MULTI-VIEW TECHNIQUE FOR 3D POLYHEDRAL OBJECT RECOGNITION USING SURFACE REPRESENTATION

Maria de Fátima Santos Farias[1]
João Marques de Carvalho[2]

[1]Universidade Federal do Maranhão - CT - DEE
Campus do Bacanga, São Luís, MA, Brazil
(farias@dee.ufma.br)
[2]Universidade Federal da Paraíba - CCT - DEE
Caixa Postal 10.105, 58109-970 Campina Grande, PB, Brazil
(carvalho@dee.ufpb.br)

Abstract: This work proposes a 3D object recognition method based on the matching of geometric attributes extracted from polyhedral models. 3D attributes are determined by crossing data obtained from three 2D views of the object. The recognition method matches global, local and relational attributes, in order to make possible the identification and positioning of both, the models present in the scene and the visible surfaces in the views.

1 INTRODUCTION

Technological development of modern society has led to increasing research on methods and systems to automate human tasks. “Machines with vision capability", has been a recurrent research theme for the last three decades. The scope of application of such machines is very large, ranging from military weapons and space probes to industrial systems and medical devices.

Many applications of computer vision systems have been found in industrial automation, where the need for recognizing 3D objects, or parts of objects, is frequently present. Such is the case of industrial inspection systems, used for quality control in production lines. In general, whenever only classification is required, it is possible to perform 3D recognition by analysis of 2D images of the object, taken from distinct view points by two or more cameras. This approach has the advantage, compared to a direct analysis of 3D contours, of utilizing the great number of existing 2D image processing algorithms, besides considerably reducing the amount of computation required. This technique is known as stereovision or multi-view (Marshal and Martin, 1992; Chiou *et alii*, 1992).

The greatest problem in stereovision resides in performing a reliable and fast matching between corresponding points present in the scene images captured from different viewpoints. This correspondence is determined by the geometry of the image acquisition set-up, which has to be known in advance.

Model-based vision systems represent the most used strategy for object or shape recognition. In this approach, features extracted from the objects in a scene are matched against features of previously stored object models. For 3D recognition, a three-dimensional formulation of these features is required. Comprehensive studies of techniques and methods for model-based vision systems have been presented by Chin and Dyer (1986), Besl and Jain (1985), Arman and Aggarwal (1993) and Requicha (1980). More recently, Borges (1996) developed a solution to the problem of 3D recognition of complex curved objects with articulations.

This paper presents a new model-based recognition method for scenes composed of three-dimensional objects, possibly with partial occlusion. The main guideline of this work has been to keep computational complexity low, in order to increase speed of operation, while still achieving good recognition rates. For that, the method relies on the matching of a few simple geometric attributes extracted from the scene with known geometric attributes of a set of polyhedral object models. Operation can be viewed in three stages: 1) construction of a library of object models; 2) extraction of attributes of scene objects; and 3) matching of scene-model attributes (recognition and location).

A block diagram for this operation is shown in Figure 1. The library consists of three-dimensional models plus a corresponding set of model surface geometric attributes. Three 2D images of scene objects are captured during the attribute extraction stage. After some processing operations are performed on the images, 2D features are obtained. From these features a set of 3D geometric attributes for all visible surfaces of the objects is built.

The “Object Identification” block in Figure 1 searches the library, looking for models with attributes that match those extracted from the scene. Besides selecting the matching models, the system also defines which model surfaces correspond to the visible surfaces in the scene. Finally, model and scene information are used to determine the spatial location of the identified model in the scene frame of reference.

Section 2 of this paper presents the object modeling technique utilized in the work. Image acquisition and feature extraction are discussed in Section 3, while Section 4 presents the proposed method for object recognition. Section 5 analyzes test results, for which both synthetic and real scene images were used. Conclu-

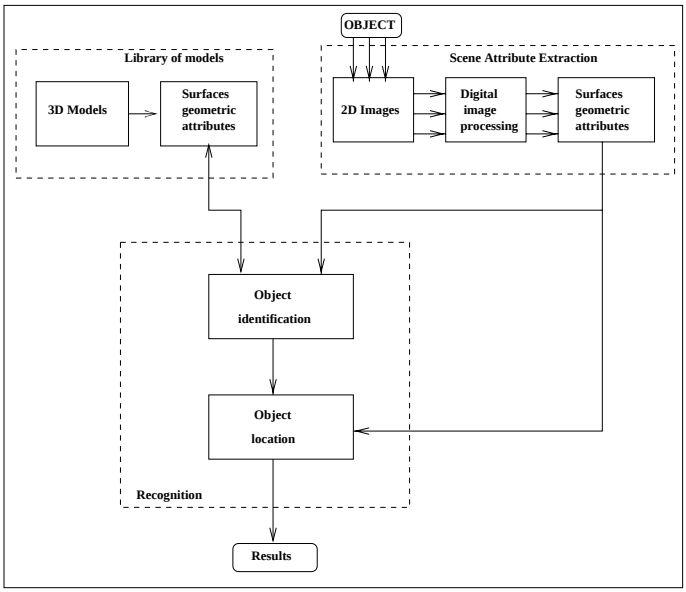

Figure 1: Block Diagram of Recognition System

sions are taken and discussed in Section 5.

2 3D OBJECT MODELING

The models composing the library were constructed using a commercial CAD system (AutoCAD R12), instead of built from real images of the objects they represent. This choice of modeling technique is due to the the facilities offered by those systems, which considerably ease the task (especially that of feature extraction) and make it very straightforward to expand and update the library.

Each model in the library is composed by two basic parts: 1) *a 3D shape* and 2) *the 3D geometric attributes*. The exact methodology used to elaborate these models has been presented by the authors in a previous work (Farias and Carvalho, 1997). For the tests reported here ten polyhedral models, representing shapes commonly found in industrial environments, were developed. They are assumed to be in a stable position on a table, whose lower left-hand corner represents the origin of the 3D coordinate system. Figure 2 shows 3D views of the complete set of models.

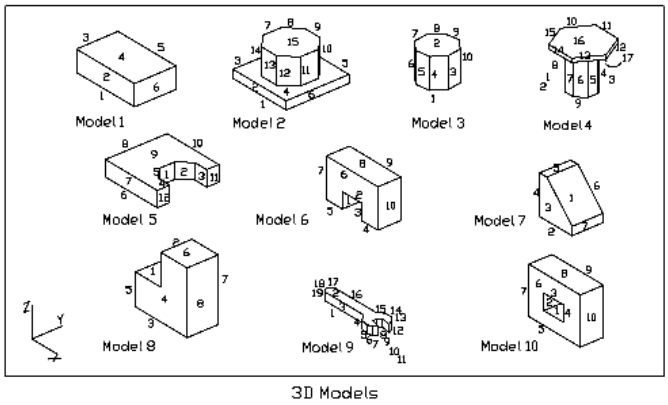

Figure 2: 3D View of Models

The 3D attributes extracted are related to the models' surfaces: *area, normal vector, perimeter, number of edges* and *vertices*. From these, some other attributes related to the solid can be calculated: *minimum and maximum area* and *number of edges* for all surfaces.

