OCCLUSION, CLUTTER, AND ILLUMINATION INVARIANT OBJECT RECOGNITION

Carsten Steger
MVTec Software GmbH
Neherstraße 1, 81675 München, Germany
steger@mvtec.com

Commission III, Working Group III/5

KEY WORDS: Computer Vision, Real-Time Object Recognition

ABSTRACT

An object recognition system for industrial inspection that recognizes objects under similarity transformations in real time is proposed. It uses novel similarity measures that are inherently robust against occlusion, clutter, and nonlinear illumination changes. They can be extended to be robust to global as well as local contrast reversals. The matching is performed based on the maxima of the similarity measure in the transformation space. For normal applications, subpixel-accurate poses are obtained by extrapolating the maxima of the similarity measure from discrete samples in the transformation space. For applications with very high accuracy requirements, least-squares adjustment is used to further refine the extracted pose.

1 INTRODUCTION

Object recognition is used in 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 or quality control. In most cases, the model of the object is generated from an image of the object. This 2D approach is taken 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 usually interested in matching a 2D model of an object to the image. The object may be transformed by a certain class of transformations, depending on the particular setup, e.g., translations, euclidean transformations, similarity transformations, or general 2D affine transformations (which are usually taken as an approximation to the true perspective transformations an object may undergo).

A large number of object recognition strategies exist. The approach to object recognition proposed in this paper uses pixels as its geometric features, i.e., not higher level features like lines or elliptic arcs. Therefore, only similar pixel-based strategies will be reviewed.

Several methods have been proposed to recognize objects in images by matching 2D models to images. A survey of matching approaches is given in (Brown, 1992). In most 2D matching approaches the model is systematically compared to the image using all allowable degrees of freedom of the chosen class of transformations. The comparison is based on a suitable similarity measure (also called match metric). The maxima or minima of the similarity measure are used to decide whether an object is present in the image and to determine its pose. To speed up the recognition process, the search is usually done in a coarse-to-fine manner, e.g., by using image pyramids (Tanimoto, 1981).

The simplest class of object recognition methods is based on the gray values of the model and image itself and uses normalized cross correlation or the sum of squared or absolute differences as a similarity measure (Brown, 1992). Normalized cross correlation is invariant to linear brightness changes but is very sensitive to clutter and occlusion as well as nonlinear contrast changes. The sum of gray value differences is not robust to any of these changes, but can be made robust to linear brightness changes by explicitly incorporating them into the similarity measure, and to a moderate amount of occlusion and clutter by computing the similarity measure in a statistically robust manner (Lai and Fang, 1999).

A more complex class of object recognition methods does not use the gray values of the model or object itself, but uses the object's edges for matching (Borgefors, 1988, Rucklidge, 1997). In all existing approaches, the edges are segmented, i.e., a binary image is computed for both the model and the search image. Usually, the edge pixels are defined as the pixels in the image where the magnitude of the gradient is maximum in the direction of the gradient. Various similarity measures can then be used to compare the model to the image. The similarity measure in (Borgefors, 1988) computes the average distance of the model edges and the image edges. The disadvantage of this similarity measure is that it is not robust to occlusions because the distance to the nearest edge increases significantly if some of the edges of the model are missing in the image.

The Hausdorff distance similarity measure used in (Rucklidge, 1997) tries to remedy this shortcoming by calculating the maximum of the k-th largest distance of the model edges to the image edges and the l-th largest distance of the image edges and the model edges. If the model contains n points and the image contains m edge points, the similarity measure is robust to $100k/n$% occlusion and $100l/m$% clutter. Unfortunately, an estimate for m is needed to determine l, which is usually not available.

All of these similarity measures have the disadvantage that they do not take into account the direction of the edges. In (Olson and Huttenlocher, 1997) it is shown that disregarding the edge direction information leads to false positive instances of the model in the image. The similarity measure proposed in (Olson and Huttenlocher, 1997) tries to improve this by modifying the Hausdorff distance to also measure the angle difference between the model and image edges. Unfortunately, the implementation is based on multiple distance transformations, which makes the algorithm too computationally expensive for industrial inspection.

Finally, another class of edge based object recognition algorithms is based on the generalized Hough transform (Ballard, 1981). Approaches of this kind have the advantage that they are robust to occlusion as well as clutter. Unfortunately, the GHT requires extremely accurate estimates for the edge directions or a complex and expensive processing scheme, e.g., smoothing the accumulator space, to determine whether an object is present and to determine its pose. This problem is especially grave for large models. The required accuracy is usually not obtainable, even in low noise images, because the discretization of the image leads to edge direction errors that already are too large for the GHT.

