Maschine Vision and Applications (2003) 13: 234–244

Machine Vision and Applications

A bin picking system based on depth from defocus

Ovidiu Ghita[1], **Paul F. Whelan**[2]

[1] Vision Systems Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
(e-mail: ghitao@eeng.dcu.ie, Tel.: +353-1-7005869, Fax: +353-1-7005508)
[2] Vision Systems Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland

Received: 2 December 2000 / Accepted: 9 September 2001

Abstract. It is generally accepted that to develop versatile bin-picking systems capable of grasping and manipulation operations, accurate 3-D information is required. To accomplish this goal, we have developed a fast and precise range sensor based on active *depth from defocus* (DFD). This sensor is used in conjunction with a three-component vision system, which is able to recognize and evaluate the attitude of 3-D objects. The first component performs scene segmentation using an edge-based approach. Since edges are used to detect the object boundaries, a key issue consists of improving the quality of edge detection. The second component attempts to recognize the object placed on the top of the object pile using a model-driven approach in which the segmented surfaces are compared with those stored in the model database. Finally, the attitude of the recognized object is evaluated using an eigenimage approach augmented with range data analysis. The full bin-picking system will be outlined, and a number of experimental results will be examined.

Key words: Range sensor – Depth from defocus – Edge linking – Surface matching – Eigenimage analysis

1 Introduction

One task that is commonly found across a broad range of modern integrated manufacturing environments is the need to present parts to automated machinery from a supply bin. To do this safely and efficiently within a flexible robotic environment, it is necessary to know the items identity, location, shape and orientation.

One of the main challenges facing such a bin picking system is its ability to deal with overlapping objects. Initial approaches to this problem were based on modeling parts using 2-D surface representation. Typical 2-D representations include invariant shape descriptors (Zisserman et al. 1994), algebraic surfaces (Kriegman and Ponce 1990) and appearance-based approaches (Murase and Nayar 1995; Ohba and Ikeuchi 1997). Other systems try to recognize the scene objects from the range data using various volumetric primitives such as generalized cylinders (Ponce et al. 1989; Zerroug and Nevatia 1996), polyhedra (Lowe 1987) and local 3-D shape descriptors (Johnson and Hebert 1999). While each approach has its associated advantages and disadvantages, 2-D approaches are generally better suited to planar object recognition. When dealing with non-planar objects, 2-D representations may not provide enough information, hence the need for the incorporation of an additional cue in the form of range data.

In this paper, we describe the implementation of a bin picking system based on depth from defocus. Section 2 outlines the overall system, while Sect. 3 describes the implementation of our range sensor. Section 4 presents the edge-based segmentation algorithm and Sect. 5 describes the object recognition algorithm. This is followed in Sect. 6 by an outline of the pose estimation algorithm. Section 7 presents a number of experimental results illustrating the benefits of the approach outlined in this paper.

Correspondence to: O. Ghita

2 System overview

The operation of the system described in this paper can be summarized as follows (Fig. 1). The range sensor determines the depth structure using two images captured with different focal settings. This is followed by the image segmentation process that decomposes the input image into disjoint meaningful regions. The recognition framework consists of matching the geometrical primitives derived from the segmented regions with those contained in a model database. The region that gives the best approximation with respect to the matching criteria is then referred to the pose estimation algorithm in which the position of the object under investigation is determined by using a *principal component analysis* (PCA) approach in conjunction with range data analysis. Once the object's pose is estimated, the grasping coordinates of the identified object are passed to the bin picking robot.

3 Range sensor

The range sensor employed in this application is based on active depth from defocus approach. This estimates depth by

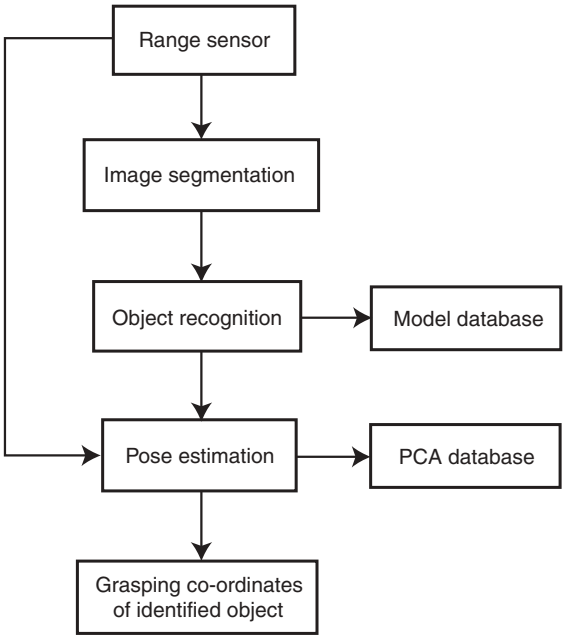


Fig. 1. Overall system architecture

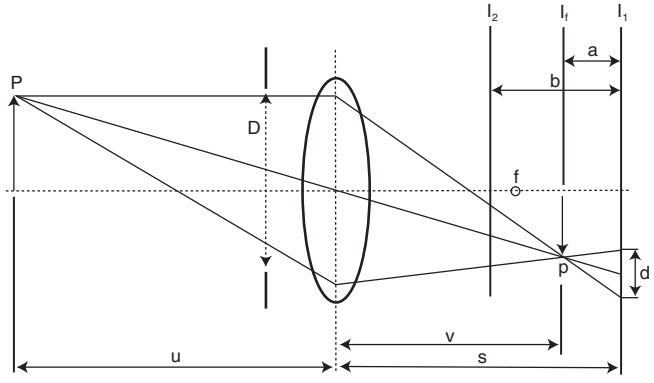


Fig. 2. The image formation process. Depth is determined by measuring the relative level of blurring

measuring the relative blurring between two images captured with different focal settings (referred to as the *near*- and *far*-focused images).

The principle of this range sensor extends from the fact that a lens has a finite depth of field. In this way, if the object to be imaged is placed on the focal plane, the image formed on the sensing element is sharply in focus as every point P from the object plane is refracted by the lens onto a point p on the sensor plane. Alternatively, if the object is shifted from the focal plane, the points situated in the object plane are distributed over a patch on the active surface of the sensing element. As a consequence, the image formed on the sensing element is blurred. From this observation, the distance from the sensor to each point in the scene can be estimated by evaluating the degree of blurring, which is in direct relation to the size of the patch formed on the sensing element (Fig. 2).

Consequently, the diameter of the patch (blur circle) d is of interest and can be easily determined by the use of similar triangles as follows:

$$\frac{D/2}{v} = \frac{d/2}{s-v} \Longrightarrow d = Ds\left(\frac{1}{v} - \frac{1}{s}\right) \tag{1}$$

where v is the focal distance, D is the aperture of the lens and s is the sensor distance. Since the parameter v can be expressed as a function of f and u (Gaussian lens law), Eq. (1) becomes

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \Longrightarrow d = Ds\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right) \tag{2}$$

where u is the object distance and f is the focal length.

It can be observed that d can be positive or negative depending on whether the image plane is behind or in front of the focal plane I_f. This uncertainty indicates that one image is not sufficient to estimate uniquely the level of blurring unless a priori information is available (Pentland 1987; Subbarao and Surya 1994). A solution to solve this uncertainty consists of employing two images separated by a known distance b as illustrated in Fig. 2 (Nayar et al. 1995; Subbarao 1989). This setup enables us to uniquely estimate the depth irrespective of the sign of d.

