# Two Machine Vision Strategies for Robotics 

Pentcho Venkov


#### Abstract

Two basic approaches are implicated in a computer vision system, designed to identify and locate flat shapes for robotics applications. If the contrast of the image is high enough a binary description is obtained by using global or local thresholds. The regions of the objects are located and recognized by connectivity analysis. The scene is presented by a labeled directed graph. In general conditions, when partial occlusion is allowed and no special care is taken for the illumination, algorithms for edge detection are applied, combining gradient operators with local maxima detection. The segmentation stage is based on procedures for curvature estimation. The recognition stage is based on matching the segmented descriptions of the scene with contour models by a technique of hypothesis generation and verification, coupled with a recursive estimation of the object position in the scene.


Keywords: computer vision, connectivity analysis, contour models, robots.

## I.Introduction

Generalized image analysis and interpretation iz still one of the most challenging and important research areas in robotics and machine intelligence. Although most current research is concerned with recognizing 3D shapes in grey scale images, taken for an arbitrary perspective, some results dealing with flat shapes confined to a plane, are being used in practical robot vision systems for a variety of applications. The main reason for this is the compactness of representation and the simplicity of analysis algorithms that result from structured scenes of the production environment [1\}, [2].
This paper describes a vision system for robots, designed to handle 2D shapes of multiple randomly oriented objects [5]. The acquisition device is a CCD camera, connected to the computer memory via a Matrox-Meteor frame-grabber. The basic software products are designed for achromatic gray level images, containing either $256 \times 256$ pixels for medium resolution tasks or $512 \times 512$ pixels for higher resolution applications. The use of RGB images is involved in the preprocessing stage to simplify the identification and extraction of colored objects from complex scenes. The basic functions of the system are presented by a modular structure of menus, written in Visual $\mathrm{C}^{++}$. These menus combine operators of the processing, analysis and recognition stages, designed for binary, grey level and color images. They allow implementing the approach of connectivity analysis of regions or the strategy of contour modeling and recognition. In the experimental stage the vision system was coupled with desktop or mobile robots.
P.Venkov is with the Technical University of Sofia, 1756 Sofia, 8, Kl.Ohridsky bld, e-mail: pven@vmei.acad.bg

## II. REGION SEGMENTATION AND MODELING

In many robot vision tasks, such as classifying and locating parts on a conveyor belt, the designer can structure the environment to control the illumination, viewing angle and background composition. Under these circumstances the vision system can distinguish target objects from the background on the basis on contrast or color differences. By dyadic subtraction of the achromatic components of an RGB image one can detect the shape and the position of colored objects or signs in complex scenes. The binary description is defined by using automatic histogram analysis and threshold detection of grey level images. [4].
The first approach is aimed to partition a binary image into regions or blobs that correspond to objects, holes and background in the scene. The blobs are composed of connected pixels, which are all the same color (i.e. black or white). The method of finding blobs and building the linked list of blob descriptors of a scene with many randomly oriented objects is the connectivity analysis [4]. The blob descriptors are records of global features of the shape: color, area, centroid, perimeter, moment invariants, number of holes and other shape features. The proposed algorithm examines the local connectivity around each pixel of the image, building up the related blob descriptors on a pixel-by-pixel basis. In addition the algorithm determines links which superimpose a hierarchical structure on the blobs to present nesting relationships. A compact way to represent these relationships is a labeled directed graph. The arrows of the graph describe the "parent", "child" and "sibling" relations between blobs, and the labels link the blobs with the records, containing the descriptors (Fig.1).


Fig.1.Structural model of a binary image

## III.SHAPE RECOGNITION

The labeled structural model of the scene involves a twostage process of object classification [4]. The first stage
named "row" classification uses structural information in the directed graph to separate the objects from the scene and classify them in accordance to their structural sub-models. The number of holes of every sub-model defines the dimension of the feature vector of his shape, used for final or "precise" classification. In the second stage of final classification the patterns of classes are presented in a multi-dimensional feature space by clusters of samples, gained by a training session of the system. The goal of the recognition rule is to verify if a deterministic or a probabilistic distance from the feature vector of the classified object to the center of a pattern cluster is less or equal to a predefined decision threshold, named radius of the current class.