In all approaches above, the edge image is binarized. This makes the object recognition algorithm invariant only against a narrow range of illumination changes. If the image contrast is lowered, progressively fewer edge points will be segmented, which has the same effects as progressively larger occlusion. The similarity measures proposed in this paper overcome all of the above problems and result in an object recognition strategy robust against occlusion, clutter, and nonlinear illumination changes. They can be extended to be robust to global as well as local contrast reversals.

2 SIMILARITY MEASURES

The model of an object consists of a set of points $p_i = (x_i, y_i)^T$ and associated direction vectors $d_i = (t_i, u_i)^T$, $i = 1, \ldots, n$. The direction vectors can be generated by a number of different image processing operations, e.g., edge, line, or corner extraction, as discussed in Section 3. Typically, the model is generated from an image of the object, where an arbitrary region of interest (ROI) specifies that part of the image in which the object is located. It is advantageous to specify the coordinates p_i relative to the center of gravity of the ROI of the model or to the center of gravity of the points of the model.

The image in which the model should be found can be transformed into a representation in which a direction vector $e_{x,y} = (v_{x,y}, w_{x,y})^T$ is obtained for each image point (x, y). In the matching process, a transformed model must be compared to the image at a particular location. In the most general case considered here, the transformation is an arbitrary affine transformation. It is useful to separate the translation part of the affine transformation from the linear part. Therefore, a linearly transformed model is given by the points $p'_i = Ap_i$ and the accordingly transformed direction vectors $d'_i = Ad_i$, where

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad .$$

As discussed above, the similarity measure by which the transformed model is compared to the image must be robust to occlusions, clutter, and illumination changes. One such measure is to sum the (unnormalized) dot product of the direction vectors of the transformed model and the image over all points of the model to compute a matching score at a particular point $q = (x, y)^T$ of the image, i.e., the similarity measure of the transformed model at the point q, which corresponds to the translation part of the affine transformation, is computed as follows:

$$\begin{aligned} s &= \frac{1}{n} \sum_{i=1}^{n} \langle d'_i, e_{q+p'} \rangle & (1) \\ &= \frac{1}{n} \sum_{i=1}^{n} t'_i v_{x+x'_i, y+y'_i} + u'_i w_{x+x'_i, y+y'_i} \quad . \end{aligned}$$

If the model is generated by edge or line filtering, and the image is preprocessed in the same manner, this similarity measure fulfills the requirements of robustness to occlusion and clutter. If parts of the object are missing in the image, there are no lines or edges at the corresponding positions of the model in the image, i.e., the direction vectors will have a small length and hence contribute little to the sum. Likewise, if there are clutter lines or edges in the image, there will either be no point in the model at the clutter position or it will have a small length, which means it will contribute little to the sum.

The similarity measure (1) is not truly invariant against illumination changes, however, since usually the length of the direction vectors depends on the brightness of the image, e.g., if edge detection is used to extract the direction vectors. However, if a user specifies a threshold on the similarity measure to determine whether the model is present in the image, a similarity measure with a well defined range of values is desirable. The following similarity measure achieves this goal:

$$\begin{aligned} s &= \frac{1}{n} \sum_{i=1}^{n} \frac{\langle d'_i, e_{q+p'} \rangle}{\|d'_i\| \cdot \|e_{q+p'}\|} & (2) \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{t'_i v_{x+x'_i, y+y'_i} + u'_i w_{x+x'_i, y+y'_i}}{\sqrt{t'^2_i + u'^2_i} \cdot \sqrt{v^2_{x+x'_i, y+y'_i} + w^2_{x+x'_i, y+y'_i}}} \quad . \end{aligned}$$

Because of the normalization of the direction vectors, this similarity measure is additionally invariant to arbitrary illumination changes since all vectors are scaled to a length of 1. What makes this measure robust against occlusion and clutter is the fact that if a feature is missing, either in the model or in the image, noise will lead to random direction vectors, which, on average, will contribute nothing to the sum.

