A New Contour-based Approach to Object Recognition for Assembly Line Robots

Markus Suing, Lothar Hermes, and Joachim M. Buhmann

Institut für Informatik III, Universität Bonn
Römerstr. 164, D-53117 Bonn
{suing, hermes, jb}@cs.bonn.edu

Abstract. A complete processing chain for visual object recognition is described in this paper. The system automatically detects individual objects on an assembly line, identifies their type, position, and orientation, and, thereby, forms the basis for automated object recognition and manipulation by single-arm robots. Two new ideas entered into the design of the recognition system. First we introduce a new fast and robust image segmentation algorithm that identifies objects in an unsupervised manner and describes them by a set of closed polygonal lines. Second we describe how to embed this object description into an object recognition process that classifies the objects by matching them to a given set of prototypes. Furthermore, the matching function allows us to determine the relative orientation and position of an object. Experimental results for a representative set of real-world tools demonstrate the quality and the practical applicability of our approach.

Keywords: Object Recognition, Shape, Mesh Generation, Model Selection, Robotics

1 Introduction

Object manipulation with assembly line robots is a relatively simple task if the exact type, position and spatial alignment of objects are fixed or at least known in advance. The difficulty of the problem increases significantly if the assembly line transports a large set of various different types of objects, and if these objects appear in arbitrary orientation. In this case, the successful application of robots crucially depends on reliable object recognition techniques that must fulfill the following important requirements:

Reliability: The algorithm should be robust with respect to noise and image variations, i.e. its performance should not degrade in case of slight variations of the object shape, poor image quality, changes in the lightning conditions etc.

Speed: The online scenario of the application implies certain demands on the speed of the image recognition process. It must be able to provide the result of its computations in real time (compared to the assembly line speed) so that the robot can grasp its target from a slowly moving assembly line.

B. Radig, S. Florczyk (Eds.): Pattern Recognition - 23rd DAGM Symposium, pp. 329-336, LNCS 2191, Springer, 2001.

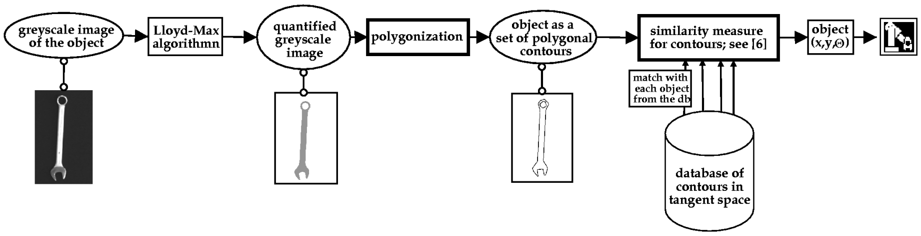
greyscale image of the object
Lloyd-Max algorithmn
quantified greyscale image
polygonization
object as a set of polygonal contours
similarity measure for contours; see [6]
object (x,y,Θ)
match with each object from the db
database of contours in tangent space

Fig. 1. The process pipeline. The system is trained by processing a single picture of each object and storing its contour in a database. By taking snapshots of the assembly line, it then automatically locates objects and describes their contours by a polygonal mesh. After matching them to the database objects, it is able to determine their type, position, and orientation, which are passed to the robot controller.

Automation: The process should be completely automated, i.e., it should work reliably without any additional user interaction.

In this paper, we describe a combination of algorithms that match the above requirements (fig. 1). It assumes that the robot is equipped with an image database in which a variety of different objects is stored by single prototype images. Using a camera which is mounted above the assembly line, the robot is expected to locate and identify by-passing objects, to pick them up and to put them to their final destination, thereby performing a task like sorting. The object recognition strategy proposed here is a two-stage process. The first stage locates individual objects and describes them by polygonal shapes. It is able to cope with almost all kinds of shapes, including those with holes or strong concavities, which is an important advantage over other contour descriptors such as many active contour approaches [3]. The second stage matches these shapes to the prototypes, detects the type of the objects, and computes their relative orientation.

Alternative approaches to image segmentation combined with polygonization typically depend on an initial edge detection processes and do not offer any strategies to effectively control the precision of the triangular approximation in a statistically well-motivated manner [9, 10]. Other object vision systems for assembly line robots require 3D models of each object, which they iteratively fit to 3D information from the scene [2, 11]. A similar approach was followed in [12] where 2D contours were used to classify objects based upon [1].