3.1 Depth estimation using DFD

The blurring effect can be seen as a convolution between the focused image and the blurring function. The blurring function (also referred to as *point spread function*; PSF) can be approximated by a two-dimensional Gaussian function (Pentland 1987; Subbarao and Surya 1994), where the standard deviation σ indicates the level of blurring contained in a defocused image.

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

This model accurately approximates the actual situation and as a consequence the standard deviation (also known as blur parameter) is proportional to the blur circle d. The PSF implements a low-pass filter, thus suppressing the high frequencies (especially those greater than $1/\sigma$). Therefore, to isolate the effect of blurring, it is necessary to extract the high-frequency information derived from the scene. In order to achieve this goal, the near- and far-focused images are filtered with a 5×5 Laplacian operator, where the filtered images can be used to determine the focus level, which is directly related to the blur parameter. One problem with the approach as described is that the high frequencies derived from the scene are directly employed to estimate depth and, as a consequence, this approach cannot be used when dealing with scenes defined by textureless objects. To overcome this problem, a structured light is projected onto the scene, thus imposing an artificial texture and allowing us to calculate the depth by measuring the apparent blurring of the projected pattern. This is referred to as *active* depth from defocus (Nayar et al. 1995; Pentland et al. 1994).

3.2 Physical implementation

Our range sensor must be capable of extracting depth information derived from dynamic scenes. This is implemented using two ITI frame grabbers, thus allowing us to capture both the near- and far-focused images simultaneously. The scene is imaged using an AF Micro Nikkor 60-mm f2.8 lens. A 22-mm

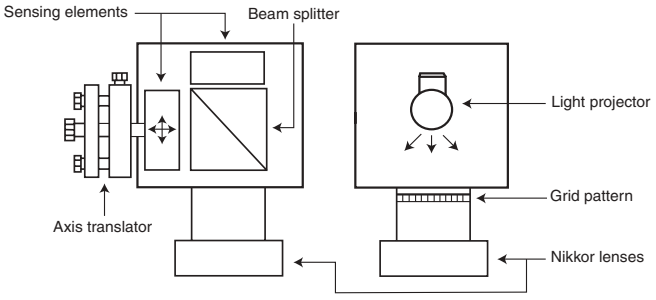


Fig. 3. The bifocal range sensor

beam splitter cube is placed between the NIKKOR lens and the sensing elements. The sensing elements used for this implementation are two low-cost 256×256 VVL CMOS sensors and are precisely positioned to ensure that one captures the near-focused image while the other acquires the far-focused image. The displacement b between the CMOS sensors is set to approximately 0.8 mm (see Fig. 3).

A structured light pattern is projected onto the scene using a MP-1000 projector fitted with a MGP-10 Moire grating (which is a striped grid with a density of 10 lines/mm). The lens attached to the projector is the same type as that used to image the scene.

The range sensor is calibrated using a two-step procedure. The first step involves obtaining a precise alignment between the near- and far-focused sensing elements. This is achieved by employing a calibration pattern consisting of a dense rectangular grid, and the misregistrations between the sensing elements are eliminated using the multi-axis translator attached to one of the CMOS sensors. The second step involves a pixel-by-pixel calibration (Ghita and Wheelan 2001) applied to determine the gain factor, improve linearity and eliminate the depth offsets.

The range sensor computes a depth map in 95 ms on a Pentium 133 with 32 Mb RAM and running Windows 98. The relative accuracy was estimated for successive depth acquisitions (50 cycles) where a planar object was placed at different distances from the sensor. Accuracy was defined by the maximum error between the real and estimated depth values contained in a 32×32 pixel area from the depth map. An accuracy level of 3.4% of the overall ranging distance was achieved when the range sensor was applied to scenes containing non-specular objects (see Fig. 4 for a typical textureless scene). A detailed description of the developed range sensor can be found in Ghita and Whelan (2001).

4 The edge-based segmentation process

Edges are commonly used in image analysis to detect region boundaries. Typically, edges are associated with the sharp transitions in the grey-level distribution. But edges are also determined by abrupt changes in the depth structure. In the former case, the edge information is detected from the intensity images; while in the latter case, the range images are used as input (Hoover et al. 1996; Jiang and Bunke 1999). But which approach gives the better results? Henderson (1983) suggests that in the case of shape representation where the objects of interest are highly textured and the relative depth between the objects in the scene is significant, the scene analysis should be performed on range images. It is important to note that the precision of the range sensor plays a crucial role in this approach. Alternatively, if the objects are small and textureless (as is the case in our application) then the information contained in a range image is not sufficient to achieve accurate segmentation. As a consequence, better results may be obtained if the intensity images are considered as the input to the segmentation algorithm.

The quality of the segmentation process is also related to the precision of the edge operator involved. Although robust edge detection has been a goal of computer vision for many decades, the current range of edge operators fail to correctly recover the entire edge structure associated with a given image. This is due to the presence of image noise (which can generate extraneous edges) and the small variation of the image intensity distribution (which can contribute to gaps in the edges). These facts have a negative influence on the segmentation results, and as an immediate result, the segmentation process will fail to identify the meaningful regions derived from the image under analysis. Therefore to improve the quality of the edge detection stage and achieve meaningful segmentation, further processing that takes into account the local information revealed in the edge detection output has to be considered. A wide range of techniques have been used to addressed this problem, including morphological methods (Casadei and Mitter 1996; Vincent 1993), Hough transform (Gupta et al. 1993), probabilistic relaxation techniques (Hancock and Kittler 1990), multiresolution methods (Bergholm 1987; Vincken et al. 1996; Eichel and Delp 1985) and the use of additional cues such as colour (Saber et al. 1997). In general, morphological approaches offer a fast solution that attempts to maximally exploit the local information. In contrast, multiresolution and multiscale methods try to enhance the edge structure by combining the information contained in a stack of images with different spatial resolutions. In the next section, we present a two-step, morphological-based algorithm that performs edge reconstruction followed by edge linking based on information derived from the edge terminators.

4.1 Sequential edge reconstruction

A general problem related to morphological approaches is the choice of optimal parameters for an advanced edge operator. In order to reduce the spurious responses generated by image noise, the image under investigation is usually smoothed by applying a Gaussian filter (Marr and Hildreth 1980). Hence, the first parameter is the standard deviation σ, a parameter that determines the size and ultimately the scale of the Gaussian operator. To further improve the edge detection output, Canny (1986) proposes a method based on thresholding with hysteresis. This technique evaluates the output of the edge detector using two threshold levels (referred to as the *high* and *low* thresholds) in order to remove the weak edge responses. Shen and Castan (1992) employ a similar approach to develop an optimal edge detector based on an *infinite symmetric exponential filter* (ISEF) for recovering step-like edges. It is important to note that the optimal set of these parameters is dependent on the input image.