3 IMAGE ACQUISITION AND DATA EXTRACTION

A methodology for image acquisition and 3D data extraction for use in trinocular stereo vision systems is presented in this section. A possible object/camera set-up for such a system is ilustrated in Figure 3. The object is assumed to be in a stable position on top of a flat table. Three images from different viewpoints are taken, so that data corresponding to the width, height and depth of the object can be obtained. The lower left corner of the table is taken as the origin of a 3D coordinate system, the $X-Y$ plane corresponding to the table top. The cameras should be placed on orthogonal planes, at the same distance from the table center and should have the same focal length.

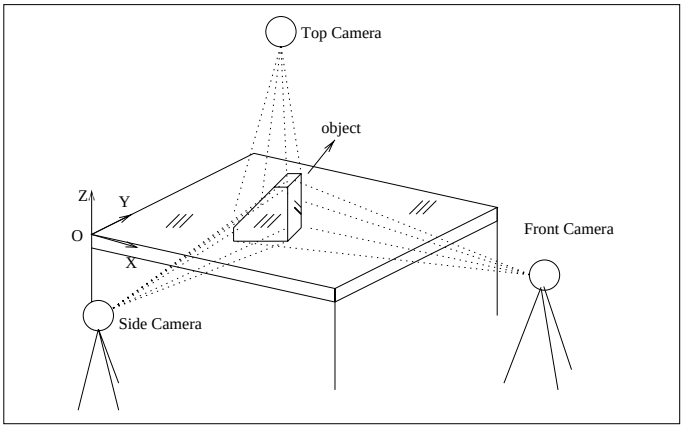

Figure 3: Image Acquisition Set-up

As shown in Figure 3, one camera views the scene from a plane parallel to the table top ($X-Y$ plane), originating the top view image of the scene. Another camera views the scene from a plane parallel to the $X-Z$ plane, producing the side view of the objects on the table. Finally, the third camera is placed on a plane parallel to the $Y-Z$, generating a front view image.

Figure 4 shows a block diagram for the proposed methodology. The three camera images are digitized and sent to the preprocessing and segmentation stage. Vertex extraction from the 2D views of the scene surfaces is performed at this stage. The collected data is used to create a 2D database, from which the regions forming each view can be analyzed. A data crossing technique is applied on the coordinates of the extracted 2D surface vertices in order to identify the 3D vertices characterizing the scene objects in the three-dimensional space. The set of 3D data thus obtained allows calculation of surface attributes for the recognition process. The surfaces which can not be described in 3D are discarded. A final analysis is needed to detect and eliminate redundancy, caused by surfaces which are visible in more than one view. A report is then produced by the system, listing the vertices and the attributes of the visible surfaces.

In order to have meaningful image processing and analysis, some aspects concerning image acquisition and camera calibration must be considered, as examined next (Marshall and Martin, 1992).

3.1 Camera Geometrical Transformations

Camera geometrical transformations are procedures used to determine real physical parameters from object images. These operations involve the characteristics of the camera utilized for

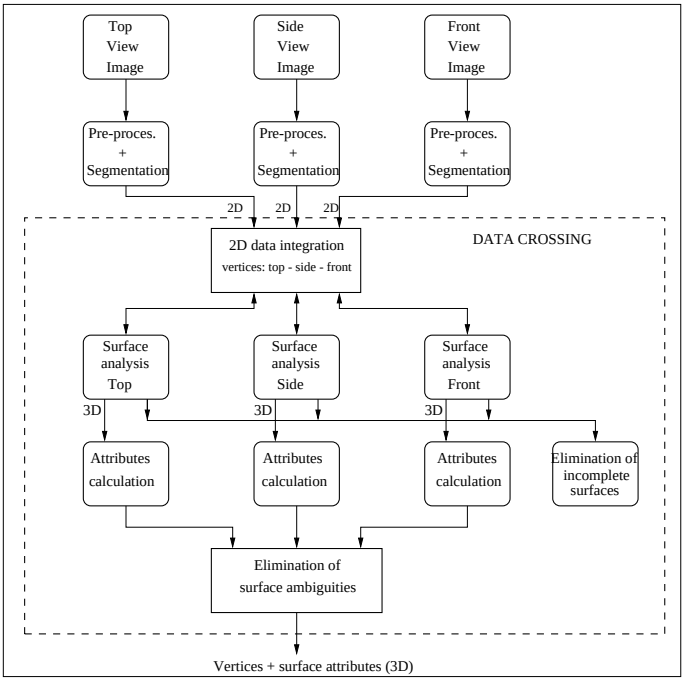

Figure 4: Image Processing and Data Extraction

image acquisition and digitizing. Their choice depends on the specific application intended for the vision system (Marshal and Martin, 1992; Liu and Tsai, 1990; Chen and Kak, 1989; Dhond and Aggarwal, 1989; Chiou *et alii*, 1992).

In general, camera characteristics concern internal parameters, like focal length and lens distortion. The relationship between real object and image coordinates is considered an external parameter. Digitizing consists in sampling image light intensity (or brightness) over a uniform retangular grid.

The regular procedure for digitizing pictures taken by ordinary photographic cameras is to use a scanner. The relationship between real lenght in the scene and the lenght correponding to a certain number of pixels in the digitized image, is given by:

$$X_r = \left(\frac{d-f}{fAt}\right) np \qquad (1)$$

with parameters: X_r - real object length; d - camera/object distance; f - camera focal length; A - negative/positive amplification factor; t - scanner sampling rate; and np - number of pixels.

Therefore, once the number of pixels in the processed image is known, the real object length can be obtained. The geometrical transformation is the factor $T_g = \left(\frac{d-f}{fAt}\right)$.

3.2 Image Pre-processing and Segmentation

Image processing and segmentation techniques are applied to the images representing the three views of the scene. The objective is to reduce each of those images to line drawings, where only the contours of the surfaces are visible. These contours should be continuous and well defined, in order to guarantee good surface extraction during the segmentation stage. As a result of segmentation, a list of 2D vertices is obtained, defining the visible surfaces in each view image.

The pre-processing and segmentation algorithms used in this work were developed by Melcher for VLSI implementation of real-time contour and vertex extraction in sequences of images (Melcher *et alii*, 1996; Melcher *et alii*, 1996a; Melcher *et alii*, 1997). These methods are briefly reviewed now.

The contour extraction algorithm, consists of three stages: 1) gradient image calculation; 2) calculation of local maximum and extraction of first contour; and 3) extraction of second and final contour.