The similarity measure (2) will return a high score if all the direction vectors of the model and the image align, i.e., point in the same direction. If edges are used to generate the model and image vectors, this means that the model and image must have the same contrast direction for each edge. Sometimes it is desirable to be able to detect the object even if its contrast is reversed. This is achieved by:

$$s = \left| \frac{1}{n} \sum_{i=1}^{n} \frac{\langle d'_i, e_{q+p'} \rangle}{\|d'_i\| \cdot \|e_{q+p'}\|} \right| \quad . \qquad (3)$$

In rare circumstances, it might be necessary to ignore even local contrast changes. In this case, the similarity measure can be modified as follows:

$$s = \frac{1}{n} \sum_{i=1}^{n} \frac{|\langle d'_i, e_{q+p'} \rangle|}{\|d'_i\| \cdot \|e_{q+p'}\|} \quad . \qquad (4)$$

The above three normalized similarity measures are robust to occlusion in the sense that the object will be found if it is occluded. As mentioned above, this results from the fact that the missing object points in the instance of the model in the image will on average contribute nothing to the sum. For any particular instance of the model in the image, this may not be true, e.g., because the noise in the image is not uncorrelated. This leads to the undesired fact that the instance of the model will be found in different poses in different images, even if the model does not move in the images, because in a particular image of the model the random direction vectors will contribute slightly different amounts to the sum, and hence the maximum of the similarity measure will change randomly. To make the localization of the model more precise, it is useful to set the contribution of direction vectors caused by noise in the image to zero. The easiest way to do this is to set all inverse lengths $1/\|e_{q+p'}\|$ of the direction vectors in the image to 0 if their length $\|e_{q+p'}\|$ is smaller than a threshold that depends on the noise level in the image and the preprocessing operation that is used to extract the direction vectors in the image. This threshold can be specified easily by the user. By this modification of the similarity measure, it can be ensured that an occluded instance of the model will always be found in the same pose if it does not move in the images.

The normalized similarity measures (2)–(4) have the property that they return a number smaller than 1 as the score of a potential match. In all cases, a score of 1 indicates a perfect match between the model and the image. Furthermore, the score roughly corresponds to the portion of the model that is visible in the image. For example, if the object is 50% occluded, the score (on average) cannot exceed 0.5. This is a highly desirable property because it gives the user the means to select an intuitive threshold for when an object should be considered as recognized.

A desirable feature of the above similarity measures (2)–(4) is that they do not need to be evaluated completely when object recognition is based on a threshold $s_{\min}$ for the similarity measure that a potential match must achieve. Let s_j denote the partial sum of the dot products up to the j-th element of the model. For the match metric that uses the sum of the normalized dot products, this is:

$$s_j = \frac{1}{n} \sum_{i=1}^{j} \frac{\langle d'_i, e_{q+p'} \rangle}{\|d'_i\| \cdot \|e_{q+p'}\|} \quad . \tag{5}$$

Obviously, all the remaining terms of the sum are all ≤ 1. Therefore, the partial score can never achieve the required score $s_{\min}$ if $s_j < s_{\min} - 1 + j/n$, and hence the evaluation of the sum can be discontinued after the j-th element whenever this condition is fulfilled. This criterion speeds up the recognition process considerably.

Nevertheless, further speed-ups are highly desirable. Another criterion is to require that all partial sums have a score better than $s_{\min}$, i.e., $s_j \geq s_{\min}$. When this criterion is used, the search will be very fast, but it can no longer be ensured that the object recognition finds the correct instances of the model because if missing parts of the model are checked first, the partial score will be below the required score. To speed up the recognition process with a very low probability of not finding the object although it is visible in the image, the following heuristic can be used: the first part of the model points is examined with a relatively safe stopping criterion, while the remaining part of the model points are examined with the hard threshold $s_{\min}$. The user can specify what fraction of the model points is examined with the hard threshold with a parameter g. If $g = 1$, all points are examined with the hard threshold, while for $g = 0$, all points are examined with the safe stopping criterion. With this, the evaluation of the partial sums is stopped whenever $s_j < \min(s_{\min} - 1 + fj/n, s_{\min}j/n)$, where $f = (1 - gs_{\min})/(1 - s_{\min})$. Typically, the parameter g can be set to values as high as 0.9 without missing an instance of the model in the image.

3 OBJECT RECOGNITION

The above similarity measures are applied in an object recognition system for industrial inspection that recognizes objects under similarity transformations, i.e., translation, rotation, and uniform scaling, in real time. Although only similarity transformations are implemented at the moment, extensions to general affine transformations are not difficult to implement. The system consists of two modules: an offline generation of the model and an online recognition.