## IV. EDGE DETECTION

Edges are significant local features of the patterns. They occur on the boundary between different regions, associated with a discontinuity in the image intensity or its first derivative. Edge detection is the first step of the second more general contour strategy for recovering information concerning the type and the localization of objects. The term edge in this work is used for edge fragment. An edge fragment correspond to the coordinates $[i, j]$ of the edge and its orientation $\varphi$, related with the angle of the gradient vector. The contour is an ordered
list of edges by traversal in a pre-defined direction. Contours are defined by an edge-linking algorithm from an unordered set of edges, produced by an edge detector.
The developed algorithms for edge detection realize the following steps:
1.The filtering is used as an optional first step to improve the performance of edge detectors. We use $3 \times 3$ and $5 \times 5$ Gaussian or median filters for images with significant noise level, remembering the trade-off between edge strength and noise reduction [4].
2.The second step is the calculation of the intensity changes in the neighborhood of a point by a gradient and/or a Laplace operator. To compute the gradient magnitude one can use Sobel or Prewitt operator. Some slight difference in the performance of their convolution masks is related with an emphasis on pixels that are closer to the center of the mask in Sobel operator. The both operators compute the magnitude as a sum of absolute values of $\Delta_{x}$ and $\Delta_{y}$ component of the gradient vector to determine the position of edge points in the images. The presence of an edge point is assumed if the magnitude of the gradient is above a global threshold. This results in detection of too many edge points. A better approach would be to detect points that have local maxima in gradient values and consider them as points of the contour. This means to find edge fragments and to use the orientation of the gradient vector for local maxima detection. We develop an algorithm of Canny-like edge detector [5]. After a smoothing stage with a Gaussian filter, the algorithm computes the gradient orientation into the eight directions of the chain code with a set of $3 \times 3$ step approximation masks, defined by the Kirsch operator. In order to reduce the number of false edge fragments the algorithm applies a global threshold to the gradient magnitude. All values below the threshold are changed to zero. The magnitude and the orientation of the edge fragments in four possible directions (pairs of codes) allow to identifying
the edges that are local maxima in the gradient magnitude array. This process, called nonmaxima suppression, results in thinned contours.
3.The Laplacian operator is highly affected by noise. The edge detection menu proposes a $5 \times 5$ Laplacian of Gaussian mask due to Marr and Hilderth [3] to find the zero crossing in the edge points, whose corresponding gradient magnitude is above some global threshold.
The performance of the edge detectors, mentioned above, has been evaluated with a number of test images of real objects received at different lighting conditions. A camera against a bright background (back-illumination screen) views some of the objects, while another are received against a dark background. One can detect thin contours by using the Canny-like edge detector or the Laplacian of Gaussian operator in images with relatively high contrast even if the background representation is not uniform. If the background is uniform, the best way to detect contours is to convert the gray level image in binary image by using automatic histogram analysis [4]. The operators for contour extraction in binary images are rather smart and reliable. They need limited computer resource [1].

## V.Contour Tracking

The edge following algorithm, used in our machine vision system, scans the image to find an edge segment used as a starting point for the boundary following. The basic approach is that in the current edge point the algorithm looks at its neighbors. If the direction of nearest edge fragment is compatible with that of the current edge, they are linked. Incompatible edges are removed. By looking at an increasing neighborhood in front of the contour, one may fill in missing edges.
A second task of the edge following algorithm is the generation of chain codes, specifying the direction of the contour at each point. The direction is quantified in one of the eight vectors, leading to the eight neighbor of an edge. Starting at the first edge in the list and going around the contour one receives a compact representation of an edge list, containing the coordinates of the first point and the list of codes leading to subsequent edges. The algorithm calculates the derivative of the chain code, by using the first difference modulus 8 . This difference code is a rotationalinvariant description of the contour. It is the basic local description of the forms at the end of the early stage of image processing, before the contour segmentation and analysis.