A 2D spline gradient function, defined by four 5×5 masks, is used to generate the gradient images (Melcher *et alii*, 1996). The masks parameters correspond to the coefficients of the derivative of the cubic spline function which interpolates the central pixel in a 5×5 window along the four possible directions: horizontal, vertical and the two diagonals. The values output by the four masks are averaged to produce the approximation to the image gradient at the central pixel coordinates. This averaging of the gradient along four different directions, makes the algorithm very robust against the presence of random noise in the image. Spline functions produce the smoothest interpolation in a mean-square sense, with null error at the original sample points, being therefore extremely adequate to this type of application (de Carvalho and Hanson, 1984).

The surfaces contours are extracted by sweeping the gradient images with windows. Initially a 5×5 window is applied, for which the relative weight of the central pixel gray level with respect to its neighbours is calculated. Only neighbours with gray level smaller than or equal to the central pixel are considered in this calculation. If the calculated value is higher than a given threshold, the central pixel is hypothesized as belonging to a contour. The contour thus obtained is further verified by comparison against a set of 3×3 masks, representing all possible contour paths passing by the central pixel of a 3×3 window. Contours extracted at this stage may be more than one pixel wide.

Further refinement is attained in a second stage of contour extraction, which produces contour lines with one pixel width. For each image window being tested, a value is calculated for the central pixel, to determine whether it should be eliminated or preserved as a countour point. This value corresponds to an weighted sum of gray levels, for the window contour pixels defined in the previous stage (Melcher *et alii*, 1997).

The segmentation algorithm performs detection of the vertices defining the surfaces (polygonal regions) in the 2D images (Melcher *et alii*, 1996a). This method is based on the work by Cortez and de Carvalho (1995) and Villar (1997), testing local curvature for the contour pixels extracted in the two previous stages. A set of predefined vertex patterns for the central pixel of a 5×5 window provides precalculated comparative curvature values. This procedure allows for significative reduction of on-line computation.

This set of methods has been successfully applied to the images in this work. Small weight adjustments were needed to sharpen the edges of internal surfaces. Some extra patterns also had to be generated to account for the case of internal vertices defined by three or more edges.

3.3 Data Crossing for 3D Vertex Determination

The first step towards 3D vertex determination is to build a data base with the 2D data extracted from the three scene views. For that, consider the following definition:

Definition 3.1 *Let k_s, k_l and k_f be the number of surfaces present at the top, side, and front views of the scene, respectivelly. Further, let W_k, W_m, W_n be the total number of vertices belonging to the k-th surface of the top view, the m-th surface of the side view, and the n-th surface of the front view, respectively.*

For each view the vertices are given by;

$$V_{kp} = (x_{kp}, y_{kp}) \quad (2)$$
$$V_{mq} = (x_{mq}, z_{mq}) \quad (3)$$
$$V_{nr} = (y_{nr}, z_{nr}) \quad (4)$$

where
V_{kp} = p-th vertex of the top view k-th surface; V_{mq} = q-th vertex of the side view m-th surface; V_{nr} = r-th vertex of the front view n-th surface; and

$$1 < p \leq W_k, \quad 1 < q \leq W_m, \quad 1 < r \leq W_n$$
$$1 < k \leq k_s, \quad 1 < m \leq k_l, \quad 1 < n \leq k_f$$

As an example, consider the situation where a vertex is visible in the three views, as ilustrated in Figure 5. In the top view image one has vertex V_{kp} given by coordinates (x_{kp}, y_{kp}) e wants to find the z_kp coordinate of that vertex at the two remaining views, V_{mq} e V_{nr}. For that, a search is done on the side and front views, attempting to match x_{kp} with x_{mq} and y_{kp} with y_{kr}. The objective is to find a vertex which is visible at the top and at one of the other, side or front, views. The match between coordinates is said to occur when the following criteria are satisfied, for the side and front views, respectivelly:

$$|x_{kp} - x_{mq}| \leq Error \quad (5)$$
$$|y_{kp} - y_{nr}| \leq Error \quad (6)$$

For each match thus determined, the corresponding z value, z_{mq} and/or z_{nr} is stored. Whenever more than one z is found, the largest value is selected, since the top view registers the highest points of the object. Therefore,

$$z_{kp} = Largest < z_{mq}, z_{nr} >$$

If the search of the side view fails, there is still the possibility of a sucessfully search in the front view. In case both searches fail, the corresponding surface is declared incomplete and discarded.

When analysing the side view, only the the top view is searched. Vertex V_mq with coordinates $(x_{mq}, \ z_{mq})$ is known and one wants to determine the y_{mq} coordinate. By searching the top view a match is attempted between x_{mq} and x_{kp}, according to the criterion:

$$|x_{mq} - x_{kp}| \leq Error \quad (7)$$

For each match detected the corresponding value of y_kp is stored. If more than one correspondence is found the smallest value is selected, corresponding to the point which is nearest to the camera and therefore with highest probability of being seen. Thus,

$$y_{mq} = Smallest < y_{kp} > .$$

As before, those surfaces with incomplete vertices are discarded.

For the front view analysis, again only the top view is searched. In this case, one knows vertex V_{nr} with coordinates (y_{nr}, z_{nr}) and wishes to determine x_{nr}. A search of the top view is performed, looking for matches between y_{nr} and y_{kp}, in order to satisfy the criterion:

$$|y_{nr} - y_{mp}| \leq Error \quad (8)$$

For each match the corresponding value of x_{nr} is stored. If more than one correspondence is established the largest value of x_{nr} is selected, yielding:

$$x_{nr} = Largest < x_{kp} >$$

As before, surfaces with incomplete vertices are rejected.

The procedures described above are repeated until all vertices of each visible surface have been examined.

An empirical error thresholding, *Error*, has been determined for use in equations 5,6, 7 e 8. This error value, measured in length units, allows for a safety margin against errors and deviations that may occur during image alignment and vertex extraction. It is based on the smallest distance between two edges on an object or between two objects.

Once the visible surfaces in the scene are completely specified by the 3D vertices, their geometric attributes can be calculated to be used in the recognition process. The surface attributes used in the present work are: *area, perimeter, normal vector, number of vertices and centroid*, all directly calculated from the vertices extracted in the previous phase.

In multi-view vision systems redundancy may occur due to surfaces which are visible by more than one camera. In this case identification and elimination of duplicate surfaces is required, in order to avoid ambiguities. In this work, elimination of duplicate surfaces has been achieved through the matching of attributes among surfaces of different views. A priority is established, with the top view being analyzed first, followed by the side and the front views, in that order. Those surfaces of the side view which have already been described in the top view are eliminated. The same occurs with surfaces of the front view which have been described either by the top or the side view. The top view, therefore, has the highest priority and its surfaces are never eliminated.

Lemma 3.1 proposes error criteria to avoid duplicity of surfaces, according with the geometric model and scene attributes given by definitions 3.2 and 3.3.