The model is generated from an image of the object to be recognized. An arbitrary region of interest specifies the object's location in the image. Usually, the ROI is specified by the user. Alternatively, it can be generated by suitable segmentation techniques. To speed up the recognition process, the model is generated in multiple resolution levels, which are constructed by building an image pyramid from the original image. The number of pyramid levels $l_{\max}$ is chosen by the user.

Each resolution level consists of all possible rotations and scalings of the model, where thresholds $\phi_{\min}$ and $\phi_{\max}$ for the angle and $\sigma_{\min}$ and $\sigma_{\max}$ for the scale are selected by the user. The step length for the discretization of the possible angles and scales can either be done automatically by a method similar to the one described in (Borgefors, 1988) or be set by the user. In higher pyramid levels, the step length for the angle is computed by doubling the step length of the next lower pyramid level.

The rotated and scaled models are generated by rotating and scaling the original image of the current pyramid level and performing the feature extraction in the rotated image. This is done because the feature extractors may be anisotropic, i.e., the extracted direction vectors may depend on the orientation of the feature in the image in a biased manner. If it is known that the feature extractor is isotropic, the rotated models may be generated by performing the feature extraction only once per pyramid level and transforming the resulting points and direction vectors.

The feature extraction can be done by a number of different image processing algorithms that return a direction vector for each image point. One such class of algorithms are edge detectors, e.g, the Sobel or Canny (Canny, 1986) operators. Another useful class of algorithms are line detectors (Steger, 1998). Finally, corner detectors that return a direction vector, e.g., (Förstner, 1994), could also be used. Because of runtime considerations the Sobel filter is used in the current implementation of the object recognition system. Since in industrial inspection the lighting can be controlled, noise does not pose a significant problem in these applications.

To recognize the model, an image pyramid is constructed for the image in which the model should be found. For each level of the pyramid, the same filtering operation that was used to generate the model, e.g., Sobel filtering, is applied to the image. This returns a direction vector for each image point. Note that the image is not segmented, i.e., thresholding or other operations are not performed. This results in true robustness to illumination changes.

To identify potential matches, an exhaustive search is performed for the top level of the pyramid, i.e., all precomputed models of the top level of the model resolution hierarchy are used to compute the similarity measure via (2), (3), or (4) for all possible poses of the model. A potential match must have a score larger than a user-specified threshold $s_{\min}$ and the corresponding score must be a local maximum with respect to neighboring scores. As described in Section 2, the threshold $s_{\min}$ is used to speed up the search by terminating the evaluation of the similarity measure as early as possible. With the termination criteria, this seemingly brute-force strategy actually becomes extremely efficient. On average, about 9 pixels of the model are tested for every pose on the top level of the pyramid.

After the potential matches have been identified, they are tracked through the resolution hierarchy until they are found at the lowest level of the image pyramid. Various search strategies like depth-first, best-first, etc., have been examined. It turned out that a breadth-first strategy is preferable for various reasons, most notably because a heuristic for a best-first strategy is hard to define, and because depth-first search results in slower execution if all matches should be found.

Once the object has been recognized on the lowest level of the image pyramid, its position and rotation are extracted to a resolution better than the discretization of the search space, i.e., the translation is extracted with subpixel precision and the angle and scale with a resolution better than their respective step lengths. This is done by fitting a second order polynomial (in the four pose variables) to the similarity measure values in a $3 \times 3 \times 3 \times 3$ neighborhood around the maximum score. The coefficients of the polynomial are obtained by convolution with 4D facet model masks. The corresponding 2D masks are given in (Steger, 1998). They generalize to arbitrary dimensions in a straightforward manner.

