AUTOMATIC HIERARCHICAL OBJECT DECOMPOSITION FOR OBJECT RECOGNITION

M. Ulrich [a, b,] *, A. Baumgartner [a], C. Steger [b]

[a] Chair for Photogrammetry and Remote Sensing, Technische Universität München, Arcisstr. 21,
80290 München, +49 89 289-22671, www.remotesensing.de,
(markus.ulrich, albert.baumgartner)@bv.tum.de
[b] MVTec Software GmbH, Neherstr. 1, 81675 München, +49 89 457695-0, www.mvtec.com,
(ulrich,steger)@mvtec.com

KEYWORDS: Object Recognition, Real-time, Hierarchical Model, Computer Vision, Industrial Application

ABSTRACT:

Industrial applications of 2D object recognition such as quality control often demand robustness, highest accuracy, and real-time computation from the object recognition approach. Simultaneously fulfilling all of these demands is a hard problem and has recently drawn considerable attention within the research community of close-range photogrammetry and computer vision. The problem is complicated when dealing with objects or models consisting of several rigid parts that are allowed to move with respect to each other. In this situation, approaches searching for rigid objects fail since the appearance of the model may substantially change under the variations caused by the movements. In this paper, an approach is proposed that not only facilitates the recognition of such parts-based models but also fulfills the above demands. The object is automatically decomposed into single rigid parts based on several example images that express the movements of the object parts. The mutual movements between the parts are analyzed and represented in a graph structure. Based on the graph, a hierarchical model is derived that minimizes the search effort during a subsequent recognition of the object in an arbitrary image.

1. INTRODUCTION

Object recognition is part of many computer vision applications. It is particularly useful for industrial inspection tasks, where often an image of an object must be aligned with a model of the object. The transformation (pose) obtained by the object recognition process can be used for various tasks, e.g., pick and place operations, quality control, or inspection tasks. In most cases, the model of the object is generated from an image of the object. Such pure 2D approaches are frequently used, because it usually is too costly or time consuming to create a more complicated model, e.g., a 3D CAD model. Therefore, in industrial inspection tasks one is typically interested in matching a 2D model of an object to the image. A survey of matching approaches is given in (Brown, 1992). The simplest class of object recognition methods is based on the gray values of the model and the image (Brown, 1992; Lai and Fang, 1999). A more complex class of object recognition uses the object's edges for matching, e.g., the mean edge distance (Borgefors, 1988), the Hausdorff distance (Rucklidge, 1997), or the generalized Hough transform (GHT) (Ballard, 1981).

All of the above approaches do not simultaneously meet the high industrial demands: robustness to occlusions, clutter, arbitrary illumination changes, and sensor noise as well as high recognition accuracy and real-time computation. Therefore, we developed two approaches, a new similarity measure (Steger, 2001), which uses the edge direction as feature, and a modification of the GHT (Ulrich et. al, 2001), which eliminates the disadvantages of slow computation, large memory amounts, and the limited accuracy of the GHT. Extensive performance evaluations (Ulrich and Steger, 2001), which also include a comparison to standard recognition methods, showed that our two novel approaches have considerable advantages.

All of the above mentioned recognition methods have in common that they require some form of a rigid model representing the object to be found. However, in several applications the assumption of a rigid model is not fulfilled. Elastic or flexible matching approaches (Bajcsy and Kovacic, 1989; Jain et al., 1996) are able to match deformable objects, which appear in medicine when dealing with magnetic resonance imaging or computer tomography, for example. Approaches for recognizing articulated objects are also available especially in the field of robotics (Hauck et el., 1997).

Indeed, for industrial applications like quality control or inspection tasks it is less important to find elastic or articulated objects, but to find objects that consist of several rigid parts that show arbitrary mutual movement, i.e., variations in distance and orientation. These variations potentially occur whenever a process is split into several single procedures that are – by intention or not – insufficiently "aligned" to each other, e.g., when applying a tampon print using several stamps or when equipping a circuit board with transistors or soldering points. An example is given in Figure 1, which shows several prints on the clip of a pen. The four images illustrate the mutual movements (variations) of the object parts: the position of the print on the clip varies and the dark gray part of the print moves relatively to the light gray part. Clearly, when taking the object as rigid it may not be found by the recognition approach. However, when trying to find the individual parts separately the search becomes computationally expensive since each part must be searched for in the entire image and the relations between the parts are not taken into account. This problem can hardly be solved taking articulated objects into account since there is no true justification for hinges, but the mutual variations can be more general. Because the object consists of several rigid parts, obviously, also elastic objects cannot model these movements. One possible solution is to generate several models each representing one configuration of the model parts and to match all of these models to the image. However, for large variations, this is very inefficient and not practical considering real-time computation.

