in Proc. IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany, 1998

REAL-TIME OBJECT DETECTION USING DISTANCE TRANSFORMS

D.M. Gavrila

Image Understanding Systems
Daimler-Benz Research
Ulm 89081, Germany
gavrila@dbag.ulm.DaimlerBenz.com

V. Philomin

Computer Vision Laboratory
University of Maryland
College Park, MD 20742, U.S.A.
vasi@umiacs.umd.edu

ABSTRACT

This paper presents an efficient shape-based object detection method using Distance Transforms (DTs). The proposed method extends previous DT-based matching techniques by using multiple features and a template hierarchy associated with a coarse-to-fine search over the template transformation parameters. Significant speed-up factors are typically obtained when comparing the proposed hierarchical method with an equivalent brute-force technique; we have measured speed-up gains in the order of two magnitudes. This brings a number of template matching applications which previously required special-purpose correlation hardware onto the realm of the ubiquitous PC. We present results on real-time traffic sign detection to illustrate our approach.

1. INTRODUCTION

A concerted effort is currently underway at Daimler-Benz to extend vision-based navigation beyond the highway scenario into the complex urban environment; for an overview see [3]. One part of this effort is dedicated to the detection and recognition of relevant objects in urban traffic (e.g. road marks, traffic signs, vehicles, pedestrians). Various vision cues were previously used for object detection: object motion [6], color [8] and depth [4]. This paper presents a method which uses shape cues for object detection; it is based on an efficient application of template matching and 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. DTs approximate global distances by propagating local distances at image pixels. The object of interest is represented by a binary template using the same feature representation as the scene image. Matching proceeds by correlating the template against the DT image; the correlation value is a measure of similarity in image space.

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 efficent 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

Matching with DT is illustrated schematically in Figure 1. 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 transformed template. These pixel values form a distribution of 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 mea-

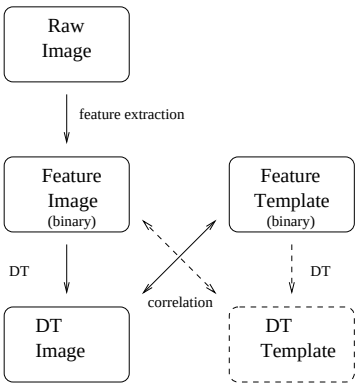


Figure 1: Matching using a DT

sures 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.

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

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. Thus, the chamfer distance consists 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] [10].

In applications, a template is considered matched at locations where the distance measure $D(T, I)$ is below a user-supplied threshold θ

$$D(T, I) < \theta \tag{2}$$

Figure 2 illustrates the matching scheme of Figure 1 for the typical case of edge features. Figure 2a-b shows an example image and template. Figure 2c-d shows the edge detection and DT transformation of the edge image. The distances in the DT image are intensity-coded; lighter colors denote larger distance values.

The advantage of matching a template (Figure 2b) with the DT image (Figure 2d) rather than with the edge image (Figure 2c) is that the resulting similarity measure will be more smooth as a function of the template transformation parameters. This enables the use of various efficent 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 [9] [7] or a coarse-to-fine approach [10] 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 [5] [7]. In this case, a DT is applied on both the image and template. Matching takes places with the feature image and feature template, vice versa, as seen in Figure 1.

Previous work on DT-based matching [1] [2] [5] [7] [9] [10] has dealt with the case of matching one template against an image, allowing certain geometrical transformations (e.g. translation, rotation, affine). In Subsection 3.2 we will consider the 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.

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

$$\{ [\frac{i}{M}2\pi, \frac{i+1}{M}2\pi] \, | i = 0, ..., M-1 \} \tag{3}$$

Thus a template edge point with edge orientation ψ is assigned to the typed template with index

$$\lfloor \frac{\psi}{2\pi} M \rfloor \tag{4}$$

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

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

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

$$\Delta\phi = \Delta\phi_T + \Delta\phi_I + \Delta\phi_{tol} \tag{6}$$

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 ψ 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, however, 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 ($l = 1, ..., L$), determined by the size σ_l of the underlying uniform grid and the distance threshold θ_l which determines when a template matches sufficiently enough to consider matching on a finer grid (in the neighborhood of the promising solution). Let τ_{tol} denote the allowed tolerance on the distance measure between template and image at a "correct" location. Let μ denote the distance along the diagonal of a unit grid element. Then by having

$$\theta_l = \tau_{tol} + \frac{1}{2}\mu\sigma_l \tag{7}$$

one has the desirable property that, using un-truncated 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 θ_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 $C = \{\mathbf{t}_1, ..., \mathbf{t}_c\}$ be the set of templates corresponding to its

children nodes. Let δ_p be the maximum distance between $\mathbf{p}$ and the elements of C.