4 LEAST-SQUARES POSE REFINEMENT

While the pose obtained by the extrapolation algorithm is accurate enough for most applications, in some applications an even higher accuracy is desirable. This can be achieved through a least-squares adjustment of the pose parameters. To achieve a better accuracy than the extrapolation, it is necessary to extract the model points as well as the feature points in the image with subpixel accuracy. If this would not be done, the image and model points would be separated radially by about 0.25 pixels on average if each model point is matched to its closest image point. However, even if the points are extracted with subpixel accuracy, an algorithm that performs a least-squares adjustment based on closest point distances would not improve the accuracy much since the points would still have an average distance significantly larger that 0 tangentially because the model and image points are not necessarily sampled at the same points and distances. Because of this, the proposed algorithm finds the closest image point for each model point and then minimizes the sum of the squared distances of the image points to a line defined by their corresponding model point and the corresponding tangent to the model point, i.e., the directions of the model points are taken to be correct and are assumed to describe the direction of the object's border. If, for example, an edge detector is used, the direction vectors of the model are perpendicular to the object boundary, and hence the equation of a line through a model point tangent to the object boundary is given by $t_i(x - x_i) + u_i(y - y_i) = 0$. Let $q_i = (v_i, w_i)^T$ denote the matched image points corresponding to the model points p_i. Then, the following function is minimized to refine the pose a:

$$d(a) = \sum_{i=1}^{n} [t_i(v_i(a) - x_i) + u_i(w_i(a) - y_i)]^2 \rightarrow \min \quad . \quad (6)$$

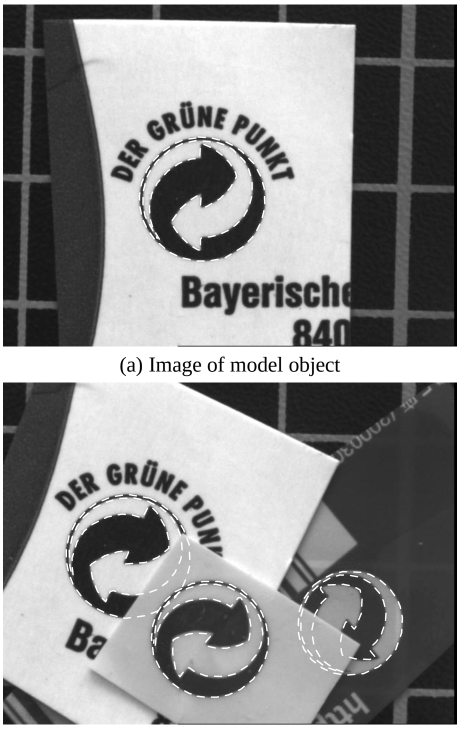
The potential corresponding image points in the search image are obtained by a non-maximum suppression only and are extrapolated to subpixel accuracy (Steger, 2000). By this, a segmentation of the search image is avoided, which is important to preserve the invariance against arbitrary illumination changes. For each model point the corresponding image point in the search image is chosen as the potential image point with the smallest euclidian distance using the pose obtained by the extrapolation to transform the model to the search image. Because the points in the search image are not segmented, spurious image points may be brought into correspondence with model points. Therefore, to make the adjustment robust, only correspondences with a distance smaller than a robustly computed standard deviation of the distances are used for the adjustment. Since (6) results in a linear equation system when similarity transformations are considered, one iteration suffices to find the minimum distance. However, since the point correspondences may change by the refined pose, an even higher accuracy can be gained by iterating the correspondence search and pose refinement. Typically, after three iterations the accuracy of the pose no longer improves.

5 EXAMPLE

Figure 1 displays an example of recognizing multiple objects at different scales and rotations. The model image is shown in Figure 1(a), while Figure 1(b) shows that all three instances of the model have been recognized correctly despite the fact that two of them are occluded, that one of them is printed with the contrast reversed, and that two of the models were printed with slightly different shapes. The time to recognize the models was 103 ms on an 800 MHz Pentium III running under Linux.

6 PERFORMANCE EVALUATION

To assess the performance of the proposed object recognition system, two different criteria were used: the recognition rate and the subpixel accuracy of the results.



(a) Image of model object

(b) Found objects

Figure 1: Example of recognizing multiple objects. Note that the model is found despite global contrast reversals and despite the fact that two of the models were printed with slightly different shapes.

To test the recognition rate, 500 images of an IC were taken. The IC was occluded to various degrees with various objects, so that in addition to occlusion, clutter of various degrees was created in the image. Figure 2 shows six of the 500 images that were used to test the recognition rate. The model was generated from the print on the IC in the top left image of Figure 2. On the lowest pyramid level it contained 2127 edge points.

An effort was made to keep the IC in exactly the same position in the image in order to be able to measure the degree of occlusion. Unfortunately, the IC moved very slightly (by less than one pixel) during the acquisition of the images. The true amount of occlusion was determined by extracting edges from the images and intersecting the edge regions with the edges that constitute the model. Since the objects that occlude the IC generate clutter edges, this actually underestimates the occlusion.