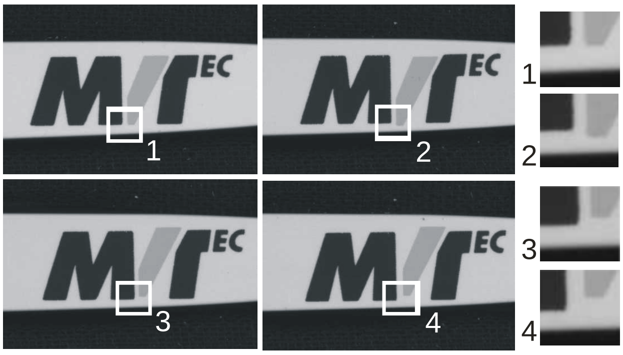

Figure 1. Four example images showing the variations caused by the printing with two different stamps. The marked windows with the white border are enlarged to make the variations clear.

In this paper, a novel approach is introduced that automatically decomposes the object into its rigid parts using several example images in which the mutual movements (variations) of the object parts are shown. Additionally, the variations of the object parts are analyzed and used to build a hierarchical model that contains all rigid model parts and a hierarchical search strategy, where the parts are searched relatively to each other taking the relations between the parts into account.
This model generation is called *offline phase* and has to be executed only once and therefore is not time-critical. But in the time critical *online phase* the hierarchical model facilitates a very efficient search.

2. OUTLINE OF THE APPROACH

In this section a coarse description of the algorithm to create a hierarchical model from the input data is presented. The whole process is summarized in the flowchart of Figure 2. In section 3, the single steps are explained in detail.
The only input data of the algorithm are a sample image of the object (*model image*), in which the object is defined by a region of interest (*ROI*), and some additional *example images* that, at least qualitatively, describe the mutual movements of the single object parts. Figure 3 shows an artificial example that is used to illustrate the algorithm.
The first step is to decompose the object, which is defined by the ROI within the model image, into small initial components. Note that these components need not coincide with the real object parts. For instance, if we use the connected components of the image edges as criterion for decomposition we would get the following components in our example: 1 hat, 1 face, 2 arms, 2 hands, 2 legs, 2 feet, the outer rectangle of the upper body, the inner rectangle of the upper body and at least 1 for each letter printed on the upper body. For each initial component a rigid model is built using a recognition method that is based on the image edges (cf. section 1) and able to find the object under rigid transformation (translation and rotation). Since we want to fulfill industrial demands we prefer to either use the similarity measure described in (Steger, 2001) or the modified Hough transform (Ulrich et al., 2001).

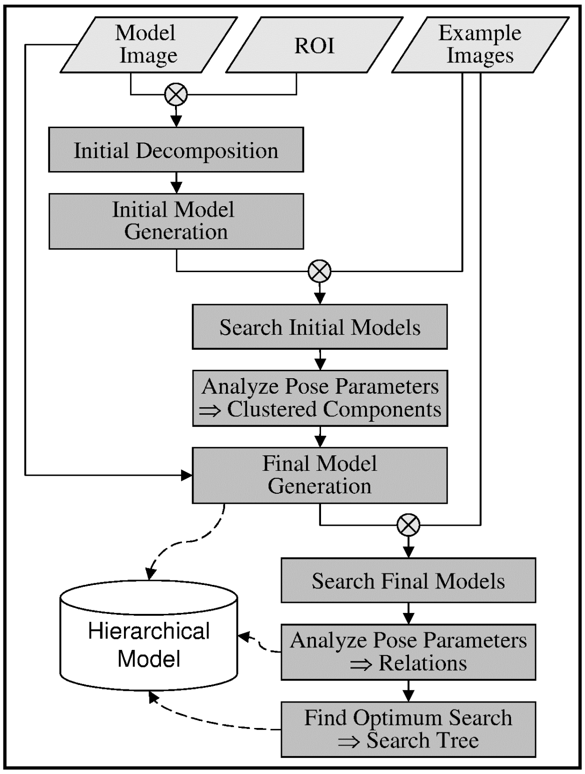

Figure 2. Flowchart of the algorithm that is used to create a hierarchical model from the input data (Model Image, Region of Interest (ROI), and Example Images).