Definition 3.2 *Let $\mathcal{M}$ be a set of models, such that:*

$$\mathcal{M} = \{M_1, M_2, \cdots, M_m\}.$$

The attributes of model $M_j \in \mathcal{M}$, are:

- *Global attributes: number of surfaces (NF_j), minimum ($AMIN_j$) and maximum ($AMAX_j$) surface area, minimum ($NLMIN_j$) and maximum ($NLMAX_j$) number of edges by surface.*

- *Surface S attributes: area (A_{jS}), perimeter (P_{jS}), normal vector ($\vec{N}_{jS}$), centroid (C_{jS}), number of edges (NL_{jS}) and vertex set (V_{jS}).*

Definition 3.3 *The surface attributes for a scene with nf visible surfaces are:*

- *surface s: area (a_s), perimeter (p_s), normal vector ($\vec{n}_s$), centroid ($\vec{c}_s$), number of edges (nl_s) and the set containing all vertices (VC_s).*

Lemma 3.1 *Consider that the k-th surface of a view is to be analyzed for detection of duplicity. Let t be a surface already described by a higher priority view. The analyzed k-th surface is eliminated if the following conditions are simultaneously satisfied:*

$$|a_k - a_t| \leq 0.20 a_k \tag{9}$$
$$|p_k - p_t| \leq 0.20 p_k \tag{10}$$
$$|nl_k - nl_t| \leq 0.20 nl_k \tag{11}$$
$$|\vec{n}_k - \vec{n}_t| \leq 0.20 \vec{n}_k \tag{12}$$
$$|\vec{c}_k - \vec{c}_t| \leq 0.20 \vec{c}_k \tag{13}$$

The threshold factor 0.20 allows for a 20% relative attribute difference between the surfaces, before detecting duplicity. This factor has been empirically determined.

4 OBJECT IDENTIFICATION AND POSITIONING

As in most model-based vision systems, recognition or identification of the objects take place in two steps. Initially the matching of surface attributes is used to generate hypotheses about the presence of models in the scene. In a second step the spatial relationships among matched surfaces are used to verify hypotheses, eliminating the false ones and identifying the objects.

4.1 Object Identification

Lemma 4.1 *Model k is examined for hypothesis generation if the area a_i and the number of edges nl_i of the i-th scene surface are within the range of variation of these parameters for that model, i.e,*

$$AMIN_k \leq a_i \leq AMAX_k \tag{14}$$
$$NLMIN_k \leq nl_i \leq NLMAX_k \tag{15}$$

The purpose of this lemma is to avoid those models which are clearly distinct from the objects in the scene, therefore reducing search time in the library. Whenever the lemma is satisfied, every surface of model k is tested against the scene. The I-th surface of the k-th model is selected for hypothesis generation if the following matches are simultaneously verified:

1. between the areas of the i-th scene surface (a_i) and the I-th surface of the k-th model ($A_k I$), where $1 \leq i \leq nf$ and $1 \leq I \leq NF_k$:

$$ea = \frac{|a_i - A_{kI}|}{A_{kI}} \leq 0.20; \tag{16}$$

2. between the perimeters of the i-th scene surface (p_i) and the I-th surface of the k-th model ($P_k I$):

$$ep = \frac{|p_i - P_{kI}|}{P_{kI}} \leq 0.20; \tag{17}$$

3. between the number of edges of the i-th scene surface (nl_i) and that of the I-th surface of the k-th model ($NL_k I$):

$$enl = \frac{|nl_i - NL_{kI}|}{NL_{kI}} \leq 0.20; \tag{18}$$

4. between the length of one edge of the i-th scene surface ($length_i$) and the length of any edge of the I-th surface of the k-th model ($LENGTH_{kI}$):

$$el = \frac{|length_i - LENGTH_{kI}|}{LENGTH_{kI}} \leq 0.20. \tag{19}$$

The above restrictions, plus those of equations 14 and 15, generate all possible identification hypotheses. Equations 16-19 mean that a maximum difference of 20%, relative to the model, is allowed between scene and model attributes in order to have a match detected. This error threshold has been empirically determined. The search space resulting from the application of equations 14 and 15 is significantly reduced, compared to the total library space, and very reliable, in the sense that it has always produced the correct hypotheses, for the tests performed.

At the end of this stage a set of hypotheses is produced for each scene surface i which found a match among the searched models. This set is:

$$V_i = \{(k_q; I_{k_{qS_1}}, I_{k_{qS_2}}, ...)\}, q = 1, 2, ... \tag{20}$$

where k_q represents the model and I_{k_q} the respective matched surfaces.

The hypotheses generation process, just described, does not consider any relationship among the surfaces being tested. Only local and global surface attributes are taken into account to detect possible matches. The next phase of operation is hypotheses verification, which is based on the matching of relational attributes. Those are the attributes that define the position of surfaces in the object/model, as well as the spatial relationship between them. The objective is to verify if the spatial relations among surfaces in the scene are the same as those among the respective matched surfaces in the model.

The most appropriate attributes for expressing the spatial relationship of the surfaces of a solid are the *normal vector* or *normal* (Grimson and L.-Pérez, 1984; Murray, 1987; Klemt and Infantosi, 1995) and the *centroid* (Oshima and Shirai, 1983), associated to each surface. Since different coordinate systems are used for the scene and the models, the angle between the normal vectors and the distance between centroids have been used as relational attributes to the objects surfaces. The angle between two normal vectors represents the solid angle between the two corresponding surfaces and the distance between two centroids represents the average distance between the surfaces.

Hypotheses verification happens in two steps. Initially the surfaces of the hypothesized models (those in the set V_i) are grouped by pairs, such that the angle between normal vectors and the distance between centroids for each pair thus created match the same parameters for some pair in the scene. All possible combinations of distinct surface pairs are considered in this

analysis, to account for the possibility of more than one object being present in the scene. The error (matching) criteria for selecting pairs of scene surfaces are defined in lemmas 4.2 and 4.3.

Lemma 4.2 *Let θ_i be the angle between the normal vectors ($\vec{n}_i, \vec{n}_{i+1}$) for a pair of scene surfaces and θ_I be the angle between the normal vectors ($\vec{N}_{kI}, \vec{N}_{kI+1}$) for a pair of model surfaces. Let e_θ be the angular error given by the absolute difference between the scalar products (cosines) of these angles. The matching error criteria to be satisfied is:*

$$e_\theta = |\cos\theta_i - \cos\theta_I| \leq 0.28 \tag{21}$$

where:

$$\cos\theta_i = \vec{n}_i \cdot \vec{n}_{i+1}$$
$$\cos\theta_I = \vec{N}_{kI} \cdot \vec{N}_{kI+1}$$

Lemma 4.3 *Let $dist_i$ and $dist_I$ be the distances between the centroids[1] of two scene surfaces (c_i, c_{i+1}) and two model surfaces (C_{kI}, C_{kI+1}), respectively. Let e_d be the absolute distance error relative to $dist_I$. The matching error criteria is given by:*

$$e_d = \frac{|dist_i - dist_I|}{dist_I} \leq 0.10 \tag{22}$$

where:

$$dist_i = \sqrt{(\bar{x}_i - \bar{x}_{i+1})^2 + (\bar{y}_i - \bar{y}_{i+1})^2 + (\bar{z}_i - \bar{z}_{i+1})^2}$$
$$dist_I = \sqrt{(\bar{x}_I - \bar{x}_{I+1})^2 + (\bar{y}_I - \bar{y}_{I+1})^2 + (\bar{z}_I - \bar{z}_{I+1})^2}$$

The error threshold value (0.28) in equation 21 was empirically determined, corresponding to an approximate angular tolerance of 15° near $\frac{\pi}{2}$. The 0.10 error threshold in equation 22, also empirically found, was set lower than the threshold value used to generate hypotheses, in order to allow for a further refinement.