In addressing this problem, many researchers attempt to tackle this issue on a global basis by building a stack of images

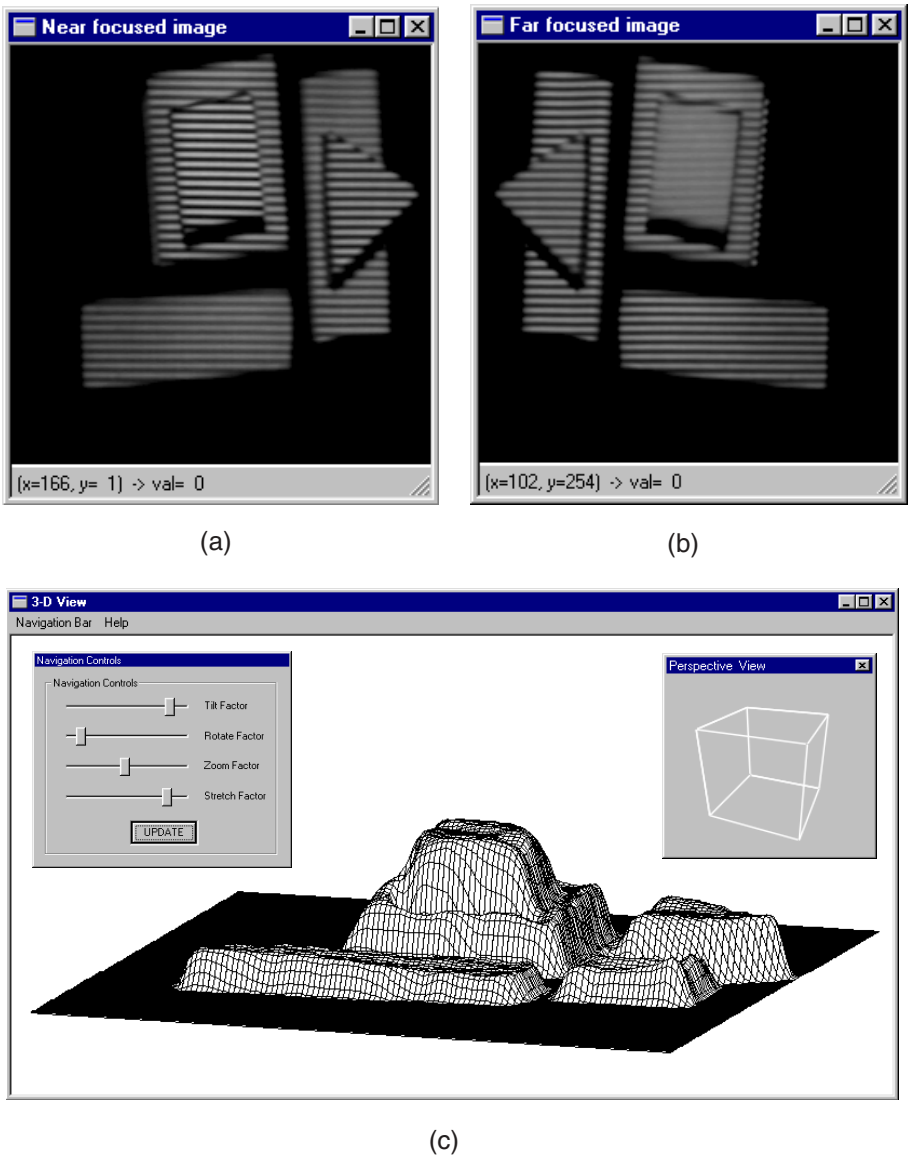


Fig. 4. Sample depth map using active DFD: **a** the near-focused image; **b** the far-focused image; and **c** the resulting depth map

in which the scale parameter is varied. The edge reconstruction scheme consists of aggregating the edge information, starting from images with low resolutions and working towards those with a higher resolution. While this approach is very intuitive, it is computationally expensive as the convolution masks that implement the Gaussian operator become larger as σ increases. Choosing the right scales also presents a difficult problem (Lindeberg 1993). Moreover, the appearance and the localization of edges within the image are increasingly disturbed as σ increases, leading to possible difficulties during the edge reconstruction process.

To avoid the problem associated with multiresolution approaches and to maintain a low computational overhead, we choose to vary the threshold parameters while the scale parameter is set to the default value ($\sigma = 1.0$ for Canny and $a_0 = 0.45$ for the ISEF-based gradient exponential filter (GEF) edge operator). This approach has the advantage that the Gaussian filter and the edge operator has to be applied once, while the hysteretic threshold is sequentially applied in order to obtain the stack of images with different resolutions. (In this context the term *resolution* defines the level of edge detail that is present in the image following clipping the edge image at a given pair of threshold values.)

At this stage a key question is, What criteria should be employed to select the optimal range for the threshold parameters? To answer this, it is necessary to analyze the length of the edge segments contained in the edge detection output. Our goal is to maximize the length of the edge segments. Small isolated segments (less than 4 pixels when dealing with images of 256×256 resolution) are generally due to noise. In this regard, we propose to select the low threshold by analyzing the level of small edge segments that are present in the edge detection output. The algorithm is initialized with the minimal value for the low threshold. (During this stage, the high and low threshold values are set to the same value.) This value is incrementally increased until the ratio between the number of edge pixels derived from small edge segments and the number of pixels derived from large edge segments is less than a preset value. When this criterion is upheld, the low threshold is fixed, and by increasing the value of the high threshold, a

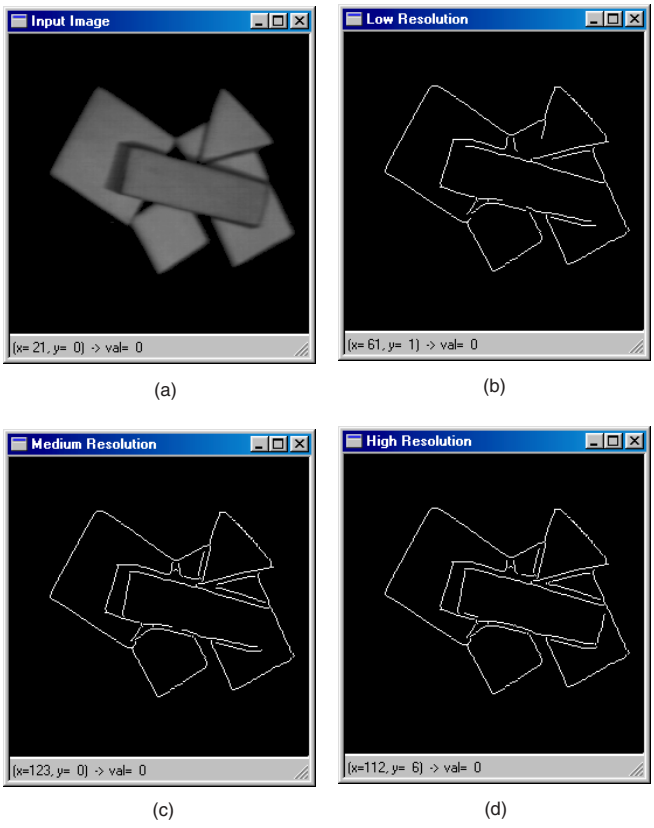


Fig. 5. The image stack: **a** an input image the low-resolution image **b** the medium-resolution image; and **c** the high-resolution image