2 Polygonal Image Segmentation

Formally, we describe an image by a function $I(o)$ that assigns each possible position o_i to a pixel value $I(o_i)$. In our current implementation, we just operate on binary images, i.e. $I(o) \in \mathbb{B}$, but the theoretical framework also applies to the multi-valued case. The image is assumed to consist of several areas a_λ, each being characterized by a homogeneous distribution of pixel values. We aim at finding a decomposition of the image $I(o)$ into segments $\hat{a}_\nu$ that are (at least after consistent renaming of their indices) in good accordance with the areas a_λ.

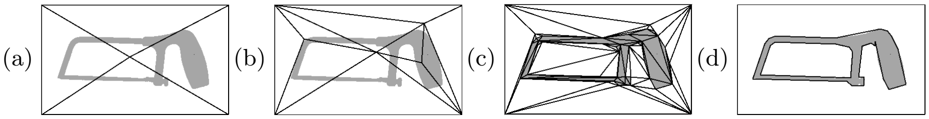

Fig. 2. (a) Binarized input image with superimposed initial mesh. (b) Early optimization stage after few splits. (c) Final triangulation. (d) Corresponding polygonal shape.

Thus the desired output of our algorithm is a function $\hat{a}\left(o\right)$ that, up to the best possible degree, mirrors the composite structure $a\left(o\right)$ of the image.

We assume that the image formation can be described by a simple generative model: First an image site o_i is selected according to a distribution $p\left(o_i\right)$ which is assumed to be uniform over all pixel positions. The site o_i is then assigned to an area $a_\lambda = a\left(o_i\right)$. Depending exclusively on this area information, the image site is finally provided with a pixel value $I_\mu = I\left(o_i\right)$ according to the conditional distribution $p\left(I_\mu | a_\lambda\right)$. Replacing a_λ by their estimates $\hat{a}_\nu$ therefore yields

$$p\left(o_i, I_\mu | a_\lambda\right) = p\left(o_i\right) \cdot p\left(I_\mu | a_\lambda\right) = p\left(o_i\right) \cdot p\left(I_\mu | \hat{a}_\nu\right) \quad . \tag{1}$$

According to [6], the latent parameters $\hat{a}\left(o\right)$ should maximize the *complete data likelihood*, which – when assuming statistically independent pixel positions – can be written as $\mathcal{L} = \prod_i p\left(o_i, I\left(o_i\right), \hat{a}\left(o_i\right)\right) = \prod_i p\left(o_i, I\left(o_i\right) | \hat{a}\left(o_i\right)\right) \cdot p\left(\hat{a}\left(o_i\right)\right)$. Inserting (1) and dropping constant factors, we obtain

$$\mathcal{L} \propto \prod_{\mu,\nu} \left(p\left(I_\mu | \hat{a}_\nu\right) p\left(\hat{a}_\nu\right)\right)^{n\left(I_\mu, \hat{a}_\nu\right)} \quad , \tag{2}$$

where $n\left(I_\mu, \hat{a}_\nu\right)$ denotes the number of occurrences that the pixel value I_μ is observed in segment $\hat{a}_\nu$. The corresponding negative log likelihood per observation is given by $-\frac{1}{n} \log \mathcal{L} \propto -\sum_{\mu,\nu} p\left(I_\mu, \hat{a}_\nu\right) \log\left(p\left(I_\mu | \hat{a}_\nu\right) \cdot p\left(\hat{a}_\nu\right)\right)$, where $p\left(I_\mu, \hat{a}_\nu\right)$ is the probability of a joint occurrence of I_μ and $\hat{a}_\nu$, and n is the total number of observations. $-\frac{1}{n} \log \mathcal{L}$ can be decomposed into two parts. The first part is the *conditional entropy* of the pixel values $I\left(o_i\right)$ given their assignments to polygons $\hat{a}\left(o_i\right)$. The second part is the *entropy* of the a-priori distribution for the polygons, which is discarded to avoid any prior assumptions on the size of individual polygons. We arrive at the cost function

$$H\left(\hat{a}\right) = -\sum_{\mu,\nu} p\left(I_\mu, \hat{a}_\nu\right) \log p\left(I_\mu | \hat{a}_\nu\right) \quad , \tag{3}$$

which is insensitive with respect to consistent renaming of the polygon indices. It can be shown to be minimal for perfect correspondances between the estimated and the true segmentations $\hat{a}\left(o_i\right)$ and $a\left(o_i\right)$, respectively. Besides it is concave in $p\left(\hat{a}_\nu, a_\lambda\right)$, which has the advantage that there are no local minima inside the probability simplex.