Each initial model is searched for in all example images. Thus, we get the rigid transformation or pose parameters (position and orientation) of each initial component in each image. These parameters are analyzed and those initial components that form a rigid object part are merged together leading to the final decomposition. In our example, the hat and the face are clustered into one rigid part since they show the same movement in each image. The same holds for all initial components that form the upper part of the body. They are also clustered into one rigid part. Rigid models are built for each of the newly generated (clustered) parts and searched in the example images. Together with the models of the components that have not been clustered they describe the final models. The

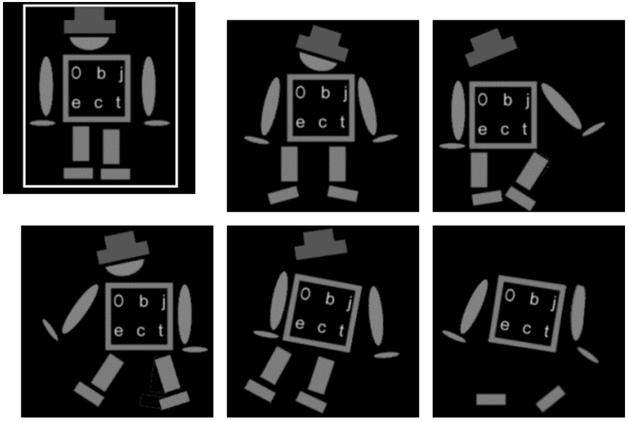

Figure 3. Input data: The upper left image represents the model image, in which the object is defined by the ROI (white rectangle). Additionally, five example images that show the mutual movements of the object parts are provided.

relations (relative movements) between each pair of the rigid object parts are computed by analyzing the pose parameters and stored in a fully connected directed graph, where the vertices represent the object parts and the link between vertices *i* and *j* describes the overall movement of part *j* relatively to part *i*. By computing the shortest arborescence of the graph we are able to ascertain a hierarchical search tree that incorporates an optimum search strategy in the sense that the search effort is minimized. Finally, the hierarchical model consists of the final models of the rigid object parts, the relations between the parts, and the optimum search strategy. Figure 4 shows the search tree and the corresponding search ranges for each part, which are described in the relations.

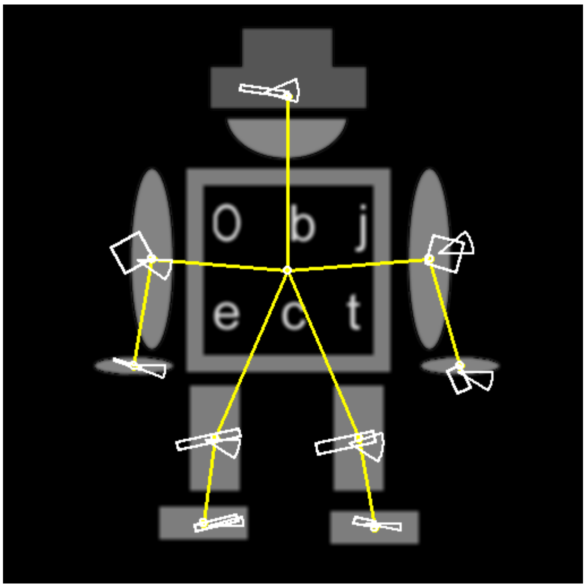

Figure 4. Result of the automatic hierarchical model generation. The vertices in the search tree correspond to the reference points (center of gravity) of each final model part. The search tree represents the optimum search strategy, e.g., the left hand is searched relatively to the left arm and not relatively to the upper body since the search range is smaller. The relative search ranges for the reference point of each part are visualized by white rectangles. The orientation search range is visualized by white circle sectors.

The hierarchical model can then be used to search the entire object containing the movable parts in an arbitrary search image. This is performed by searching the model parts in the image using the chosen similarity measure. Note that only one part must be searched within the entire search range, whereas the remaining parts can be searched in a very limited search space, which is defined by the relations in combination with the search tree.