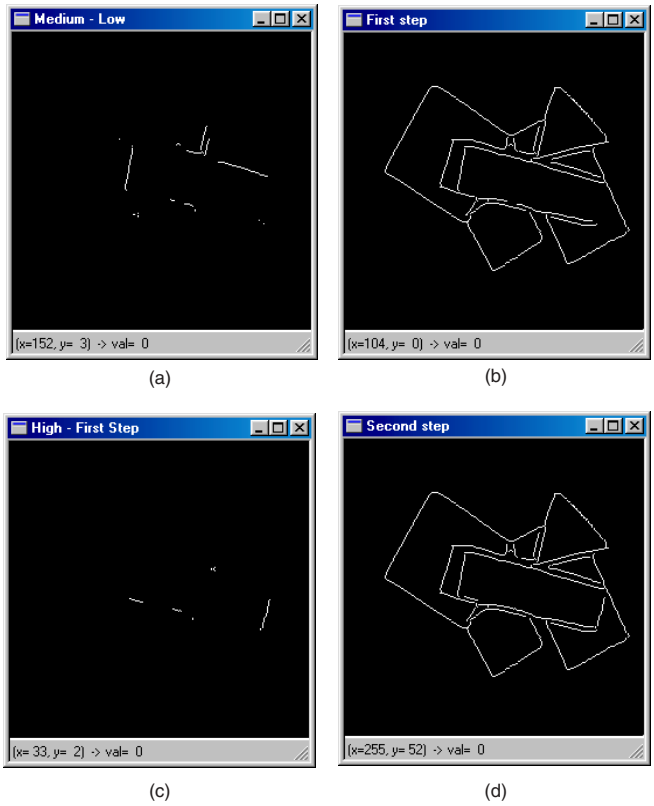


Fig. 6. The edge reconstruction process for images illustrated in Fig. 5: **a** difference between the medium- and low-resolution images; **b** the resulting image after first iteration **c** difference between (**b**) and the high-resolution image; and **d** the output image

coarser resolution image is obtained. The maximum value of the high threshold is dependent upon the edge detector employed (for example, it takes a value of 20 for the GEF edge operator). For this implementation, an image stack that contains three images of different resolutions (referred to as *low-*, *medium-* and *high-resolution* images) was considered.

Once the image stack is processed (see Fig. 5), the edges are reconstructed by analyzing the edge structure of the images contained in the image stack. Initially, edges are combined between the low-resolution image and the medium-resolution image. The reconstruction process consists of analyzing the image resulting after the subtraction of the low-resolution image from the medium-resolution image. The edge segments that are contained in this difference image are then added to the output image if they are connected to the edge structure in the low-resolution image or their length is greater than 4 pixels. When this process is completed, the output image is subtracted from the high-same procedure outlined above. Fig. 6 illustrates the results obtained after the application of the proposed edge reconstruction scheme. Note that this approach allows the removal of noise at each iteration.

4.2 Edge linking

Although the proposed edge-reconstruction scheme significantly enhances the edge structure, there are situations where gaps in edges exist in the output image. In order to eliminate these errors, we propose to bridge the gaps in edges by analyzing the pixels around the edge terminators. The operation required to extract the edge terminators involves a simple morphological analysis by the application of a set of 3×3 masks to the image resulting from the edge reconstruction process (Ghita 2000). The next step involves determining the scanning direction for each edge terminator by analyzing the edge structure that generates it. The possible edge paths between unlinked edges are determined by evaluating the pixels situated in the edge terminator's neighborhood according to its direction. If a connection is detected, an edge path is established between the edge terminator and the detected edge pixel by using the Bresenham algorithm (Bresenham 1965). Results of the edge linking process are depicted in Fig. 7.

5 Object recognition

Since our application deals with a set of polyhedral objects, it is convenient to describe them in terms of their surfaces (regions). Thus, the recognition of the target objects is defined as the recognition of their visible surfaces. The main problem associated with this approach is deciding which of the features derived from the regions' geometrical characteristics should be employed as primitives for the recognition process. The criteria employed to select the optimal feature set has to take into consideration factors such as consistency, accuracy and computation complexity. In this regard, the local features such as junctions, lines and contour segments appear to be better suited when dealing with object occlusion. Unfortunately,

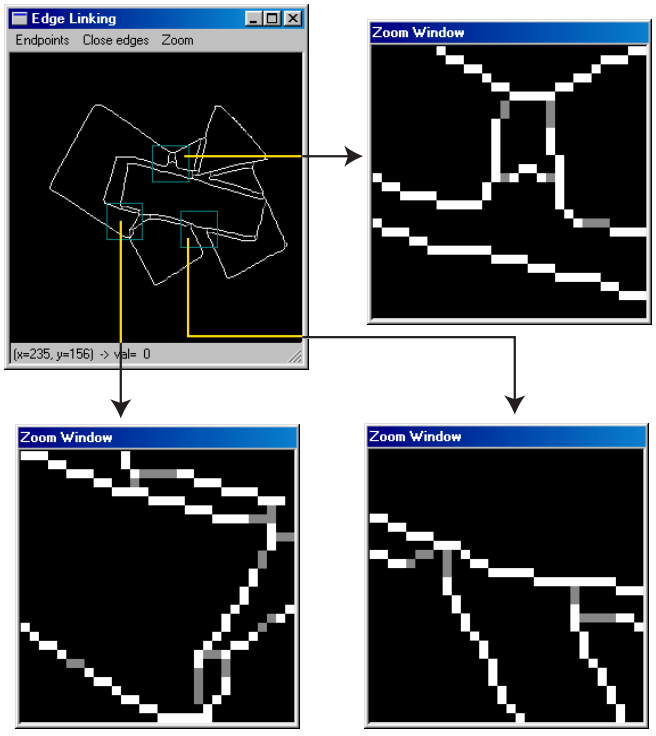


Fig. 7. The edge linking results when the algorithm is applied to the image illustrated in Fig. 6d (linking pixels are shaded)

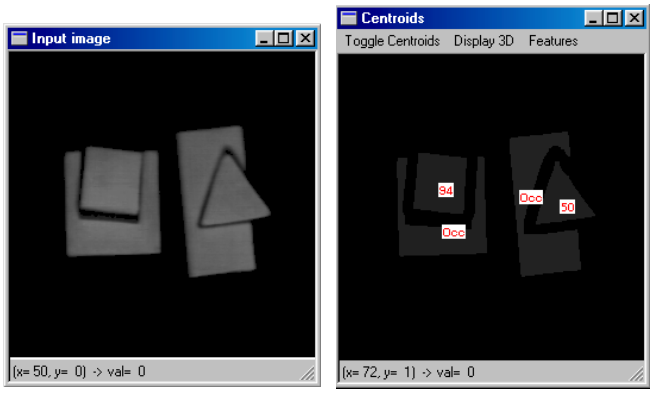
these features are viewpoint dependent, and in addition, they create a large number of hypotheses, a fact that requires a computationally intensive verification scheme. In contrast, global features (attributes) derived from the surfaces associated with the scene objects offer good viewpoint invariance, with the degree of ambiguity drastically reduced. In the present implementation, basic features such as area, perimeter, shape factor and the maximum distance from the region's centroid to the region's border are chosen.

The recognition algorithm has two main stages. The training stage consists of building the database by extracting the aforementioned features from each object of interest. Because the features that describe the objects have different ranges, it is necessary to apply a feature normalization scheme in order to avoid the situations where the features with the largest values overpower the remaining ones. The adopted feature normalization scheme initially subtracts the feature mean from each feature of the pattern, and then the result is divided with the feature variance (see Eqs. (4) and (5)). As a result, each feature of the pattern is standardized to zero mean and unit variance.

$$m_i = \frac{\sum_{j=1}^{k} x_j[i]}{k} \qquad s_i = \sqrt{\frac{\sum_{j=1}^{k}(x_j[i] - m_i)^2}{k}} \tag{4}$$

$$X_j[i] = \frac{x_j[i] - m_i}{s_i} \quad \text{for } j = 1, \ldots, k, \quad i = 1, \ldots, n \tag{5}$$

where n defines the number of features per pattern, m_i and s_i are the mean and the variance of the ith feature, x_j is the unprocessed jth pattern, k defines the number of patterns contained in the model database and X_j represents the normalized jth pattern.