## VI.Contour SEGMENTATION AND ANALYSIS

A list of linked edges is the simplest representation of a contour, but is the least compact description for subsequent operations. Fitting an appropriate curve model to the edge increases efficiency by providing an appropriate description. The segmentation stage is based on the approximation of the linked edges by line or curve segments. These segments must passe close to the edge points, but not necessarily exactly through these points.
In the described vision system the segmentation algorithms generate polyline contour models. They fit the
edge list with a sequence of line segments. At the first step, called "raw" segmentation, the ends of each line segment of the contour are selected as vertices by one of the following procedure [5]:

- Edge points with nonzero code, received by filtering the sequence of difference chain codes by a set of masks;
- Edge points selected by finding the local extreme of distances from a current point $C$ to the median $M$ of the line between two equally spaced contour points $A, B$ at the both sides of the current point.
- Edge points selected by a step moving window, which detects the tops of the segments with maximal curvature;
The vertices of the polyline representation at the end of this step are not precisely positioned and their place can change due to the rotation of the shape.
The second step realizes an estimation of the quality of approximation by the criteria of normalized maximal error [7]. A polygonal split and merge algorithm recursively adds vertices, starting with the initial vertice of the "raw" contour and finding the edge point from the segment sublist that is farthest from the current strait line segment. In the same time the algorithm decides adjacent segments to be replaced by a single segment, if the new segment fits the edges with a subthreshold normalized error. The distance between an edge point $\left(\mathrm{x}_{\mathrm{e}}, \mathrm{y}_{\mathrm{e}}\right)$ and a line segment with end points $\left(x_{1}, y_{l}\right),\left(x_{2}, y_{2}\right)$ is computed by:

$$
\begin{equation*}
d=x_{e}\left(y_{1}-y_{2}\right)+y_{e}\left(x_{2}-x_{1}\right)+y_{2} x_{1}-y_{1} x_{2} \tag{1}
\end{equation*}
$$

For a segment with length $L$ the normalized maximum error is:

$$
\begin{equation*}
\varepsilon=\frac{\max _{i}\left(d_{i}\right)}{L} \tag{2}
\end{equation*}
$$

With an appropriate threshold values the split and merge algorithm can determine a stable position of all vertices, invariant with translation and rotation of objects in the scene.
The third step of the analysis determines the polyline models of contours and generates a connectivity graph description of a scene. The record of a contour contains the list of segments with some integral parameters: number of segments, length of the contour etc. Every model $\left(M_{\mathrm{j}}\right)$ or scene $\left(S_{\mathrm{j}}\right)$ segment is described by the coordinates $\left(x_{j}, y_{j}\right)$ of its midpoint, by its length $l_{\mathrm{j}}$ and by two angles: $\alpha_{j}$ - the segment orientation relatively to the axe $X, \theta_{j}$ - the angle with the next segment.

## Vii.Model Based Recognition

At the top stage of the decision the problem is to match in a scene one or more models while allowing for distortion by a similarity transformation, including a translation, a rotation and a scaling. The matching process generates and evaluates a number of hypotheses, related with the contour models of the objects [6].

## A. GENERATION OF HYPOTHESES

To generate a hypothesis is to predict the position of the model in the scene by matching a privileged segment of the model contour $(M C)$ with a segment in a scene contour $(S C)$. The planar transformation $T$, defining the position $\left(x^{*}, y^{*}\right)$ of a model point $(x, y)$ in the scene, is given by the equations:
$x^{*}=t \mathrm{x}+\mathrm{x} \cdot \mathrm{k} \cdot \cos \varphi-\mathrm{y} \cdot \mathrm{k} \cdot \sin \varphi ;$

$$
\begin{equation*}
y^{*}=t y+x \cdot k \cdot \sin \varphi+y \cdot k \cdot \cos \varphi, \tag{3}
\end{equation*}
$$