We represent the segmentation $\hat{a}\left(o\right)$ by a triangular mesh [5], which is refined by a hierarchical optimization strategy. Starting with 4 initial triangles, we iteratively add new vertices to achieve a finer resolution. Once a new vertex v_λ has been added, it is moved to the position where it causes the minimal partial costs with respect to (3). During this optimization, the movement of the vertex has to be restricted to the convex polygon that is formed by the straight lines connecting its adjacent vertices. Under certain preconditions, however, this constraint can be circumvented by *swapping* individual edges, which gives the algorithm additional flexibility. After having found the optimal position for the new vertex, all adjacing vertices are inserted into a queue from which they are extracted for further optimization (fig. 2).

The described algorithm can be implemented as a multiscale variant by downsampling the original image into several resolution levels, optimizing the mesh on a coarse level, mapping the result onto the next finer level, and continuing the optimization there. In this case, however, one has to detect at which granularity the current image resolution is too low to justify any further mesh refinement. For the definition of an appropriate decision criterion, it is important to note that the cost function (3) does not take the exact position of individual pixels into account. Instead, it completely relies on the joint distribution of grey values and image segments. From Sanov's theorem [4], one can thus infer that the probability of measuring a certain cost value H^* is completely determined by the minimal KL-distance between the generating model $p\left(I_\mu, a_\lambda\right)$ and the set of all empirical probability distributions $q\left(I_\mu, \hat{a}_\nu\right)$ for which the cost value H^* is obtained. It can be shown that, among these distributions, the one with minimal KL divergence has the parametric form

$$q^*\left(I_\mu, a_\lambda\right) \propto p\left(\hat{a}_\nu\right) \cdot p\left(I_\mu \middle| \hat{a}_\nu\right)^\beta \quad . \tag{4}$$

The correct value of β can easily be found by an iterated interval bisection algorithm. According to Sanov's theorem, this leads to the probability estimate

$$Pr\left\{H = H^*\right\} \approx 2^{-nD(q^*||p)} \quad . \tag{5}$$

It can be used to compute the probability that a previous model generates an image with the same costs as the costs measured for the actual image model. If this probability is above a threshold p^{stop}, the optimization algorithm decides that the optimization progress might also be due to noise, and switches to the next finer resolution level in which the respective grey value distributions can be estimated with higher reliability. If it already operates on the finest level, adjacing triangles that share their dominant grey value are fused into larger polygons, which are then passed to the shape matching algorithm.

3 Application to Object Recognition

For the shape matching, we employ the algorithm described in [14], which has been found to feature both high accuracy and noise robustness. It first maps the shapes onto their normalized tangent space representations, which has the advantage of being invariant with regard to position and scale. After smoothing

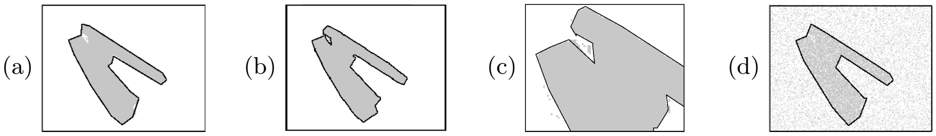

Fig. 3. (a), (b): Stapler segmented with $p^{stop} = 0.75$ and $p^{stop} = 0.99$, respectively. (c) Close-up of the fitted object boundary. (d) Result at a noise level of 40%.

the shapes by an appropriate shape evolution process [13], they are divided into their maximal convex and concave sub-arcs. This is motivated by the fact that visual object parts relate to the convex sub-arcs of the object shape. Based on this decomposition of shapes the similarity of two shapes is defined as a weighted sum over a set of many-to-one and one-to-many matching of consecutive convex and concave sub-arcs.