(a) (b)

Fig. 8. Determining occluded objects: **a** the input image; and **b** the segmented image where the objects' elevations are highlighted. The occluded objects are marked *Occ*

The matching stage consists of computing the Euclidean distance between the input region and the regions contained in the database.

$$dist_j = \sqrt{\sum_{i=1}^{n}(X_j[i] - Y[i])^2}, \quad \text{for} \quad i = 1, \ldots, n \tag{6}$$

where X_j is the jth pattern from the model database and Y defines the pattern associated with the input region. The input region is contained in the database if the minimum distance that gives the best approximation is smaller than the threshold value φ.

$$\min(dist_j) \leq \varphi, \quad \text{for} \quad j = 1, \ldots, k \tag{7}$$

It is important to note that the features derived from the scene regions describe the object of interest globally. Consequently, they are consistent only if the scene objects are mildly occluded. To accomplish this requirement, it is necessary to determine the object situated on the top of the pile. This approach is very appropriate because the topmost object is rarely occluded and thus allows easy robotic manipulation.

In order to determine the object placed on the top of the pile, a framework that deals with a variable number of regions that fulfill the 3-D criteria is implemented. The aim of this framework is to identify the situations when the scene reveals obvious occlusions (Fig. 8).

The number of remaining regions is further decreased by applying other selection constraints. For this implementation, the area of the selected regions has to be bigger than a preset value that is 80% of the smallest region contained in the database. Next, from selected regions, the one that gives the best approximation with respect to the matching criteria belongs to the object situated on the top of the object pile. This process is illustrated in Fig. 9. If the matching criterion is not upheld (i.e. the minimum distance is greater than the preset value φ), the robot rearranges the scene in order to obtain a better configuration.

6 Pose estimation

Region matching can be extended to pose estimation as follows. For each model in the database, its appearance is sampled

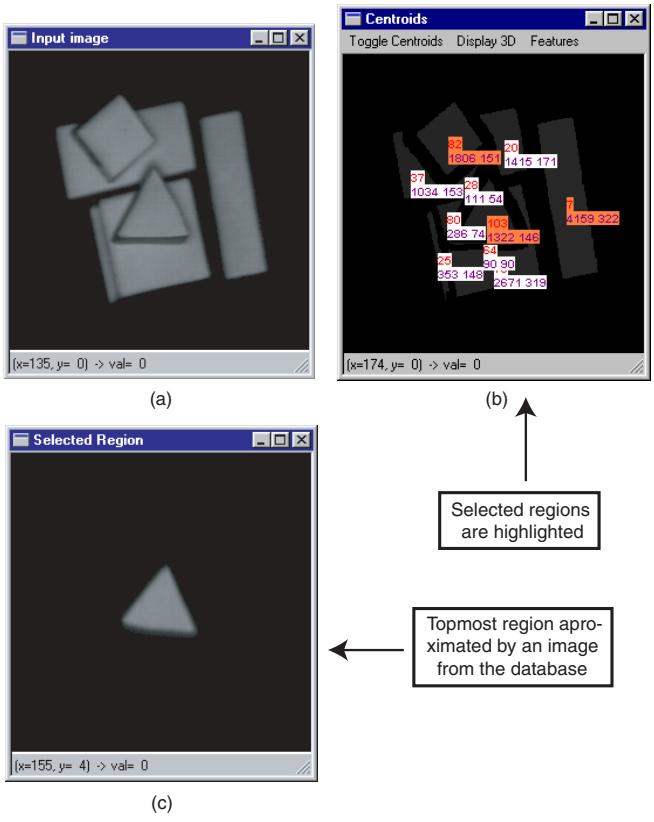


Fig. 9. The region-based recognition process: **a** the input image; **b** selecting the best-placed objects; and **c** the recognized objects considered to be on the top of the pile

over a range of viewing directions. The resulting images form an image set that encodes the attitude of the object in question. After recognition, the attitude of the object is determined by matching an image contained in the database. This form of region matching is not appropriate for two reasons. The first is associated with the dimension of the images contained in the image set. A typical image is represented by a two-dimensional 256×256 array of 8-bit intensity, and it would not be feasible to use this information directly as part of the pose estimation process. The second is related to the number of images used to sample the object's appearance. This represents a real problem when dealing with large databases, where the computational load associated with the matching process becomes impractical. Fortunately, the images contained in the image set can be compressed in order to speed up matching considerably. A popular technique for image compression is *principal component analysis* (PCA), also known as *eigenimage analysis* or *Karhunen–Loeve expansion*.

6.1 Image compression using PCA

PCA is a well-known technique for computing the direction of greatest variance for an image set. In this formulation, a low-dimensional, orthogonal subspace called the *eigenspace* (which describes the entire image set) is created by computing the eigenvectors of the covariance matrix of the image set. By projecting the image set on the eigenspace, the result is a collection of vectors that are the compressed representations of the image set (Turk and Pentland 1991). The operations mentioned above are briefly described in the next section.

6.2 Computing eigenspace

Let's consider P is the number of images contained in the image set. The image set matrix is obtained by converting each image into a row vector I_l of size N. The mean of all images contained in the image set is

$$\overline{I} = \sum_{l=1}^{P} I_l \tag{8}$$

In order to increase the variance between the images that form the image set, it is necessary to subtract the mean of the image set from each image.

$$\hat{I}_l = I_l - \overline{I}, \quad S = [\hat{I}_1, \hat{I}_2, ..., \hat{I}_P]^T \tag{9}$$

where S is the image set matrix and T defines the transpose matrix.

The next step involves computing the covariance matrix C of the image set (i.e. $C = S^T S$). The dimension of this matrix is $N \times N$, a fact that makes the calculation of its eigenvectors extremely difficult. If the number of images contained in the image set P is smaller than N, it is easier to calculate the eigenvectors of the reduced covariance matrix. The reduced covariance matrix is computed using $Q = SS^T$, but the dimension of the space is reduced to P. The eigenvectors of Q are computed by solving the eigenvector equation using the combination Householder Transform – QL algorithm (Press et al. 1992).

$$Qu_i = v_i u_i \tag{10}$$

where u_i is the ith eigenvector and v_i is the corresponding eigenvalue. The eigenspace is obtained by multiplying the matrix of eigenvectors U with the matrix S. The resulting matrix E defines the eigenspace and is $P \times N$ dimensional. If P is still too large, this space can be further reduced by using only the largest M eigenvalues. In this case the amount of compression is M/N and the dimension of space is M.

6.3 Database generation

The generation of the database is implemented by the two-stage training procedure illustrated in Fig. 10. The first stage deals with building and computing the object eigenspace as mentioned in the previous section. One of the problems associated with this approach is its sensitivity to the location of the object. To compensate for this problem, the objects are centered within the image. Another key problem consists of normalizing the image set by discarding the background (Murase and Nayar 1995). The second stage computes the database by projecting the normalized image set on the object's eigenspace.

$$h_l = [e_1, e_2, ..., e_P]^T(\hat{I}_l), \quad l = 1, ..., P \tag{11}$$

where $[e_1, e_2, ..., e_P]$ is the eigenspace matrix E, and h_l is a collection of vectors that define the PCA database.