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

Then by having

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

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 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 3. 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]. The prototypes at the first two levels were sub-sampled. Furthermore, each template (or prototype) is partitioned into 8 typed templates based on edge orientation (or 4, if the sign of the edge orientation is unspecified). 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 $\theta_l = 3.5, 1.35, 0.6$ 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 over 90%, when allowing solutions to deviate by 2 pixels and by radius 1 from the values obtained by a human. On the average, there was one false positive per image (in a later verification phase, more than 95% of these were rejected using a pictograph classifier). Figure 4 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 5, including some false positives The traffic signs in the database that were not detected 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. On-line experiments were performed with our E-class T-model vehicle (see Figure 6). The detection system runs currently at about 8-10 Hz using the on-board dual-Pentium II 333 Mhz.

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 200-300.

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.

6. ACKNOWLEDGEMENT

We would like to thank Larry Davis for his constructive feedback on this work.

7. 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] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Pätzhold, and C. Wöhler. Autonomous driving approaches downtown. *to appear in IEEE Expert (special issue on Vision-based Driving Assistance in Vehicles of the Future)*, 1997.

[4] U. Franke and I. Kutzbach. Fast stereo-based object detection for stop & go traffic. In *Proc. of Intelligent Vehicles Conference*, pages 339–344, Tokio, 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] B. Heisele and C. Woehler. Motion-based recognition of pedestrians. In *International Conference on Pattern Recognition*, 1998.

[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.

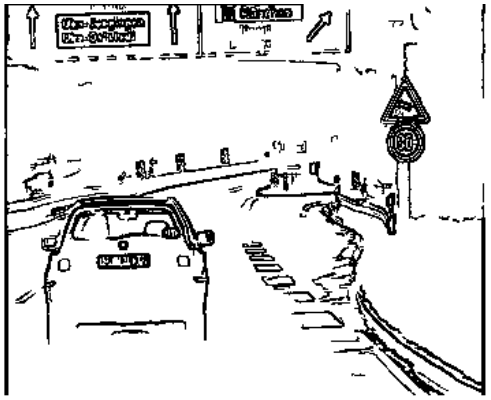
[8] R. Janssen, W. Ritter, F.Stein, and S. Ott. Hybrid approach for traffic sign recognition. In *Proc. of Intelligent Vehicles Conference*, pages 390–395, 1993.

[9] D.W. Paglieroni, G.E. Ford, 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.

[10] W. Rucklidge. Locating objects using the hausdorff distance. In *International Conference on Computer Vision*, pages 457–464, 1995.

(a)

(b)



(c)

(d)

Figure 2: (a) original image (b) template (c) edge image (d) DT image

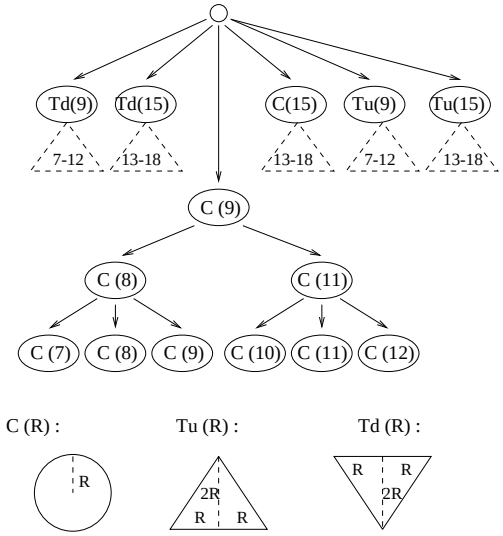
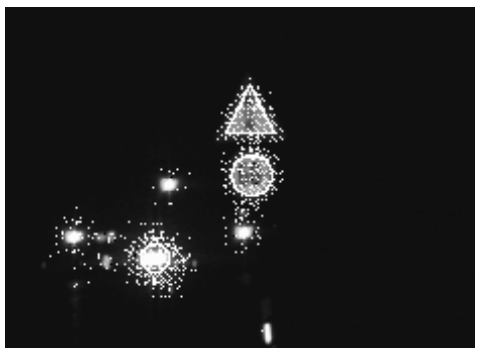


Figure 3: Template hierarchy

(a)



(b)

Figure 4: Traffic sign detection: (a) day and (b) night (white dots denote intermediate results; the locations matched during hierarchical search)

Figure 5: More detection results

(a)

(b)

Figure 6: (a) Demo vehicle (b) on-board camera and display