Once the correspondence between scene/model surface pairs has been established, the next objective is to find sequences of surface pairs in the scene that match sequences of surface pairs in a model. As an example, suppose that the pair (a, b) of scene surfaces has found a correspondence with the pair (v, w) of surfaces in some model. The next step is to search for a correspondence between some pair in the scene that starts with surface b and one of the matched model pairs that start with surface w, and so on, until all surface pairs are examined. Therefore, this procedure allows identification of a sequence of model surfaces with the same spatial relation (angles) among them as a given set of scene surfaces. It is possible that more than one model are selected as a result.

In the final verification step the sequences of surfaces are analyzed for each model, using a search tree which contains all the previously selected surface pairs. The best pairs combination corresponds to a path through the tree where no surface is crossed twice and the second element of the final pair is the same as the first element of the initial pair. The surfaces in this path are the ones identified as being present in the scene. A priority list is then elaborated containing all the selected models, ordered by decreasing number of identified surfaces. The list is further simplified by eliminating models with surfaces already identified at a higher priority.

[1]the centroid coordinates are the averages of the corresponding coordinates of the vertices defining the surface, i.e., $c_s = (\bar{x}_s, \bar{y}_s, \bar{z}_s)$ and $C_{ks} = (\bar{x}_{ks}, \bar{y}_{ks}, \bar{z}_{ks})$.

4.2 Positioning

Locating the position of the identified model in the scene is equivalent to reconstructing that model in the scene. This is achieved by a sequence of geometric operations, translations and rotations operated in 3D on the model surfaces, which transform from the model to the scene coordinate system. This sequence of operations is described next.

1. Draw a matched model surface in the scene coordinate system.
2. Translate the scene and model surfaces such that both their centroids coincide with the origin of the system.
3. Rotate the model surface such that its normal coincides with the normal of the scene surface.
4. Rotate the model surface until one of its edges coincides with one edge of the scene surface.
5. Translate both scene and model surfaces to the original position of the scene surface.

This positioning algorithm can be implemented by two transformation matrices, MT_1 and MT_2. MT_1 expresses the translation of item 2 and the rotation of item 3. The rotation angle θ_1 and axis $\vec{v}_r$ for this matrix are calculated from the normal vectors of the matched scene and model surfaces, n_c and n_m, respectively, that is,

$$\cos\theta_1 = \vec{n}_c \cdot \vec{n}_m \vec{v}_r = \vec{n}_m \times \vec{n}_c \tag{23}$$

Matrix MT_2 formulates the rotation of item 4 and the translation of item 5, of the positioning algorithm. For that, the normalized vectors pointing to the middle point of the two matched edges, pm_c for the scene and pm_m for the model, are calculated to yield the rotation angle θ_2:

$$\cos\theta_2 = \frac{\overrightarrow{pm}_c}{||\overrightarrow{pm}_c||} \cdot \frac{\overrightarrow{pm}_m}{||\overrightarrow{pm}_m||} \tag{24}$$

The rotation axis for MT_2 coincides with the normal vector of the scene surface.

5 EXPERIMENTAL RESULTS

The tests conducted can be divided in two groups, one for simulated scene images and another for real scene images. Simulated scenes were constructed using a CAD tool to generate the three 2D views. Real scenes were captured by digitizing camera images and also by using a digital photographic camera.

For the simulated scenes the pre-processing phase is not needed, since the CAD generated views are already wire-frame images of the objects in the scene. Segmentation and 2D vertex extraction were implemented in the CAD tool environment, using LISP. All the subsequent phases, including 2D data crossing for the definition of the 3D vertices, 3D attribute calculation as well as object identification and positioning have been implemented in C, compiled and executed on a 200MHz PENTIUM processor.

For each group of tests, the present work shows results obtained considering both, scenes with isolated and with superimposed objects (partial occlusion). Performance parameters analyzed are the processing time and the recognition rates for objects and surfaces in the scene.

5.1 Simulated Scenes

Figure 6 shows a scene with an isolated object and Figure 7 shows another scene with two superimposed objects. Table 1 gives, for each scene, the surfaces completely described in 3D, obtained from data crossing of the 2D vertices. Those are the surfaces analyzed in the remaining phases of the process.

Identification consists of the hypotheses generation and hypotheses verification phases. Verification of hypotheses is done in two steps, one for model identification by matching surface pairs and another to identify surfaces by analysis of the search tree containing the pairs. Table 2 shows the results of hypothesis generation for the two scenes considered. Tables 3 and 4 show the hypothesis verification results, with the “x" indicating the set of surfaces for which a match was found in the model.

For Scene 1, Table 3 shows that the set formed by surfaces s1 (top 1), l1 (side 1), l2 (side 2) and f2 (front 2) is present in at least one group of surfaces of model 8. Those groups are analyzed next to identify the coincident scene/model surfaces. In this example, model surfaces 4, 5, 3 and 8, respectively, were found to coincide with s1, l1, l2 and f2. Therefore, the correct model (model 8) has been identified and so have its surfaces which match the visible scene surfaces.

Sixteen single-object scenes, such as Scene 1, have been analyzed in the tests. In all cases the objects were correctly identified, which means that an efficiency of 100% in the identification of isolated objects has been achieved. A total of thirty-six surfaces are visible in these scenes, twenty-four of which managed to be described in 3D by the data crossing algorithm, duplicates excluded. Out of those, twenty-two have been correctly identified, yielding a 91% surface recognition rate. The two remaining surfaces were rejected. No surface has been wrongly identified.

Positioning of the identified model surfaces in Scene 1 produced an average error of less than 0.5 cm, as shown in Table 6. The error is calculated as the distance between corresponding scene and model vertices. Object dimensions vary from 5 to 10 cm, which means that an average error less than 10% of the sallest dimension was obtained. In practice, this means a quite accurate positioning.

Table 6 also shows the processing time required for identification and positioning of the objects in the analyzed scenes. This time is always less than one second, which would be acceptable for most real time industrial applications.