with $t x, t y$ defining the translation, $\varphi$ - the rotation and $k-$ the scale factor.
The longest (or privileged) segment $M_{0}$ of the $M C$ is defined as compatible with a segment $S_{0}$ of $S C$ if : (1) the angle $\theta_{j m}$ of $M_{0}$ and its preceding neighbor is close to the angle $\theta_{j s}$ of $S_{0}$ and its preceding neighbor, (2) the ratio between the length of $M_{0}$ and $S_{0}$ is close to the estimate of the scale factor $k$. When the segment $M_{0}$ is matched successfully with $S_{0}$, the parameters of $T_{0}$ are computed by resolving (1), (2). If one has an practical estimate of $k$ (e.g. $k=1$ ) he has to compute the three remaining parameters $t x$, $t y, \varphi$ :

$$
\begin{align*}
& \varphi=\alpha_{S}-\alpha_{M}  \tag{5}\\
& t x_{0}=x_{S}-k\left(x_{M} \cos \varphi-y_{M} \sin \varphi\right)  \tag{6}\\
& t y_{0}=y_{S}-k\left(x_{M} \sin \varphi+y_{M} \cos \varphi\right)
\end{align*}
$$

where $\left(x_{S}, y_{S}\right),\left(x_{M}, y_{M}\right)$ denote the coordinates of the midpoints of the segments $S_{0}$ and $M_{0}$. Given some representative segments in the scene and different contour patterns $\left\{M C_{j}\right)_{J=l, \ldots, K}$, a number of hypotheses is generated and ranked by measuring their likelihood, e.g. by the degree of coincidence of $l$ and $\theta$ parameters of $S_{0}$ and $M_{0}$. The best hypothesis is chosen for evaluation. The remainding hypotheses of the list will be evaluated in the case of rejection of the best hypotesis.

## B. EVALUATION OF HYPOTHESES

To evaluate a hypothesis is to take advantage of the predicted position to identify the matching of additional segments of the two descriptions and to refine the predicted position of the model. The matching ends with validation when the current hypothesis reaches a high score of reliability or with rejection, if this score remains under a threshold after the evaluation stage [6].
The matching of aditional segments is produced by an iterative procedure. At every iteration the program select among the not yet examined segments $M_{i}$ of $M C$, beginning with relevant segments at the neighboorhood of the longest segment $M_{0}$, used for calculation of the first estimate of $T_{0}$. A model segment $M_{j}$ is transformed in $M_{j}^{*}$ in the scene by the parameters of $T_{0}$ (3)-(5). A similarity measure $R_{i j}$ is computed between $M_{j}^{*}$ and the nearest segment $S_{i}$ of the scene as a weighted sum ot the entities:

- $\alpha_{i j}=A B S\left(\alpha^{*}{ }_{j}-\alpha_{i}\right)$-the absolute value of the difference in orientation of $M_{j}$ and $S_{i}$;
- $l_{i j}=A B S\left[\frac{l_{j}^{*}-l_{i}}{l_{j}^{*}}\right]$-the relative difference between lengths of $M_{j}$ and $S_{i}$;
- $d_{i j}=\sqrt{\left(x_{j}^{*}-x_{i}\right)^{2}+\left(y_{j}^{*}-y_{i}\right)^{2}}$-the distance between the midpoints of $M_{j}$ and $S_{i}$;

If each of these quantities is below a specified threshold ( $\alpha_{\max }$, $1_{\text {max }}, \mathrm{d}_{\text {max }}$ ) the similarity measure $R_{i j}$ is given by:

$$
\begin{equation*}
R_{i j}=k_{1} \alpha_{i j}+k_{2} l_{i j}+k_{3} d_{i j} \tag{8}
\end{equation*}
$$

where $k_{1}, k_{2}$ and $k_{3}$ are positive weights. For a number of candidates $S_{i}$ in the scene, $M_{j}$ is matched with the segment with the minimum value of $R_{i j}$. Otherwise $M_{j}$ is marked with NIL, which means that it has no homologue in the $S C$.