The model was extracted in the 500 images with $s_{\min} = 0.3$, i.e., the method should find the object despite 70% occlusion. Only the translation parameters were determined. The average recognition time was 22 ms. The model was recognized in 478 images, i.e., the recognition rate was 95.6%. By visual inspection, it was determined that in 15 of the 22 misdetection cases the IC was occluded by more than 70%. If these cases are removed the recognition rate rises to 98.6%. In the remaining seven cases, the occlusion was close to 70%. Figure 3(a) displays a plot of the extracted scores against the estimated visibility of the object. The instances in which the model was not found are denoted by a score of 0, i.e., they lie on the x axis of the plot. Figure 3(b) shows the errors of the extracted positions when extrapolating the pose as described in Section 3. It can be seen that the IC was accidentally shifted twice. The position errors are all very close to the three cluster centers. Some of the larger errors in the y coordinate result from refraction effects caused by the transparent ruler that was used in some images to occlude the IC (see the top right image of Figure 2). Figures 3(c) and (d) display the position errors

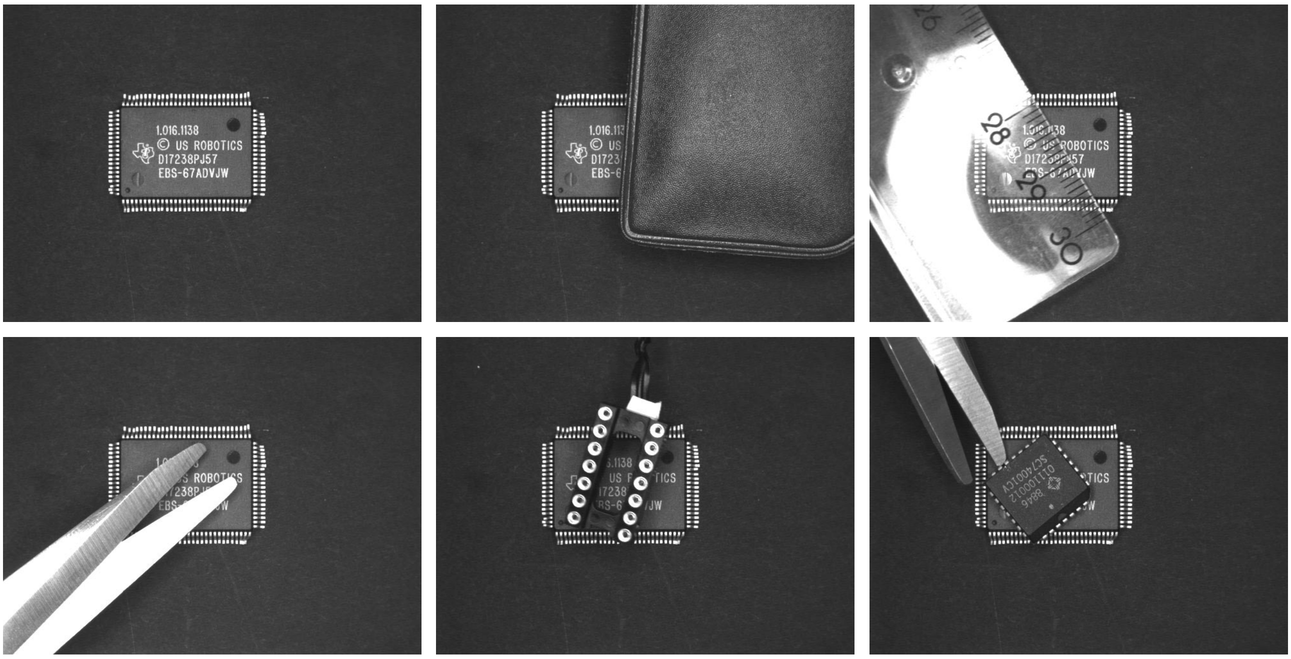


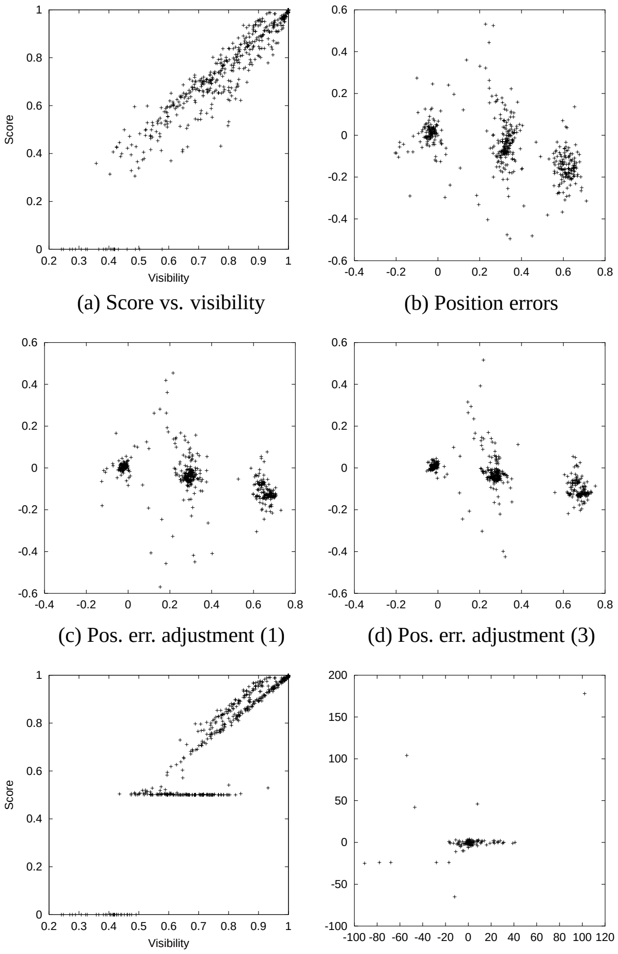
Figure 2: Six of the 500 images that were used to test the recognition rate. The model was generated from the print on the IC in the top left image. With the proposed approach, the model was found in all images except the lower right image.

after one and three iterations of the least-squares adjustment described in Section 4. Evidently, the extracted positions lie much closer to the three cluster centers. Furthermore, it can be seen that the least-squares adjustment does not introduce matching errors, i.e., outliers, and hence is very robust.

To check whether the proposed approach results in an improvement over existing approaches, the original implementation by Rucklidge (Rucklidge, 1997) of an approach that uses the partial Hausdorff distance as a similarity measure was used for the same test. The parameter for the maximum object to image and image to object distance were set to 1. Initial tests with the forward and reverse fractions set to 0.3 resulted in run times of more than three hours per image. Therefore, the forward and reverse fractions were set to 0.5. This resulted in an average matching time of 2.27 s per image, i.e., more that 100 times as long as the proposed approach. Since the method of (Rucklidge, 1997) returns all matches that fulfill its score and distance criteria, the best match was selected based on the minimum forward distance. If more than one match had the same minimum forward distance, the match with