3. DETAILED DESCRIPTION

In this section the single steps of the algorithm, which were introduced in section 2, are explained in detail.

3.1 Initial Decomposition

In the first step, the object, which is defined by the ROI in the model image, is initially broken up into small components. This can be done either automatically or interactively by the user. The condition the initial decomposition must fulfill is that each rigid object part must be represented by at least one component; otherwise the algorithm is not able to split this component later on and to find the rigid object parts automatically. Therefore, an over-segmentation should be preferred. However, very small components fail the property of being unique, but this can be balanced by our approach (see section 3.2). In our current implementation, edges are extracted in the model image by applying a threshold on the Sobel filter amplitude. The connected components of the edges are treated as individual initial components. Small components are either eliminated or affiliated to neighboring components. In Figure 5 the components are visualized by different colors.

Other grouping methods or combinations of them could also be included in our approach: Gestalt psychology has uncovered a set of principles guiding the grouping process in the visual domain (Wertheimer, 1923; Koffka, 1935; Rock and Palmer, 1990). Computer vision has taken advantage of these principles, e.g., in the field of perceptual organization and grouping (Ullman, 1979; Marr, 1982; Witkin and Tenenbaum, 1983; Lowe, 1985;).

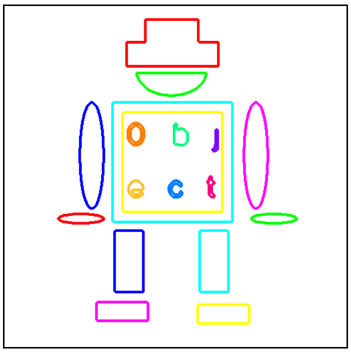

Figure 5. The initial decomposition based on image edges results in 18 components.

3.2 Initial Model Generation and Search

We use an implementation of the similarity measure presented in (Steger, 2001) as recognition approach to search the initial components in the example images. This approach uses image pyramids to speed up the recognition – like most implementations of conventional object recognition methods. However, one has to take care of unfavorable scale-space effects. In scale-space the edges of the model are influenced by neighboring edges. This is uncritical in most cases when dealing with large objects since there are still enough edges in the model left that are not influenced by neighboring edges and therefore still enable a good match. However, some problems occur when dealing with small objects, like the initial components in our example, since the ratio between the number of model edges and the number of neighboring edges is becoming small, i.e., the influence of the neighboring edges is increasing.

In Figure 6, the principle is shown using a 1D gray value profile, which includes two edges. Only the left edge belongs to the model whereas the right is a neighboring edge. In scale-space the disturbance of the model edge caused by the neighboring edge increases with the degree of smoothing (sigma). This problem could be avoided if we would not use image pyramids within the recognition method. However, this would lead to high computation times that are not suitable. Therefore, our solution is to extrapolate the gray values at the model border to the surrounding area to eliminate the disturbing neighboring edge (cf. Figure 7). Other, more sophisticated, approaches explicitly model the edges and subsequently reconstruct the gray values in the surrounding of the edges (Elder, 1999). These could be incorporated easily.

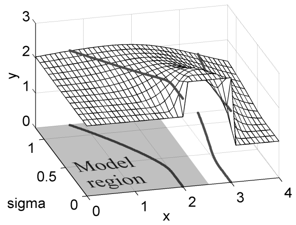

Figure 6. The model edge is disturbed by a neighboring edge in scale-space.

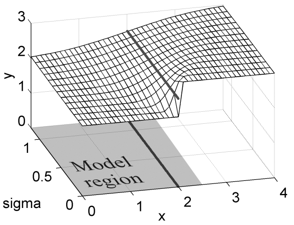

Figure 7. By extrapolating the gray values of the model the disturbing edge is eliminated.

After the disturbing edges have been eliminated for each component a model is built and used to search the components in each example image using the recognition approach. Thus, we obtain all poses P_i (including parameters position and orientation) of each component i in each example image.