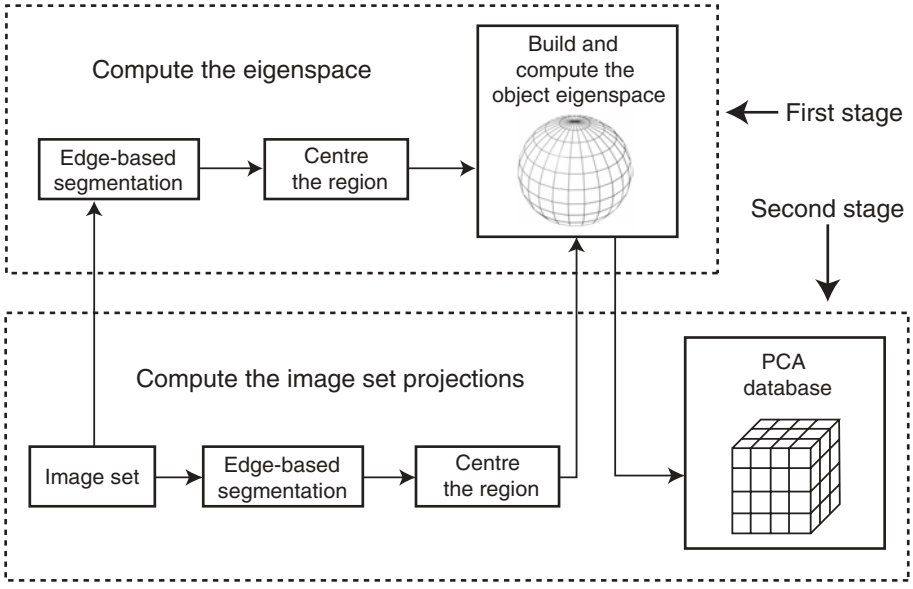


Fig. 10. The training procedure

6.4 *Pose matching*

To match the position of the recognized object, it is necessary to generate the input image for the pose estimation algorithm. In this way, the region resulting from the recognition stage is centered within the image and the mean of the corresponding image set is subtracted from it. Next, the resulting image is projected on the object's eigenspace as illustrated in Eq. (12).

$$h_{in} = [e_1, e_2, ..., e_P]^T(\hat{I}_{in}), \quad \hat{I}_{in} = I_{in} - \overline{I} \qquad (12)$$

where h_{in} is the projection on the eigenspace of the input image I_{in}.

The scene-to-model matching consists of evaluating the Euclidean distances between the projection associated with a scene object and those contained in the PCA database. The minimum distance defines the closest match. The image is contained in the database if the minimum distance is smaller than a predefined threshold value.

$$d_l = min \parallel h_{in} - h_l \parallel \leq \zeta, \quad l = 1, ..., P \qquad (13)$$

The threshold value ζ was chosen by analyzing the distribution of data in the PCA database using the procedure suggested by Nene and Nayar (1995).

6.5 *Pose sampling*

To sample the object pose using six *degrees of freedom* (DOF) with standard eigenimage analysis would require capturing all possible orientations for each object contained in the database. This approach is quite impractical since even a coarse rate of pose sample would require an extensive number of images. For instance, Edwards (1996) points out that sampling the object pose at a rate of 10 samples/DOF requires 10^6 images. Consequently, the pose estimation has been reformulated in order to reduce the size of the image set. In this way, eigenimage analysis is employed to constrain one rotational DOF, i.e. rotation about the z axis, while the remaining two rotational DOF are constrained by using the range data, i.e. computation of the normal vector to the surface in question (Ghita 2000). The translational components can be easily determined by analysing the coordinates of the centroid of the object's surface.

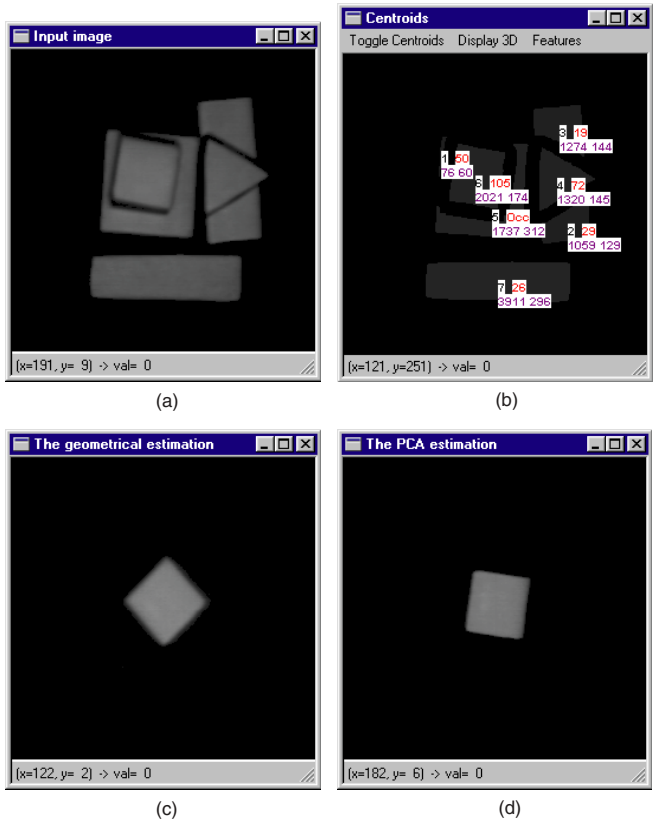


Fig. 11. Recognition results when the system is applied to a scene containing clutter: **a** input image; **b** the resulting image data (the first figure defines the rank of the region with respect to area, the second is the region's elevation and the last two represent the region's area and perimeter); **c** the topmost recognised region; and **d** the pose estimation for the topmost object

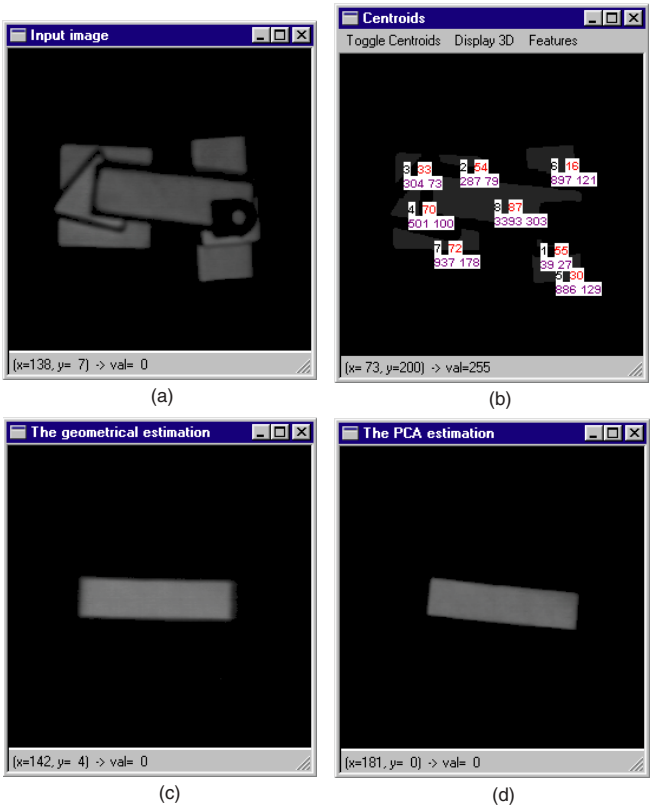


Fig. 12. Recognition results when the system is applied to a scene containing clutter and occlusion: **a** the input image; **b** the resulting image data; **c** the topmost recognized region; and **d** the pose estimation for the topmost object