For Scene 2 four surfaces have not been recognized (s1, l3, f3 and f4), out of nine surfaces described in 3D (Table 4). False vertex identification, due to vertices which are not visible or are mistaken by another due to superposition in the top view are the cause of this problem. However, the recognizable surfaces (s2, l1, l2, l4 and f2) allow correct identification of the models in the scene (models 7 and 8). These results are shown in Table 4.

Seven multiple-object scenes with varying degree of occlusion have been analyzed, containing a total of fifteen objects. Out of these, ten have been correctly identified and only one has been wrongly identified. Therefore total (raw) rates of 66% for recognition and 6,6% for error have been achieved. The remaining objects have not been identified. Considering only the cases where an identification has been performed, the correct recognition rate is 91%. As expected, a decrease in the recognition rate, compared to the isolated objects images, is observed. This is due to the fact that now the objects are partially occluded in some of the views, therefore reducing the number of visible surfaces and making data-crossing more difficult. Objects with a high degree of occlusion are not recognizable.

The effects of partial occlusion are particularly serious for scenes with superimposed objects, since the top view is used as reference for the others. In this case, surface identification is possible as long as it is visible in one of the two other views. When the objects are spreaded apart on the table, the view from the top is free of obstacles, making it easier to cross data at the top view.

Surface identification rates are also lower for multiple-object scenes. Thirty-five surfaces have been described in 3D (duplicates off) out of forty-eight closed regions extracted from the scenes. Some of these regions correspond to fragments of partially occluded surfaces. A total of seventeen of the described surfaces have been correctly identified, which means a 48.5% recognition rate. The remaining eighteen surfaces were not recognizable by the system. No wrong identification of surfaces, which could lead to object recognition error, occurred. Therefore, considering only the cases where an identification was performed by the system, a positive surface recognition rate of 100% has been achieved.

5.2 Real Scenes

Figures 8 and 9 show the views for scenes 3 and 4, two scenes composed with real objects, acquired by digitizing three camera images. The images are of size 450×300 pixels, with 256 gray levels and a sampling rate of 75 pixels/inch. The average camera/object distance was 60 cm, the lens focal length was 100 mm and the amplification factor 4.16.

The digitized images, converted to ASCII format files, are initially submitted to the pre-processing operations described in Section 3: gradient calculation, first contour extraction, second contour extraction (one pixel wide) and surface (closed regions) vertices identification. Next the camera geometric transformation is applied in order to obtain the vertices coordinates in real world length units, followed by the data crossing operation which produces the desired 3D vertices. Finally, object recognition and positioning is performed.

Figures 10 to 12 show the pre-processed images for scene 4. Tables 1 and 5 give the segmented surfaces for each scene and the final recognition results, respectively.

For the scenes with real objects used in the tests, raw recognition rates of 83.3% for models (5 out of 6) and 71.8% for surfaces (23 out of 32 described surfaces) have been obtained. There was no case of wrong object/model identification, which means that the system was completely reliable: 100% of the decisions taken (recognitions performed) were correct.

6 CONCLUSION

A new method for recognition of 3D objects has been presented, which performs hypotheses generation by matching geometric attributes of scene surfaces with attributes of objects polyhedral models. Spatial relations between sequences of matched scene/model surfaces are used for hypotheses verification. Positioning of the identified model in the scene is performed by two geometric transformations. The main features of the proposed method are:

- *fast convergence* - the analysis technique employed, which groups by models the matched pairs of surfaces, yields very fast convergence, as can be seen by the low processing times obtained;
- *accuracy and reliability* - both for isolated objects and for scenes with partial occlusion, high recognition and low error rates were obtained, attesting to the accuracy and reliability of the system;
- *light computational effort* - a very straightforward attribute matching technique, the use of few geometric features and the initial restrictions placed on hypotheses generation, all lead to a low computational load and, consequently, high speed of operation;
- *surface identification* - as an original feature, this method identifies all scene surfaces that can be described in 3D, used in the object recognition process.

Valid performance comparison with other existing techniques would require the use of a common implementation plataform and tests to be realized with the same set of objects and images. However, most of the work reported in the literature utilizes very specific sets of test objects, and very often no objective performance evaluation is provided (Oshima and Shirai, 1983; Boles and Horaud, 1986; Salari and Balaj, 1991). Additionally, the diversity and fast evolution of computers renders almost meaningless the comparison of processing times for different methods.

Nevertheless, if the amount and complexity of the operations involved is considered, it can be concluded that the present method is less complex, and therefore can be expected to perform better, as far as speed of operation is concerned, than other well known model-based 3D recognition techniques (Grimson and L.-Pérez, 1984; Faugeras and Herbert, 1986; Liu and Tsai, 1990; Chiou *et alli*, 1992), as long as dealing with objects that can be at least partially represented by polyhedral models. Since recognition of only a few surfaces is enough to guarantee object recognition, this restriction in practice does not significantly reduces the applicability of the method, while allowing for low algorithm complexity.

Dedicated parallel hardware has been designed to implement the pre-processing and segmentation stages, up to the 2D vertex extraction from the view images. This assures maximum speed, since the corresponding operations will be performed in real-time (during image acquisition). Software implementation starts with 3D vertex extraction by data-crossing and goes up to recognition and positioning, according to the algorithms described in this work. Compared to other conventional implementations, this hardware/software codesign approach will result in a faster and more reliable system, adequate for a large range of industrial inspection applications.

REFERENCES

Arman, F. and Aggarwal, R. (1993). Model-Based Object Recognition in Dense-Range Images - A Review. *ACM Computing Surveys*, 25(1):5–43.

Besl, P. J. and Jain, R. C. (1985). Three-Dimensional Object Recognition. *ACM Computing Surveys*, 17(1):75–145.

Boles, R. C. and Horaud, P.(1986). 3DPO: A Three-Dimensional Part Orientation System. *The Int. Journal of Robotics Research*, 5(3):3–25.

Borges, D. L. (1996). 3D Recognition by Parts: A Complete Solution using Parameterized Volumetric Models *Anais do SIBGRAPI 96*, Caxambu, MG, Brazil,pp.111–117.

Chen, C. H. and Kak, A. C. (1989). A Robot Vision System for Recognizing 3-D Objects in Low-Order Polynomial Time. *IEEE Trans. Systems, Man, and Cybernetics*, 19(6):1535–1563.

Chin, Roland T. and Dyer, Charles R. (1986). Model-Based Recognition in Robot Vision. *ACM Computing Surveys*, 18(1):67–108.

Chiou, R., Hung, K., Guo, J., Chen, C., Fan, T. and Lee, J. (1992). Polyhedron Recognition Using Three-View Analysis. *Pattern Recognition*, 1(25):1–16.

Cortez, P. C. and de Carvalho, J. M. (1995). Modelagem Poligonal de Contornos 2D Usando a Transformada de Hough. *Anais do XIII SBT*, Águas de Lindóia, SP, Brazil.

de Carvalho, J. M. and Hanson, J. V. (1984). Real-time interpolation with cubic splines and polyphase networks. *Canadian Electrical Engineering Journal*, 11(2):64–72.