Another problem arises when searching for small components: The result of the search may not be unique because of self-symmetries of the components or mutual similarities between the components. In our example (Figure 5) the left leg, for instance, is found four times in the first example image (Figure 3): At the true position of the left leg, at the position of the right leg and each at orientation 0° and 180°. Consequently, it is indispensable to solve these ambiguities to get the most likely pose for each component. Let n be the number of components and M_{i} the pose of component i in the model image (i=1,…,n). The pose represented by match k of component i in an example image is described by $E_{\mathrm{i}}^{\mathrm{k}}$, where k=1,…,n_{i} and n_{i} is the number of matches (found instances) of component i in the example image. We solve the ambiguities by minimizing the following equation:

$$\sum_{i=1}^{n}\arg\min_{k=1\ldots n_i}\left(\sum_{j=1}^{n}\arg\min_{l=1\ldots n_j}\left(\Psi(M_i,M_j,E_i^k,E_j^l)\right)\right)\to\min \tag{1}$$

Here, Ψ is a cost function that rates the relative pose of match l of component j to match k of component i in the example image by comparing it to the relative pose of the two components in the model image. The more the current relative pose in the example image differs from the relative pose in the model image the higher the cost value. In our current implementation Ψ takes the difference in position and orientation into account. This follows the principle of human perception where the correspondence problem of apparent motion is solved by minimizing the overall variation (Ullman, 1979).

The consequence of this step is that each component is assigned at most one pose in each example image.

3.3 Clustering of Components

Since the initial decomposition led to an over-segmentation, we now have to merge the components belonging to the same rigid object part to larger clusters by analyzing the pose parameters. Components that show similar apparent movement over all example images are clustered together.

We first calculate the pairwise probability of two components belonging to the same rigid object part. Let M_1=(x^{M}_1, $\mathrm{y}^{\mathrm{M}}_1$, φ^{M}_1), M_2(x^{M}_2, $\mathrm{y}^{\mathrm{M}}_2$, φ^{M}_2), E_1=(x^{E}_1, $\mathrm{y}^{\mathrm{E}}_1$, φ^{E}_1), and E_2=(x^{E}_2, $\mathrm{y}^{\mathrm{E}}_2$, φ^{E}_2) be the poses of two components in the model image and in an example image. Without loss of generality φ^{M}_1 and φ^{M}_2 are set to 0, since the orientations in the model image are taken as reference. The relative position of the two components in the model image is expressed by $\Delta\mathrm{x}^{\mathrm{M}}$=$x^{\mathrm{M}}_2$- x^{M}_1 and $\Delta\mathrm{y}^{\mathrm{M}}$=$\mathrm{y}^{\mathrm{M}}_2$- y^{M}_1. The same holds for the relative position $\Delta\mathrm{x}^{\mathrm{E}}$ and $\Delta\mathrm{y}^{\mathrm{E}}$ in the example image. To compare the relative position in the model and in the search image, we have to rotate the relative position in the example image back to the reference orientation:

$$\begin{bmatrix}\Delta\tilde{x}^E\\ \Delta\tilde{y}^E\end{bmatrix}=\begin{bmatrix}\cos\varphi_1^E & \sin\varphi_1^E\\ -\sin\varphi_1^E & \cos\varphi_1^E\end{bmatrix}\begin{bmatrix}\Delta x^E\\ \Delta y^E\end{bmatrix} \tag{2}$$

If the used recognition method additionally returns accuracy information of the pose parameters, the accuracy of the relative position is calculated with the law of error propagation. Otherwise the accuracy must be specified empirically. Then, the following hypothesis can be stated:

$$\begin{aligned}\Delta\tilde{x}^E &= \Delta x^M\\ \Delta\tilde{y}^E &= \Delta y^M\\ \varphi_1^E &= \varphi_2^E\end{aligned} \tag{3}$$

The probability of the correctness of this hypothesis corresponds to the probability that both components belong to the same rigid object part. It can be calculated using the equations for hypothesis tests as, e.g., given in (Koch, 1987). This is done for all object pairs and for all example images yielding a symmetric similarity matrix, in which at row i and column j the probability that the components i and j belong together is stored. The entries in the matrix correspond to the minimum value of the probabilities in all example images. To get a higher robustness to mismatches the mean or other statistical values can be used instead of the minimum value. In Figure 8 the similarity matrix for the example of Figure 3 is displayed. One can see the high probability that hat and face belong together and that the components forming the upper part of the body form a rigid part.