## VII.UpDATING THE TRANSFORMATION $T$

After each successful match of a model segment $M_{j}$ with its homologue $S_{i}$ in the scene the evaluation algorithm updates the parameters of the transformation $T$. The role of the matched segment in the refining of $T$ components depends of its length, normalized by the length of the contour $M C$. We use only the relatively long - "privileged" segments in the evaluation stage of the current hypothesis. Short segments are more sensitive to noise and can introduce errors in the parameters of $T$.

## IX.THE END OF THE MATCHING PROCEDURE

The procedure of evaluation of the current hypothesis tends to a complete examination of the privileged segments $M_{i}$ of $M C$. It can be stoped when a very high measure of reliability is reached by the hypotesis. The reliability measure is calculated from the difference between the lenght of all the successful matched privileged segments and the lenght of the segments without homologues in the scene, normalised by the perimeter of the contour [6]. If this measure exceeds a threshold before the end of the evaluation procedure, the curent hypothesis is validated, the object is recognized and its position and orientation are defined by the transformation matrix $T$. If the measure of reliability can not reach the threshold before the end of the evaluation stage or decreases bellow a level of likelihood, the current hypothesis is regected. A new hypothesis is took from the priority list (or generated) and the evaluation process is restarted with its new contour model $C M$.

## X.EXPERIMENTAL RESULTS

The vision system described in this article is tested on a large number of different objects with polygonal and round shapes of the silhouette. The shapes are presented in the scene with different position and orientation. The experiments include also a change of lighting conditions in image acquisition. The coupling of the vision system with desktop robot of Feedback, and the mobile robot, allows us to resolve the problem of camera calibration and transformation of the vision coordinates in the robot frame.

## XI.CONCLUSION

The experiments with the vision system help us to generalize some expert rules about the practical implementation of the software resources of the system at the different stages of image analysis and recognition, presented below:

- There is no one image-processing algorithm that stands above all others, in terms of its performance. Practical shape recognition procedures are variable and complex and consist of a sequence of computational steps.
- The design of an appropriate lighting sub-system, the choice of optics, sensors, conveyor belt etc. are an important tool against the computational complexity. If the contrast is high enough and the object is isolated in the scene, one can threshold the image, then smoothes the binary regions and applies the fast and reliable algorithms for contour detection and following.
- The fastest algorithms for contour detection in grey level images, presenting complex scenes with overlapping objects, is due to the use of Canny-like operator, followed by the procedure of nonmaxima suppression.
- The choice among the operators for corner detection at the "raw" level of segmentation is not critical for polygonal silhouettes. The curvature-measuring operator takes advantage in segmentation of round shaped silhouettes.
- The split and merge algorithm is very useful for a refining on the number and the position of all vertices in the polygonal contour description.
- The problem of matching models with scene description is solved by a relatively simple procedure. By coupling the evaluation process with a recursive estimation of the transformation one can achieves a high positional accuracy of manipulation with a robot effector.


## References

[1] T. Pavlidis and S. Horowitz. "Segmentation of plane curves". IEEE Trans. On Computers, C-23 Vol.8, pp. 860-870, August, 1974.
[2] B. Bachelor., Intelligent Image Processing in Prolog. Springer-Verlag, London, 1991.
[3] D. Marr and E. Hilderth, '"Theory of Edge Detection", Proc. Royal Soc. of London. B207, pp.502-511, October 1986.
[4] P. Venkov, Analysis and recognition of images and scenes. Printing House of the Technical University of Sofia, 1996. (in Bulgarian).
[5] P. Venkov, Sensor Devices for Robots. Printing House of the Technical University of Sofia, 2000. (in Bulgarian).
[6] P. Venkov , "Model-guided analysis of grey-level images", Annual of the Technical University of Sofia, Book 4, Vol.48, 1995, pp.379-388.
[7] P. Venkov, "Object identification by contour descriptions", Proc. of the Conf. "Automatics and Informatics 95", Sofia, 1995, pp.220-240. (in Bulgarian).