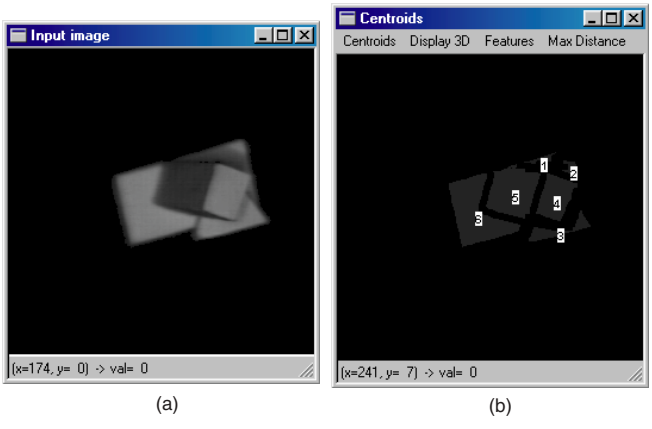


Fig. 13. A typical example that illustrates a case when the scene has to be rearranged in order to obtain a better configuration: **a** the input image; and **b** the resulting image data

7 Experiments and results

The DFD-based range sensor outlined in this paper generates 256×256 depth maps at a rate of 10 frames per second. The selection of an edge operator that maximizes the ratio quality in edge detection versus processing time was a key design decision. There is no doubt that achieving reasonable timing using a complex edge detector such as Canny is difficult, since the computational time required to extract edges from a $(256 \times 256)/256$ greyscale image is 4900 ms when running on a PC with a Pentium 133 processor. Fortunately, the ISEF-based GEF operator represents an attractive solution since the corresponding processing time is 545 ms, while the edge recovering performance is not significantly reduced. The proposed edge reconstruction and linking scheme is computationally efficient, the processing time required by these operations is about 500 ms (depending on the complexity of the input image). The computational overhead associated with the recognition and pose estimation algorithm is very low since all the computationally intensive operations are performed off-line.

The proposed bin-picking system was evaluated using scenes that contain clutter and occlusion. The cluttered scenes were created by arranging the objects contained in the database in various ways. The object recognition scheme described in Section 5 correctly identifies the topmost object if the perspective distortions or the occluding area affects less than 20% of the surface's total area. The object pose is determined using the strategy described in Section 6. Since the pose estimation process consists of two distinct stages, the system's performance was analysed for each stage separately. The first stage constrains the rotation about the z axis using eigenimage analysis. The rotation angle was sampled uniformly with the object lying flat on a dark worktable. For each object of interest, the object rotation was sampled by acquiring 24 training images. This generates a 24-dimensional eigenspace, and the resulting manifold was resampled to 720 points as described in (Murase and Nayar 1995). The pose was estimated with an error rate of 2.1% under the condition that the tilt of the object is limited to 25 degrees.

The remaining rotations are constrained by the normal vector. Since range data is used to compute the normal vector, the pose estimation error is directly related to the precision of the range sensor. In order to simplify the problems associated with the generation of accurate ground truth orientations, the object rotations about x and y axes were analysed independently. In our experiments, we obtained a maximum error rate of 7 degrees.

Some experimental results are depicted in Fig. 11 and Fig. 12. Figure 11 illustrates the case where the scene under investigation contains only clutter, while Fig. 12 illustrates the performance of the system when applied to a scene containing both clutter and occlusions. As mentioned earlier, there are situations when all the objects contained in the scene are heavily occluded or are positioned in such way that their appearance is significantly disturbed. In such situations, the robot rearranges the scene in order to obtain a better configuration (Fig. 13).

8 Conclusions

The proposed bin-picking system consists of four main components: range sensing, image segmentation, object recognition and pose estimation. The developed range sensor estimates the depth by measuring the relative blurring contained in a pair of images captured with different focal settings. To overcome the restriction associated with passive DFD, the current implementation is based on active DFD, a fact that offers the possibility of accurately estimating the depth even in cases when dealing with textureless scenes. Also, it is important to note that this range sensing technique can obtain

real-time depth estimation at very low cost. The second component of the system attempts decomposing the image into disjoint meaningful regions that have strong correlation with the objects that define the scene. A key issue for an edge-based segmentation technique relates to closing the gaps between unlinked edges and eliminating the spurious edges due to noise. In Sect. 4, we described an efficient morphological approach for edge reconstruction and linking. The particular novelty of this approach lies in the edge linking scheme that bridges the gaps in edges using only the local knowledge. As a consequence, it relaxes the demand of a priori information and assures an accurate and efficient search for edge paths in the image under investigation. The recognition process defines the third component and consists of analyzing the global geometrical primitives derived from the regions resulting after the application of the segmentation algorithm. Since this implementation addresses a bin picking application, crucial to this approach is the ability to locate the object placed on the top of the object pile in order to allow easy manipulation. In contrast with other related implementations (Dickinson et al. 1992; Fan et al. 1989), the current approach is particularly useful when dealing with small textureless objects, specifically when only a small number of faces are available. The last component determines the pose for the recognized object using an eigenimage approach augmented with a range data analysis. We believe that the current implementation can be successfully applied to industrial tasks such as sorting and packing. The experimental results demonstrate the validity of the proposed approach.

Acknowledgements. This work was funded in part by Motorola B.V. (Ireland) and the Research Institute for Networks and Communications Engineering (RINCE).

References

[Bergholm 1987] Bergholm F (1987) Edge focusing. IEEE Trans Pattern Anal Mach Intell 9(6):726–741

[Bresenham 1965] Bresenham J (1965) Algorithm for computer control of a digital plotter. IBM Syst J 4(1):25–30

[Canny 1986] Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698

[Casadei and Mitter 1996] Casadei S, Mitter S (1996) A hierarchical approach to high resolution edge contour reconstruction. In: Proceedings of the IEEE Conference for Computer Vision and Pattern Recognition (CVPR 96), San Francisco, USA, pp 149–154

[Dickinson et al. 1992] Dickinson S, Pentland A, Rosenfeld A (1992) 3-D shape recovery using distributed aspect matching. IEEE Trans on Pattern Anal and Mach Intell 14(2):174–198

[Edwards 1996] Edwards J (1996) An active, appearance-based approach to the pose estimation of complex objects. In: Proceedings of the IEEE Intelligent Robots and Systems Conference, Osaka, Japan, vol. 3, pp 1458–1465

[Eichel and Delp 1985] Eichel P, Delp E (1985) Sequential edge detection in correlated random fields. In: Proceedings of the IEEE Conference for Computer Vision and Pattern Recognition (CVPR 85), pp 14–21

[Fan et al. 1989] Fan T, Medioni G, Nevatia A (1989) Recognizing 3-D objects using surface description. IEEE Trans Pattern Anal Mach Intell 11(11):1140–1157