With this similarity measure, the type of a query object can be determined by retrieving the most similar prototype in the image database. In order to facilitate the subsequent grasp operation, the database also contains an adequate gripper position (*grasp point*) and orientation for each prototype (fig. 4 a). To initiate the grasping of the query object, we thus have to compute its rotation and translation parameters with respect to its prototype. Here we can take advantage of the fact that, to find the optimal matching between two polygonal sub-arcs a_i and b_i in tangent space, the algorithm described in [14] implicitly rotates a_i by the angle ϕ_i which minimizes the squared Euclidian distance $\int_0^1 \left(a_i\left(t\right) - b_i\left(t\right) + \phi_i\right)^2 dt$ between a_i and b_i. The shape similarity measure ignores the angles ϕ_i and is thus able to handle rigid object rotations and flexible joints. Here, however, we propose to compute the α-truncated mean of ϕ_i, which gives us a robust estimate ϕ of the relative orientation of the whole contour. To localize the grasp point on the query shape, the boundary positions of the convex and concave sub-arcs are used as reference points (fig. 4 b). Let p_d denote the grasp point for the database object, x_i the associated reference points, and y_i the reference points on the query shape. In addition define δ_i as the Euclidean distance between x_i and p_d, and d_i as the corresponding distance between y_i and the current grasp position on the query shape, p_q. Our approach is to choose p_q such that the d_i give the best fit to the corresponding δ_i. This problem setting is similar to the problem of multidimensional scaling (MDS), in which you are given pairwise dissimilarity values for a set of objects, which have to be embedded in a low-dimensional space. To find p_q, we therefore adapt the SSTRESS objective function for MDS [7] and minimize $J = \sum_i (\frac{d_i - \delta_i}{\delta_i})^2$, which can be achieved by a standard gradient descent algorithm. Note, however, that each d_i corresponds to a partial contour match and therefore to a rotation angle ϕ_i. If ϕ_i is significantly different from the rotation angle ϕ of the whole object, then the corresponding d_i should not exert a strong influence on the cost function. We, therefore, downweight each term in J by $\exp\left(-\lambda \sin^2(\mathit{\Delta}\phi_i/2)\right)$ with $\mathit{\Delta}\phi_i := \phi_i - \phi$. Besides it is possible to sample additional positions along the sub-arcs to obtain additional reference points, which increases the robustness of the algorithm.

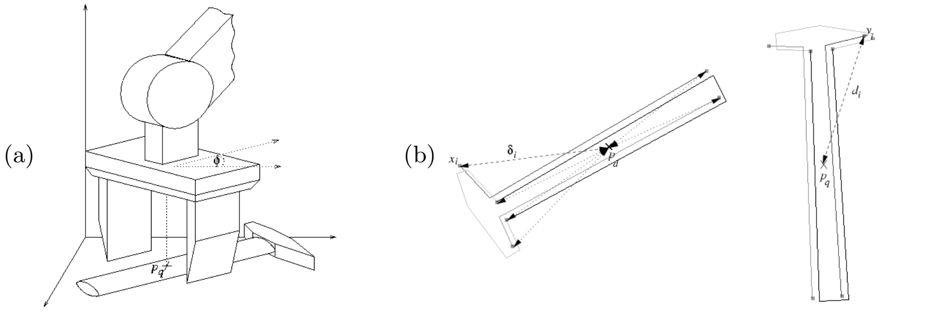

Fig. 4. (a) To adjust the position and orientation of the two-finger gripper, the system adapts the specications for the prototype in the database. (b) The grasp points of database objects (left) and query (right) objects are defined by their relative distances from a set of control points, which are by-products of the shape matching algorithm.

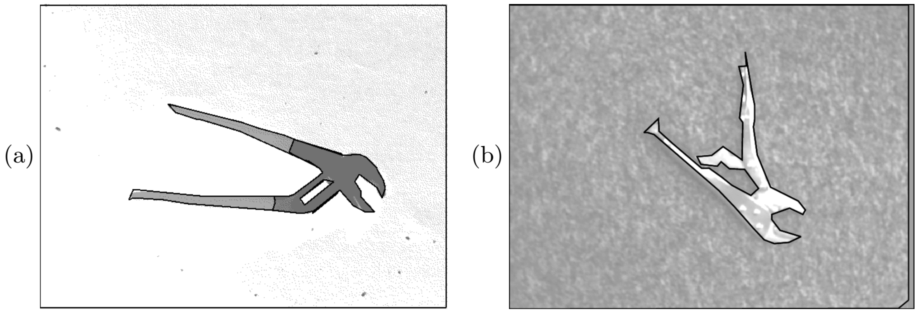

Fig. 5. (a) Segmentation with 3 quantization levels, run on a dithered camera image. (b) Segmentation with a textured background, demonstrating that the segmentation algorithm is not restricted to high contrast situations. Here a Floyd-Steinberg dithering with 32 quantization levels was used for preprocessing.

4 Experimental Results and Discussion

For a first evaluation of the system, a set of 6 different tools was chosen from a standard toolbox, including two rather similar saws, two wrenches, an allen key, and a stapler. From each object, 10 images were taken that showed the object in different positions and orientations. The images were binarized using a variant of the Lloyd-Max-Algorithm [15]. From each object, we randomly selected one image for the prototype database, while the remaining images were joined to form the validation set. The performance was evaluated using 10-fold crossvalidation.

In this scenario, we obtained an object recognition performance of 99.6% with a standard deviation of 0.8%. We also measured the precision of the orientation estimation and measured an average error of 2.2° with a standard deviation of 3.2° compared to visual inspection. The average error for the grasp point estimate was 5 pixels. The polygonization runs on a 1Ghz PC on images with a resolution of 256x256 pixels in less than five seconds. There are still several possibilities to fasten it up significantly, namely stopping the multiscale optimization at a lower stage, finding a better initial mesh and working with areas of interests.