Based on this similarity matrix the initial components are clustered using a pairwise clustering strategy that successively merges the two entities with the highest similarity until the maximum of the remaining similarities is smaller than a predefined threshold.

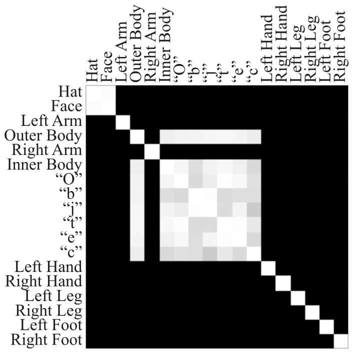

Figure 8. The similarity matrix contains the probabilities that two components belong to the same rigid object part. The higher the probability the brighter the entry.

3.4 Final Model Generation and Search

Models for the recognition approach for the newly clustered components are created and searched for in all example images as described in section 3.2. This is necessary if we want to avoid errors that are introduced when taking the average of the single initial poses of each component within the cluster as pose for the newly clustered component. However, we can exploit this information to reduce the search space by calculating approximate values for the reference point and the orientation angle of the new component in the example images. After this step for each rigid object part a model is available and the pose parameters for each object part in each image are computed.

3.5 Analysis of Relations

In this section the pose parameters of the clustered components, i.e., rigid object parts, are analyzed and the pairwise relations between part *i* and *j* are derived (where $i = 1, \ldots, n_p$ and $j = 1, \ldots, n_p$ and n_p is the number of object parts). For this purpose, in each image, the pose of part *i* defines a local coordinate system in which the pose of part *j* is calculated. The angle range that encloses all orientations of part *j* in the local coordinate systems of all images describes the angle variation of part *j* with respect to part *i*. The corresponding position variation is described by the smallest enclosing rectangle of arbitrary orientation of the reference points of part *j* in the local coordinate systems of all images. The principle is exemplified in Figure 9.

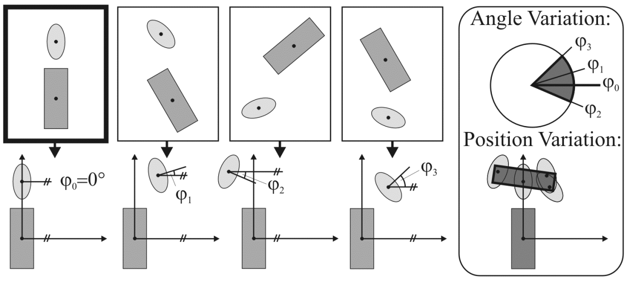

Figure 9. The relation between an object pair (rectangle and ellipse) is computed from the relative poses in the model image (bold border) and in three example images (upper pictures). In this example the rectangle is taken as reference and the relative movement of the ellipse is computed by transforming the ellipse into the reference system defined by the rectangle (lower pictures). The overall relative orientation describes the angle variation (dark circle sector in the right picture) and the smallest enclosing rectangle of arbitrary orientation of all ellipse reference points is taken as position variation (dark rectangle in the right picture). Note that the ambiguities because of symmetries of both objects are solved according to (1).

Apart from the angle variation and the position variation the relation information additionally includes statistical values like the mean and the standard deviation of the relative angle and of the relative position. This information is calculated for each ordered object pair. In order to find an optimum search strategy that minimizes the entire search effort (cf. section 3.6) we must define a variation measure that quantifies the search effort Ω_{ij} that must be expended to search part *j* if the pose of part *i* is known. We define the search effort as

$$\Omega_{ij} = l_{ij} \cdot h_{ij} \cdot \Delta\varphi_{ij}\,, \qquad (4)$$

where l_{ij} and h_{ij} is the length and the height of the smallest enclosing rectangle, respectively, describing the position variation of part *j* relative to part *i*, and $\Delta\varphi_{ij}$ specifies the corresponding angle variation. Please note that Ω is not symmetric, i.e., Ω_{ij} is not necessarily equal to Ω_{ji}. Since we cannot expect the example images to cover the variations completely but only qualitatively, the values for l_{ij}, h_{ij}, and $\Delta\varphi_{ij}$ can be adapted by applying a tolerance.