[Ghita 2000] Ghita O (2000) A real-time low-cost vision sensor for robotic bin picking. PhD thesis, Dublin City University, Dublin

[Ghita and Whelan 2001] Ghita O, Whelan P (2001) A video-rate range sensor based on depth from defocus. Optics Laser Technol 33(3):167–176

[Gupta et al. 1993] Gupta A, Chaudhury S, Parthasarathy G (1993) A new approach for aggregating edge points into edge segments. Pattern Recogn 26(7):1069–1086

[Hoover et al. 1996] Hoover A, Jean-Baptiste G, Jiang X, Flynn P, Bunke H, Goldgof D, Bowyer K, Eggert D, Fitzgibbon A, Fisher R (1996) An experimental comparison of range image segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 18(7):673–689

[Hancock and Kittler 1990] Hancock E, Kittler J (1990) Edge-labelling using dictionary-based relaxation. IEEE Trans Pattern Anal Mach Intell 12(2):165–181

[Henderson 1983] Henderson C (1983) Efficient 3-D object representation for industrial vision systems. IEEE Trans Pattern Anal Mach Intell 5(6):609–617

[Jiang and Bunke 1999] Jiang X, Bunke H (1999) Edge detection in range images based on scan line approximation. Comput Vis Image Understanding 73(2):183–199

[Johnson and Hebert 1999] Johnson A, Hebert M (1999) Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans Pattern Anal Mach Intell 21(5):433–449

[Kriegman and Ponce 1990] Kriegman D, Ponce J (1990) On recognizing and positioning curved 3D objects from image contours. IEEE Trans Pattern Anal Mach Intell 12(12):1127–1137

[Lindeberg 1993] Lindeberg T (1993) On scale selection for differential operators. In: Proceedings of 8th Scandinavian Conference on Image Analysis, Tromso, Norway, pp 857–866

[Lowe 1987] Lowe D (1987) The viewpoint consistency constraint. Int J Comput Vis 1(1):57–72

[Marr and Hildreth 1980] Marr D, Hildreth E (1980) Theory of edge detection. In: Proceedings of Royal Society B 207, London, UK, pp 187–217

[Murase and Nayar 1995] Murase H, Nayar S (1995) Visual learning and recognition of 3-D objects from appearance. Int J Comput Vis 14:5–24

[Nayar et al. 1995] Nayar S, Watanabe M, Noguchi M (1995) Real-time focus range sensor. In: Proceedings of International Conference on Computer Vision (ICCV 95), pp 995–1001

[Nene and Nayar 1995] Nene S, Nayar S (1995) A simple algorithm for nearest neighbour search in high dimension. Technical report, Columbia University

[Ohba and Ikeuchi 1997] Ohba K, Ikeuchi K (1997) Detectability, uniqueness and reliability of eigen windows for stable verification of partially occluded objects. IEEE Trans Pattern Anal Mach Intell 19(9):1043–1048

[Pentland 1987] Pentland A (1987) A new sense for depth of field. IEEE Trans Pattern Anal Mach Intell 9(4):523–531

[Pentland et al. 1994] Pentland A, Scherock S, Darrell T, Girod B (1994) Simple range cameras based on focal error. J Optical Soc America 11(11):2925–2935

[Ponce et al. 1989] Ponce J, Chelberg D, Mann W (1989) Invariant properties of straight homogenous generalized cylin-

ders and their contours. IEEE Trans Pattern Anal Mach Intell 11(9):951–966

[Press et al. 1992] Press W, Teukolski S, Vetterling W, Flannery B (1992) Numerical recipes in C. Cambridge University Press, pp 456–481

[Saber et al. 1997] Saber E, Tekalp A, Bozdagi G (1997) Fusion of color and edge information for improved segmentation and edge linking. Image Vis Comput 15(10):769–780

[Shen and Castan 1992] Shen J, Castan S (1992) An optimal linear operator for step edge detection. Comput Vis Graph Image Process: Graph Models Image Process 54(2):112–133

[Subbarao 1989] Subbarao M (1989) Efficient depth recovery through inverse optics. In: Freeman H (ed) Machine vision for inspection and measurement. Boston, MA: Academic Press, pp 101–126

[Subbarao and Surya 1994] Subbarao M, Surya G (1994) Depth from defocus: a spatial domain approach. Int J Comput Vis 13:271–294

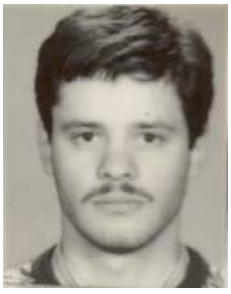
[Turk and Pentland 1991] Turk M, Pentland A (1991) Eigenfaces for recognition. J Cognitive Neurosci 3(1):71–86

[Vincent 1993] Vincent L (1993) Morphological greyscale reconstruction in image analysis: applications and efficient algorithms. IEEE Trans Pattern Anal Mach Intell 2(2):176–201

[Vincken et al. 1996] Vincken K, Niessen W, Viergever M (1996) Blurring strategies for image segmentation using multiscale linking model. In: Proceedings of the IEEE Conference for Computer Vision and Pattern Recognition (CVPR 96), San Francisco, USA, pp 21–26

[Zerroug and Nevatia 1996] Zerroug M, Nevatia R (1996) 3-D description based on the analysis of the invariant and quasi-invariant properties of some curved-axis generalized cylinders. IEEE Trans Pattern Anal Mach Intell 18(3):237–253

[Zisserman et al. 1994] Zisserman A, Forsyth D, Mundy J, Rothwell C, Liu J, Pillow N (1994) 3D object recognition using invariance. Technical report, University of Oxford, Robotics Research Group



Ovidiu Ghita received his B.E. and M.E. degrees in Electrical Engineering from Transilvania University, Brasov, Romania. From 1994 through 1996 he was an Assistant Lecturer in the Department of Electrical Engineering at Transilvania University. Since then, he has been a member of the Vision Systems Group at Dublin City University (DCU), during which time he received his Ph.D. for work in the area of robotic vision. Currently, he holds a position of Postdoctoral Research Assistant in the Vision Systems Laboratory at DCU. His current research interests are in the area of range acquisition, shape representation, object recognition and machine learning.

Paul F Whelan received his B.Eng.(Hons) degree from the National Institute for Higher Education Dublin, a M.Eng. degree from the University of Limerick, and his Ph.D. from the University of Wales, Cardiff. During the period 1985–1990 he was employed by Industrial and Scientific Imaging Ltd and later Westinghouse (WESL), where he was involved in the research and development of industrial vision systems. He was appointed to the School of Electronic Engineering, Dublin City University (DCU), in 1990 and currently holds the position of Associate Professor and Director of the Vision Systems Laboratory. As well as a wide range of scientific publications, Prof. Whelan co-edited *Selected Papers on Industrial Machine Vision Systems* (1994), and was the co-author of *Intelligent Vision Systems for Industry* (1997) and *Machine Vision Algorithms in Java* (2000). His research interests include applied morphology, texture analysis, machine vision and medical imaging. He is a senior member of the IEEE, a chartered engineer and a member of the IEE, SPIE and IAPR. He is also a member of a number of machine vision related conference program committees. He currently serves on the IEE Irish centre committee, as a member of the governing board of the International Association for Pattern Recognition (IAPR) and as the president of the Irish Pattern Recognition and Classification Society.