Dhond, Umesh R. and Aggarwal, J. K. (1989). Structure from Stereo; A Review. *IEEE Trans. Systems, Man, and Cybernetics*, 19(6):1489–1534.

Farias, M. F. S. and de Carvalho, J. M. (1997). Modelagem Geométrica por Sistemas CAD para Visão Computacional. *Anales XII Congreso Chileno de Ingenieria Eléctrica*, Temuco, Chile, pp. II:579–584

Faugeras, O. D. and Herbert, M. (1986). The Representation, Recognition and Locating of 3D Objects. *The Int. J. Robotics Research*, 5(3):27-52.

Grimson, W. L. and L.-Pérez, T. (1984). Model-Based Recognition and Localization from Sparse Range or Tactile Data. *The Int. J. Robotics Research*, 3(3):3-35.

Klemt, A. and Infantosi, A. F. C. (1995). Classification of Surface Normals in a Tutorial Microp.-Based System for Anatomy. *Proc. of 38th. MWSCAS*, Rio de Janeiro, RJ, Brasil, 2:1373–1376.

Liu, C. and Tsai, W. (1990). 3D Curved Object Recognition from Multiple 2D Camera Views. *CVGIP*, (50):177–187.

Marshall, A. D. and Martin, R. R. (1992). *Computer Vision, Models and Inspection*. World Scientific, Singapore.

Melcher, E., de Carvalho, J. M., Barros, M., Naviner, L., Naviner, J. F., et al. (1996). A Novel Gradient Operator Suited for VLSI Implementation of 2D Shape Recognition. *Anais do IX SBCCI*, Recife, PE, Brazil, pp. 321–332.

Melcher, E., de Carvalho, J. M., Cortez, P. C., Naviner, L., Naviner, J. F. (1996a). A Vertex Detection Algorithm for VLSI Implementation. *Anais do SIBGRAPI 96*, Caxambu, MG, Brazil, pp. 367–368.

Melcher, E., Naviner, L., de Carvalho, J. M., Naviner, J. F., Moreira, R. A. S., Villar, Y. M. and Morais, M. (1997). VLSI Implementation of Contour Extraction from Real Sequences. *VLSI'97*, Gramado, RS, Brazil.

Murray, D. W. (1987). Model-Based Recognition Using 3D Shape Alone. *CVGIP*, (40):250–266.

Oshima, M. and Shirai, Y. (1983). Object Recognition Using Three-Dimensional Information. *IEEE Trans. Patt. Anal. Machine Intel.*, 5(4):353–361.

Requicha, A. A. G. (1980). Representation for Rigid Solids: Theory, Methods and Systems. *ACM Computing Surveys*, 12(4):437–464.

Salari, E. and Balaji, S. (1991). Recognition of Partially Occluded Objects Using B-Spline Representation. *Pattern Recognition*, 24(7):653-660.

Villar, Yuri M. (1997). ASIC Para Extração de Vértices de Imagens em Tempo Real. *Masters Thesis in Electrical Engineering - UFPB - Brazil.*

Table 1: Visible 3D surfaces

view	scene 1	scene 2	scene 3	scene 4
top	1	1 2	1	1 2 3
side	1 2	1 2 3 4	1 2	2 3
front	2	2 3 4	2	3 4 5 6 7 8

Table 2: Hipothesis generation

Scene	view	scene surf.	Vi=(model; matched surf.)
1	top	1	(8; 4 7)
	side	1	(6; 7 10) (7; 2 3 4 6) (8; 1 2 5 6)
		2	(1; 2 5) (8; 3 8)
	front	2	(8; 3 8)
2	top	1	(1; 3 6) (6; 7 10) (7; 3 6) (10; 7 10)
		2	(7; 3 6)
	side	1	(8; 1 2 5 6)
		2	(1; 2 5) (8; 3 8)
		3	(7; 1 2 3 4 6) (8; 1 2 5 6)
		4	(6; 7 10) (7; 2 3 4 6) (8; 1 2 5 6)
	front	2	(8; 3 8)
		3	(7; 1 2 3 4 6)
		4	(4; 10 11 12 13 15)

Table 3: Verification Scene 1

s1	l1	l2	f2	Mod
x	x	x	x	8
4	5	3	8	8

Table 4: Verification - Scene 2

s1	s2	l1	l2	l3	l4	f2	f3	f4	Mod
-	-	x	x	-	-	x	-	-	8
-	x	-	-	-	x	-	-	-	7
-	-	5	3	-	-	8	-	-	8
-	3	-	-	-	2	-	-	-	7

Table 5: Recognition of real objects

	scene 3	scene 4
model	1	5 7
surfaces	2 3 1 6	(5; 6 11 1 3) (7; 1 3 5 6)

