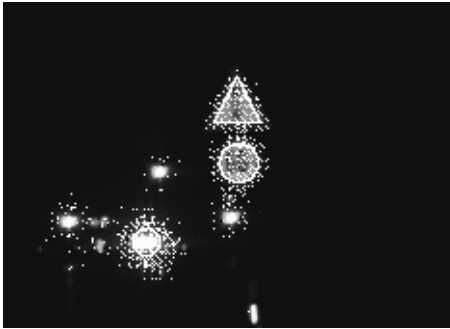
## 5 Conclusion

In this paper we proposed two extensions to DT-based matching. The first extension dealt with differentiating the features by type (i.e. by edge orientation) and the second dealt with matching using a template hierarchy. We observed that this approach can result in a significant speed-up when compared to the exhaustive approach, in the order of two magnitudes. Some interesting problems lie ahead regarding the automatic

generation of the template hierarchy.



(a)



(b)

Figure 5: Traffic sign detection: (a) day and (b) night (white dots denote intermediate results; the locations matched during hierarchical search)

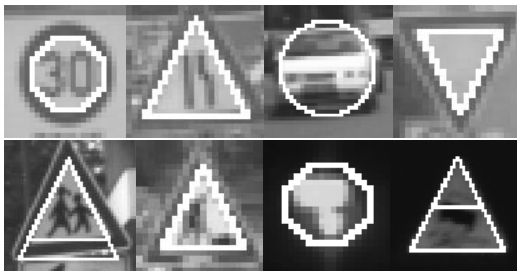


Figure 6: More detection results

## References

- [1] H. Barrow et al. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
- [2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.
- [3] G.E. Ford D.W. Paglieroni and E.M. Tsujimoto. The position-orientation masking approach to parametric search for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):740–747, 1994.
- [4] H. Embrechts and D. Roose. A parallel euclidean distance transformation algorithm. *CVIU*, 63(1):15–26, January 1996.
- [5] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, 1996.
- [6] D. Huttenlocher. Monte carlo comparison of distance transform based matching measures. In *ARPA Image Understanding Workshop*, pages 1179–1183, 1997.
- [7] D. Huttenlocher, G. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [8] F. Leymarie and Martin D. Levine. Fast raster scan distance propagation on the discrete rectangular lattice. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 55(1):84–94, January 1992.
- [9] D. Mumford. Mathematical theories of shape: Do they model perception? In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 2–10, 1991.
- [10] P.L. Rosin and G.A.W. West. Saliency distance transforms. *GMIP*, 57(6):483–521, November 1995.
- [11] W. Rucklidge. Locating objects using the hausdorff distance. In *International Conference on Computer Vision*, pages 457–464, 1995.
- [12] P. Suetens, P. Fua, and A. Hanson. Computational strategies for object recognition. *ACM Computing Surveys*, 24(1):6–61, 1992.
- [13] E. Thiel and A. Montanvert. Chamfer masks: Discrete distance functions, geometrical properties, and optimization. In *International Conference on Pattern Recognition*, pages 244–247, The Hague, 1992.