The parameter p^{stop} has been found to be a powerful tool for controlling the accuracy with which the mesh is fitted to the object. See fig. 3 (a), (b) for two results with different p^{stop} values, and fig. 3 (c) for a close-up view of a generated contour. The availability of a reliable tool like this is important to cut down the computational complexity, because the object should not be described with a higher precision than necessary. A too complicated contour has to be simplified afterwards by the curve evolution approach described in [14]. Besides, the algorithm could be shown to remain very robust in the presence of noise. Fig. 3 (d) shows a test image in which 40% of the pixels had been set to random values. It demonstrates that the segmentation result remains nearly unaffected and still produces an accurate result.

With slight modifications, the segmentation algorithm is also capable of processing non-monochrome objects or objects on textured backgrounds (fig. 5). Instead of the Lloyd-Max quantization [15], one can alternatively use the Floyd-Steinberg dithering algorithm [8], which has the additional advantage of avoiding undesirable edges at the boundaries of homogeneously quantized image regions.

5 Conclusion and Future Work

We have presented a new computer vision method for analyzing objects on an assembly line. To automatically extract all information that is necessary to grasp and e.g. sort the objects by a robot, it employs a new fast and robust image segmentation algorithm that locates the objects and describes them by sets of polygons. These object representations are then used to compare the objects to a given set of prototypes, to recognize their type, and also to compute their exact position and orientation. Real-world experiments show that the proposed object recognition strategy produces high-quality results and matches all demands of the application (in terms of speed, robustness, and automation).

Our framework offers several promising starting points for further extensions. In our current implementation, we restrict ourselves to Bernoulli distributions (i.e. binarized images). The generative model, however, is valid for arbitrary discrete probability distributions, and can thus also be applied to images where a larger number of possible grey-values is retained. When applied to these images, the segmentation algorithm is able to describe individual objects as structures with several adjacent or even nested parts. Although we currently use only the outer contour of an object as its descriptor, we expect that there is a generic extension of Latecki's shape similarity concept to compound objects, which will be in the focus of our future research as well as speed improvements and the dependance of the performance of the polygonization on the p^{stop} parameter. Further work will also include the integration of the algorithms into an operational automated production system.

Acknowledgements: This work has been supported by a research grant from Astrium GmbH, Friedrichshafen. The implementation and evaluation of the algorithm were done in cooperation with the `mz` robotics laboratory, Rheinbach.

References

[1] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchel. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Image Processing and Machine Intelligence*, 13(3):209–215, 1991.

[2] Gernot Bachler, Martin Berger, Reinhard Röhrer, Stefan Scherer, and Axel Pinz. A vision driven automatic assembly unit. In *Proc. Computer Analysis of Images and Patterns (CAIP)*, pages 375–382, 1999.

[3] A. Blake and M. Isard. *Active Contours*. Springer, 1998.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[5] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 1997.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[7] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd ed. edition, 2000.

[8] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial gray scale. In *Society for Information Display 1975 Symposium Digest of Technical Papers*, page 36, 1975.

[9] Miguel Angel Garcia, Boris Xavier Vintimilla, and Angel Domingo Sapa. Approximation and processing of intensity images with discontinuity-preserving adaptive triangular meshes. In David Vernon, editor, *Computer Vision: ECCV 2000*, volume 1842 of *Lecture Notes in Computer Science*, pages 844–855. Springer, 2000.

[10] T. Gevers and A. W. M. Smeulders. Combining region splitting and edge detection through guided delaunay image subdivision. In J. Ponce D. Huttenlocher, editor, *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1021–1026. IEEE Press, 1997.

[11] F. Keçeci and H.-H. Nagel. Machine-vision-based estimation of pose and size parameters from a generic workpiece description. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2001.

[12] S. Kunze and J. Pauli. A vision-based robot system for arranging technical objects. In *Proc. of the International Symposium on Automotive Technology and Automation (ISATA)*, pages 119–124, 1997.

[13] Longin Jan Latecki and Rolf Lakämper. Polygon evolution by vertex deletion. In M. Nielsen, Peter Johansen, Ole Fog Olsen, and Joachim Weickert, editors, *Scale Space 99*, volume 1682 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 1999.

[14] Longin Jan Latecki and Rolf Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on PAMI*, 22(10):1185–1190, October 2000.

[15] S. P. Lloyd. Least squares quantization in pcm. Technical report, Bell Laboratories, 1957.