Table 6: Positioning and processing time

SCENE	1	2	3	4
d (cm)	0.45	0.96	0.42	0.62
t (seg)	0.51	0.6	0.28	0.82

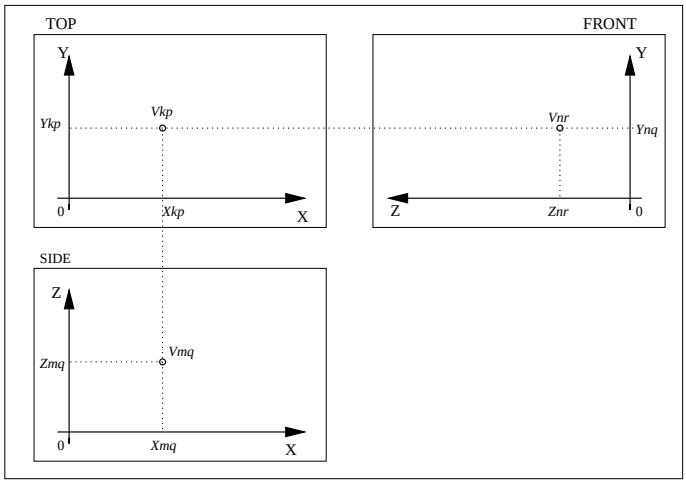

Figure 5: Point Correspondency for Data Crossing

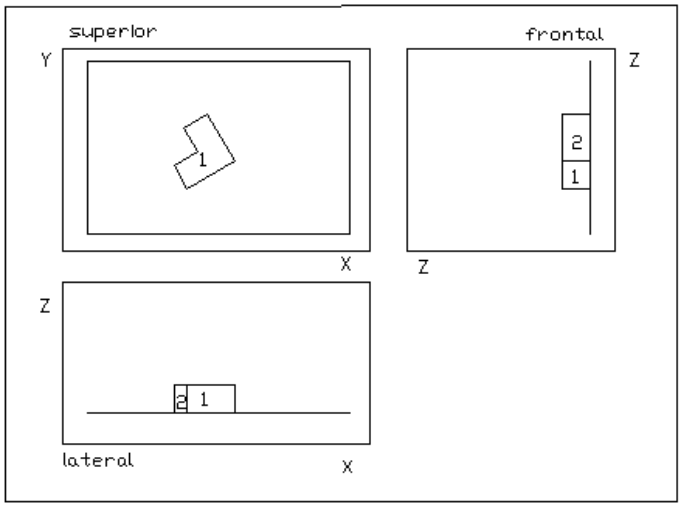

Figure 6: Scene 1 - top, side and front views

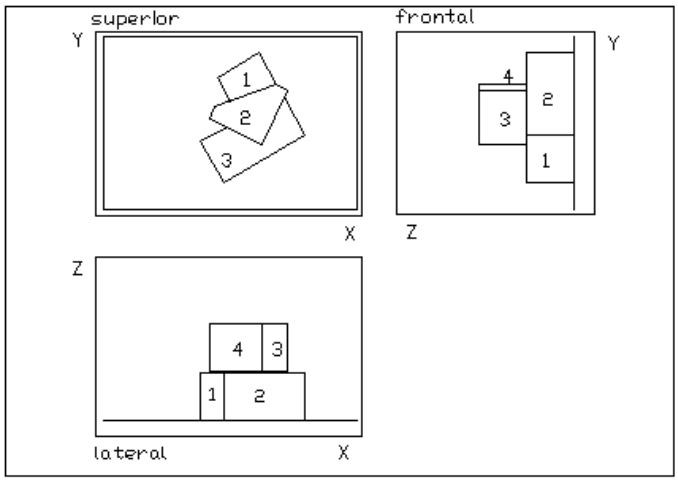

Figure 7: Scene 2 - top, side and front views

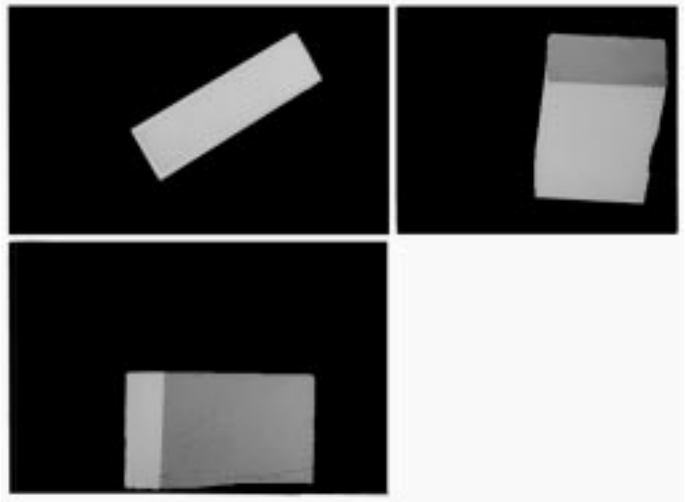

Figure 8: Scene 3 - top, side and front views

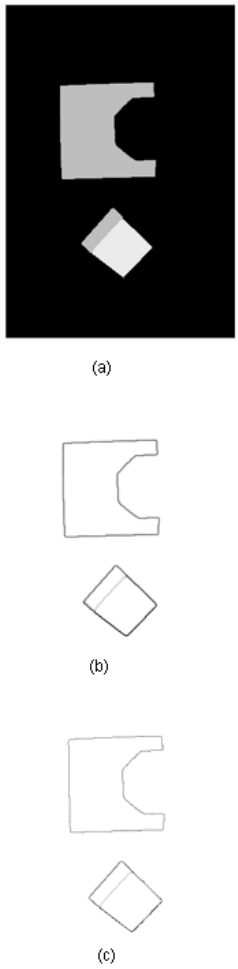

Figure 10: Scene 4: (a) top view, (b) gradient and (c) final contour

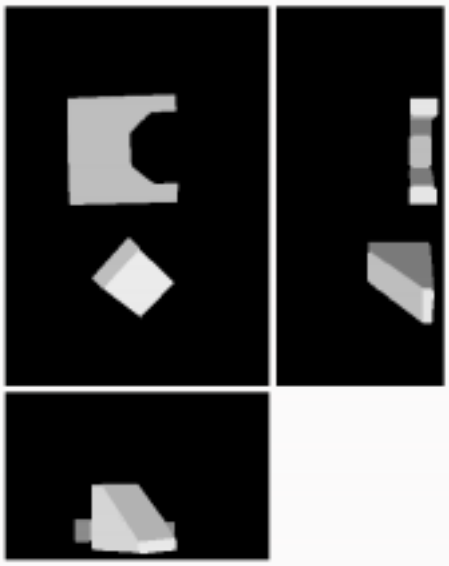

Figure 9: Scene 4 - top, side and front views

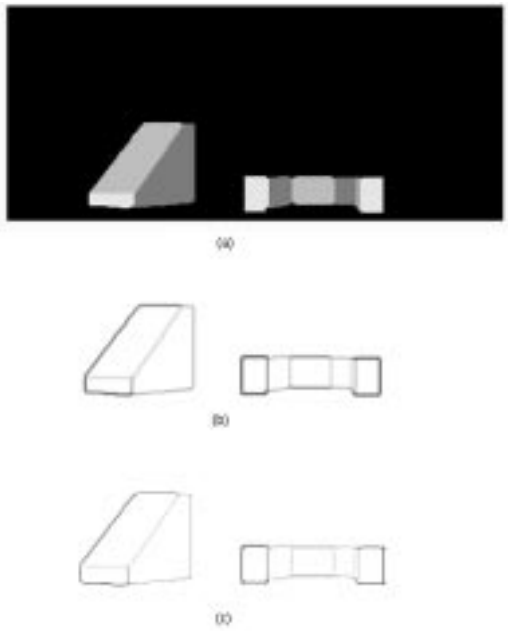

Figure 11: Scene 4: (a) front view, (b) gradient and (c) final contour

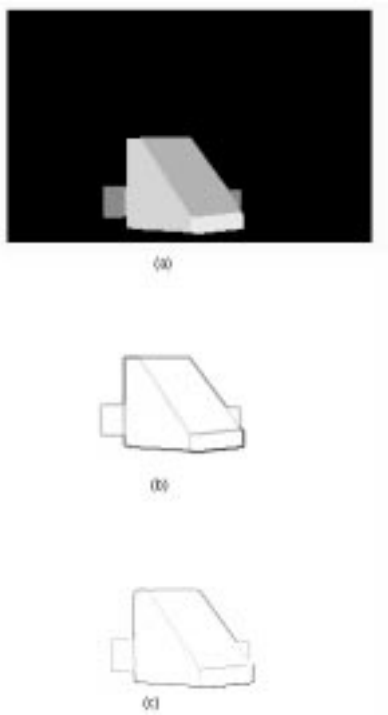

Figure 12: Scene 4: (a) side view, (b) gradient and (c) final contour