Our strategy is to search a selected root part within the entire search range and then successively search the remaining parts only relatively to the parts already found. To do so, the search region of the part's reference point is described by the enclosing rectangle transformed to the pose from which the part is searched. Since the computation time of most recognition methods increases linearly with Ω we have to minimize the sum of the Ωs that are accumulated during the search to find an optimum search strategy.

3.6 Selection of the Optimum Search Strategy

Based on the search effort Ω for all object parts we are able to compute the optimum search strategy that minimizes the overall recognition time by applying graph theory to our problem. We can interpret the object parts as vertices in a graph where the directed arc between the vertices *i* and *j* is weighted with the corresponding search effort Ω_{ij}. Thus, we get a fully connected directed graph *D=(V,A)*, where *V* denotes the set of vertices of size $|V|=n_p$ and *A* the set of arcs of size $|A|= n_p(n_p -1)$. With each arc $a_{ij} \in A$ the weight Ω_{ij} is associated. An *arborescence* of *D* is a subtree of *D* such that there is a particular vertex called the root, which is not the terminal vertex of any arc, and that for any other vertex v_i , there is exactly one arc, whose terminal vertex is v_i. A *spanning arborescence* of *D* is an arborescence that contains all vertices of *D*. Thus, the problem of finding an optimum search strategy is equivalent to finding a spanning arborescence *H=(V,B)* of *D*, such that

$$\sum_{(v_i,v_j)\in V} b_{ij} \to \min \qquad (5)$$

An algorithm for finding the spanning arborescence of minimum weight in a graph is defined in (Chu and Liu, 1965).

The root vertex can be chosen using different criteria. Since the root vertex corresponds to the only object part that is searched for not relatively to another object part, the recognition time of the online phase strongly depends on the recognition time of the root part. Therefore, when using the recognition method presented in (Steger, 2001) large object parts should be preferred to be the root part since more pyramid levels can be used to speed up the search. Furthermore, the root part should not be self-symmetric to avoid ambiguities during the online phase, which complicate the search. The root part plays another decisive role: it should be ensured that the root part is always found during the search in the online phase, since the whole object cannot be found if the root part is missing or occluded to a high degree. In practice, these criteria must be balanced.

Figure 4 illustrates the result of the optimum search strategy. Here, the upper part of the body was selected to be the root part. Thus, the upper body is searched for in the entire image, the left arm is searched relatively to the upper body taking the relations into account (cf. section 3.5), the left hand is searched relatively to the left arm, etc.

Finally, the hierarchical model consists of the final models of the rigid object parts (cf. section 3.4), the relations between the parts (cf. section 3.5), and the optimum search strategy.

4. EXAMPLES

Two short examples are presented to show the high potential of our novel approach. In the first example we applied it to the tampon print discussed in section 1. Beside the model image shown in Figure 1 and the ROI enclosing the complete print on the clip, we took 20 example images of different pen clips to decompose the object into its parts, calculate the relations and derive the search strategy. As initial components the light gray letter and the four dark gray letters were found. After the clustering step the four dark gray letters were merged to one rigid part. Therefore, our final hierarchical model combined two rigid object parts, where the merged part was selected as root part. The variation of the second part relative to the root part was determined to be l_{12}=5pixel, h_{12}=15pixel, and $\Delta\varphi_{12} = 0.7°$. From this information we can conclude that the two stamps are nearly perfectly aligned regarding the orientation but only poorly aligned regarding the position particularly in the direction perpendicular to the direction of the script. It took about 20 ms on a 2 GHz Pentium 4 to search the print in the entire image of size 652×494 allowing a 360° rotation. The 20 ms can be completely attributed to the search of the root part. The search of the second part was too fast to be measured. If we would search the second part separately it would also take 20 ms to find it. Therefore, by using our hierarchical model the recognition time was reduced to 50%.

In the second example the variations of another writing are analyzed (see Figure 10). The resulting search tree in the lower right image of Figure 10 visualizes the resulting optimum search strategy. The result is to search each letter relatively to its neighboring letter, which corresponds to our intuition.

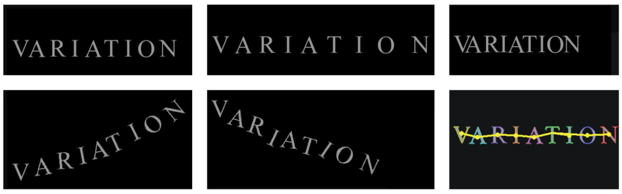

Figure 10. The variations in the writing are analyzed and used to build the hierarchical model. The calculated search tree is visualized in the lower right image, where each letter is searched relative to its neighboring letter, as we would expect.