the maximum forward fraction was selected as the best match. A match was considered correct if its distance to the reference point of the model was less than one pixel. With this, the IC was recognized in 361 images for a rate of 72.2%. If $s_{\min}$ was set to 0.5 in the proposed approach, the recognition rate was 83.8%, i.e., the proposed approach performed 11.6% better than a method using the Hausdorff distance. Figure 3(e) shows a plot of the forward fraction of the best match returned by the partial Hausdorff distance versus the visibility of the model in the image. The wrong matches either have a forward fraction of 0 or close to 0.5. Figure 3(f) displays the position errors of the best matches. Note that in some instances the best match was more than 200 pixels from the true location.

To test the subpixel accuracy of the proposed approach, the IC was mounted onto an table that can be shifted with an accuracy of 1 μm and can be rotated with an accuracy 0.7’ (0.011667°). In the first set of experiments, the IC was shifted in 10 μm increments, which resulted in shifts of about 1/7 pixel in the image. A total of 50 shifts were performed, while 10 images were taken for each position of the object. The IC was not occluded in this experiment. The pose of the object was extracted using the extrapolation of Section 3 and the least-squares adjustment of Section 4 using one and three iterations. To assess the accuracy of the extracted results, a straight line was fitted to the extracted model positions. The residual errors of the line fit, shown in Figure 4(a), are an extremely good indication of the achievable accuracy. The errors using the extrapolation are never larger than 1/22 pixel. What may seem surprising at first glance is that the position actually gets worse when using the least-squares adjustment. What can also be noted is that the errors show a sinusoidal pattern that corresponds exactly to the pixel size. This happens because the IC is moved exactly vertically and because the fill factor of the camera (the ratio of the light-sensitive area of each sensor element to the total pixel area of each sensor element) is much less than 100%. Because of this, the subpixel edge positions do not cause any gray value changes whenever the edge falls on the light-insensitive area of the sensor, and hence the subpixel edge positions are not as accurate as they could be when using a camera with a high fill factor. Unfortunately, in this example, the object’s edge positions are such that their location is highly correlated with the blind spots. Hence, this effect is not unexpected. When cameras with high fill factors are used the accuracy when using the least-squares adjustment is significantly better than when using the extrapolation.