It took 60 ms to search the entire hierarchical model in a 512x512 image allowing a 180° rotation. 50 ms can be attributed to the search of the root part and only 10 ms to the search of all other parts. In a comparison to a complete search of all parts in the entire image (450 ms) this is a reduction to 13%.

5. CONCLUSIONS

In this paper we presented an approach for hierarchical automatic object decomposition for object recognition. This is useful when searching for objects that consist of several parts that can move relative to each other, which often happens in industry, for example. A hierarchical model was automatically created using several example images that show the relative movements of the single parts. This model can be utilized to efficiently search for the object in an arbitrary image. The examples shown in section 4 emphasize the high potential of our novel approach.

REFERENCES

Bajcsy, R., Kovacic, S., 1989. Multi-resolution elastic matching. Computer Vision, Graphics, and Image Processing, 46(1): pp. 1-21.

Ballard, D. H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), pp. 111-122.

Borgefors, G., 1988. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6), pp. 849-865.

Brown, L. G., 1992. A survey of image registration techniques. *ACM Computing Surveys*, 24(4), pp. 325-376.

Chu, Y. J. and Tseng-Hong, L., 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14(10), pp. 1396-1400.

Elder, J., 1999. “Are Edges Incomplete?”. *International Journal of Computer Vision*, 34(2/3), pp. 97-122.

Hauck, A., Lanser, S., and Zierl, C., 1997. Hierarchical Recognition of Articulated Objects from Single Perspective Views. In: Proc. *Computer Vision and Pattern Recognition (CVPR’97)*, IEEE Computer Society Press, pp. 870-883.

Huttenlocher, D. P., Klanderman, and G. A., Rucklidge, W. J., 1993. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), pp. 850-863.

Jain, A. K., Zhong, Y., and Lakshmanan, S., 1996. Object matching using deformable templates. *IEEE Transactions on patterns analysis and machine intelligence*, 18(3), pp. 267-277.

Koch, K. R., 1987. *Parameterschätzung und Hypothesentests in linearen Modellen*. Dümmler, Bonn.

Koffka, K., 1935. *Principles of Gestalt Psychology*. Harcourt Brace, New York.

Lai, S. and Fang, M., 1999. Accurate and fast pattern localization algorithm for automated visual inspection. *Real-Time Imaging*, 5, pp. 3-14.

Lowe, D.G., 1985. *Perceptual Organization and Visual Recognition*, Kluwer Academics, Boston.

Marr, D., 1982. *Vision*, W.H. Freeman and Company, San Francisco, CA.

Rucklidge, W. J., 1997. Efficiently locating objects using the Hausdorff distance. *International Journal of Computer Vision*, 24(3), pp. 251-270.

Steger, C., 2001. Similarity measures for occlusion, clutter, and illumination invariant object recognition. In: *Mustererkennung 2001*, B. Radig and S. Florczyk (eds), Springer, Berlin, pp. 148-154.

Ullman, S., 1979. *The interpretation of visual motion*. MIT Press, Cambridge, MA.

Ulrich, M., Steger, C., Baumgartner, A., and Ebner, H., 2001. Real-time object recognition in digital images for industrial applications. In: *5th Conference on Optical 3-D Measurement Techniques*, Vienna, Austria, pp. 308-318.

Ulrich, M. and Steger, C., 2001. Empirical performance evaluation of object recognition methods. In: *Empirical Evaluation Methods in Computer Vision*, H. I. Christensen and P. J. Phillips (eds), IEEE Computer Society Press, Los Alamitos, CA, pp. 62-76.

Wertheimer, M., 1938. Laws of Organization in Perceptual Forms. In: *A Source Book of Gestalt Psychology*, W. D. Ellis (ed), Harcourt Brace.

Witkin, A.P. and Tenenbaum, J.M., 1983. On the Role of Structure in Vision. In: *Human and Machine Vision*, Jacob Beck and Barbara Hope and Azriel Rosenfeld (eds), Academic Press, New York.