To test the accuracy of the extracted angles, the IC was rotated 50 times for a total of 5.83°. Again, 10 images were taken in every orientation. The residual errors from a straight line fit, displayed in Figure 4(b), show that the angle accuracy is better than 1/12° (5’) for the extrapolation method, better than 1/40° (1.3’) for the least-squares adjustment using one iteration, and better than 1/100° (0.6’) for the least-squares adjustment using three iterations (ignoring the systematic error for very small angles, for which all three methods return the same result; this is probably an error in the adjustment of the turntable that was made when the images were acquired). Since in this case the IC is rotated, the errors in the subpixel positions of the edges caused by the low fill factor average out in the least-squares adjustment, and hence a significantly better accuracy for the angles is obtained using the least-squares adjustment.



(a) Score vs. visibility

(b) Position errors

(c) Pos. err. adjustment (1)

(d) Pos. err. adjustment (3)

(e) Score vs. visibility

(f) Position errors

Figure 3: Extracted scores plotted against the visibility of the object (a) and position errors of the extracted matches for the proposed approach using extrapolation (b) and least-squares adjustment with one (c) and three (d) iterations; scores vs. visibility (e) and position errors (f) using the Hausdorff distance.

A more thorough evaluation of the proposed method, including a comparison to a larger number of algorithms can be found in (Ulrich and Steger, 2001).

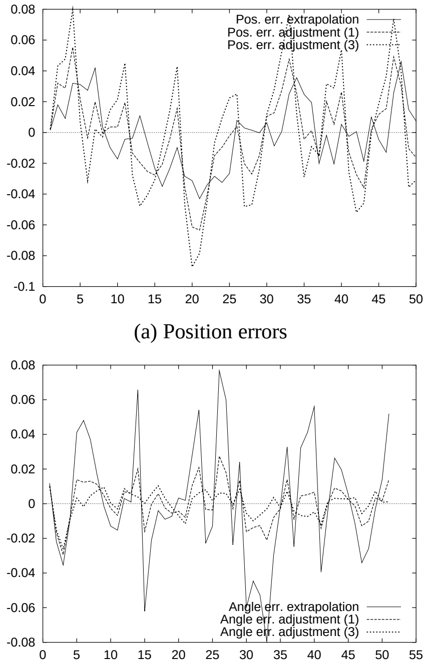
7 CONCLUSIONS

A novel object recognition approach for industrial inspection using a new class of similarity measures that are inherently robust against occlusion, clutter, nonlinear illumination changes, and global as well as local contrast reversals, has been proposed. The system is able to recognize objects under similarity transformations in video frame rate. A performance evaluation shows that extremely high object recognition rates (more than 98% in the test data set) are achievable. The evaluation also shows that accuracies of 1/22 pixel and 1/100 degree can be achieved on real images.

REFERENCES

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.



(a) Position errors

(b) Orientation errors

Figure 4: Position errors of the extracted model poses (a); orientation errors of the extracted model poses (b).

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

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), pp. 679–698.

Förstner, W., 1994. A framework for low level feature extraction. In: J.-O. Eklundh (ed.), Third European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 801, Springer-Verlag, Berlin, pp. 383–394.

Lai, S.-H. and Fang, M., 1999. Robust and efficient image alignment with spatially varying illumination models. In: Computer Vision and Pattern Recognition, Vol. II, pp. 167–172.

Olson, C. F. and Huttenlocher, D. P., 1997. Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing 6(1), pp. 103–113.

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

Steger, C., 1998. An unbiased detector of curvilinear structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(2), pp. 113–125.

Steger, C., 2000. Subpixel-precise extraction of lines and edges. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, part B3, pp. 141–156.

Tanimoto, S. L., 1981. Template matching in pyramids. Computer Graphics and Image Processing 16, pp. 356–369.

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