Building the Edge-Junction Graph from a Range Image of Curved Objects

Guy Godin Martin D. Levine

Computer Vision and Robotics Laboratory
McGill Research Center for Intelligent Machines
McGill University
Montréal, Québec, Canada

Abstract

This paper presents an approach to the computation of a representation for planar- as well as curved-face objects in range images. The result is an edge-junction graph embedding qualitative and quantitative descriptors of the structure of the discontinuities in the viewed scene. Edges are detected using local operators and triangulation shadow boundary analysis. Edge points are organized into depth- and orientation-continuous segments. Finally, junctions are built from the extremities of these segments by appealing to certain results from line drawing analysis. The resulting structure corresponds to a partial wireframe of the objects.

Sommaire

La méthode proposée ici permet d'extraire, à partir de l'image télémétrique d'objets aux faces planes et courbes, une description reposant sur les discontinuités présentes dans l'image. La nature tridimensionnelle des données est mise à profit aussi bien dans l'extraction que dans l'organisation des lieux de discontinuité. Les arêtes, détectées par des opérateurs locaux ainsi que par analyse des bordures d'ombres, sont divisées en segments continus en profondeur et en orientation. Les extrémités de ces segments sont groupés en jonctions, en accord avec les configurations derivées des techiques d'analyse de dessin de traits.

Keywords: *range image analysis, scene description, edge detection, triangulation shadows.*

1. Introduction

The research reported in this paper concentrates on the generation of a structured description based on the edges detected in a range image. These edges represent the discontinuities in depth and surface orientation of the surfaces visible in a single range image obtained from a laser rangefinder. The knowledge of the three-dimensional positions of these discontinuities given by the range data permits the construction of a *partial wire-frame* representation. The integration of several of these partial wire-frames, each extracted from a single range image, is a possible avenue for object representation from multiple range images.

Although the results reported in this paper do bear a strong similarity to conventional line-drawing analysis, a distinction needs to be made. Line drawing analysis aims at recovering the types of discontinuities represented by lines, according to the appearance of the junctions of the lines representing edges *in the projection*. These lines may be obtained in principle from human input or edge extraction in intensity images. A range image *explicitly* contains the geometric information relevant to the identification of the type of edge. Therefore, there is no need to actually infer the line type; it can be directly discerned from the edge detection process.

The method described in this paper deals with the extraction and organization of the discontinuities and their junctions, formed by the projection of vertices, curved surfaces and occlusions. Discontinuities in depth and surface orientation are computed from the range data, and structured in the form of a line drawing with supplementary information on the 3D position and orientation of edges at the junction. The resulting structured edge map is in the form of an *edge-junction graph*. The extracted edges and junctions are located in 3D, thus constituting a wire-frame of the discontinuities visible from one viewpoint.

In previous research, much effort has been devoted to the extraction of particular surface primitives from images such as planes, cylinders, cones and spheres. Other research efforts aim at describing the surface in terms of local curvature properties [2, 4, 25] . Edges, or discontinuities in depth or surface orientation, have also been used as segmentation features. A common goal of all these approaches is the decomposition of the image into homogeneous elements, according to a specific geometric criterion. These intermediate scene descriptions may then be employed for higher level interpretation.

Edge extraction and organization is the basis for several methods that have been presented for the analysis of range images. Techniques for the local detection of jump and crease edges have been proposed which analyse local surface height and orientation [6, 11, 13, 18], the scale-space behaviour of features [20], the frequency spectrum of a circular sampling [12], depth coupled with the intensity (or reflectance) [8, 16], and level curves [1, 15]. Edge-oriented scene description methods take this one step further by attempting to organize the detected edges into a meaningful and useful format, often performing additional analysis on the surfaces enclosed by the edges [3, 7, 14, 22, 24].

Herman [9] has presented a system for the extraction of a detailed scene description from the range image of polyhedra. The points are detected using local operators and are grouped into lines by separately applying a 2D Hough transform to each class of edge. A line drawing is then formed by merging segments that are collinear in the plane and extending and shortening lines to form junctions between sufficiently close points.

An a priori knowledge about the possible junctions in a

restricted world can be employed to direct the computation of the scene description. The system built by Sugihara [23] uses such an approach for the structured extraction of vertices, edges and faces from range images of planar and/or curved objects. Knowledge about the world is encoded in the form of a junction dictionary. The redundancy between this information and the range data is exploited to improve performance in the presence of noise by predicting missing edges. The algorithm also uses knowledge during the crease edge extraction process. This knowledge, in the form of a junction dictionary, assumes certain constraints on the original scene. Thus, only trihedral vertices are considered, although junctions of curved objects bounded by quadric surfaces are actually treated by using their corresponding approximation as a polyhedral junction and adding one new specific junction type to the dictionary. A general viewpoint is assumed, so that a slight variation in the viewing position does not result in a different structure for the line drawing. Since the edge *extraction* process is guided by the dictionary, the system can only recognize and produce junction types that are known to it. Any configuration which does not conform to expectations is likely to produce an erroneous interpretation, if any.

The method proposed here can be seen as extending the research of Sugihara [23] and Herman [9]. We attempt to properly characterize curved objects and truly exploit the tridimensionality of the data. The goal of this research is the computation of an edge-based description where the edges and junctions are extracted and characterized in *three-dimensional* space. In order to maintain generality, the number of constraints on the scene and viewpoint should be made as small as possible. Constraints on viewpoint are particularly significant since, while it may be feasible to determine in advance the class of objects to be encountered in a given application, their position in front of the camera cannot in general be predicted. In addition, any realistic application cannot be restricted to a polyhedral world. Thus, curved objects should be handled as well, and their particular features explicitly extracted. The wire-frame construction requires that depth discontinuities resulting from occluding convex edges be discriminated from those caused by smooth surfaces curving away from the viewer. This step, which is absent from [23], is necessary to the wire-frame construction since only the jump edges resulting from occluding convex edges actually correspond to physical edges on the objects. The system should also be expandable, in the sense that even if it is tailored to a particular class of objects, it should be possible to add new knowledge without major restructuring. Finally, in the case of scenes with unexpected elements, these should be identified without affecting the rest of the analysis.

The system proposed here computes a line drawing similar to what Sugihara and Herman have obtained but fully uses 3D information to meet the above goals. Discontinuities in depth and surface orientation are first extracted and grouped into segments. Junctions are built from these segments according to rules adapted from an a priori knowledge of the class of permissible objects in the scene. Certain results related to line drawing analysis of curved objects will be employed. It will be seen that the full use of three-dimensional information is crucial to this approach. This permits analysis without general viewpoint restrictions, while providing much more reliable techniques for feature detection. Examples using real range images will demonstrate the applicability of the method to complex scenes.

The images employed in this research were provided by the National Research Council of Canada (NRCC) and obtained from an active triangulation laser rangefinder based on the synchronized scanning principle [21] . The images are in the form of a two-dimensional (typically 256 $\times$ 256) array of floating-point (x, y, z) coordinates. The row and column coordinates, here called u and v, represent the original sampling grid corresponding to the geometry of the scanning mechanism. They differ in general from the cartesian x and y coordinates of the projection of the space into the image plane. Figure 1 illustrates the original $z(u, v)$ image.

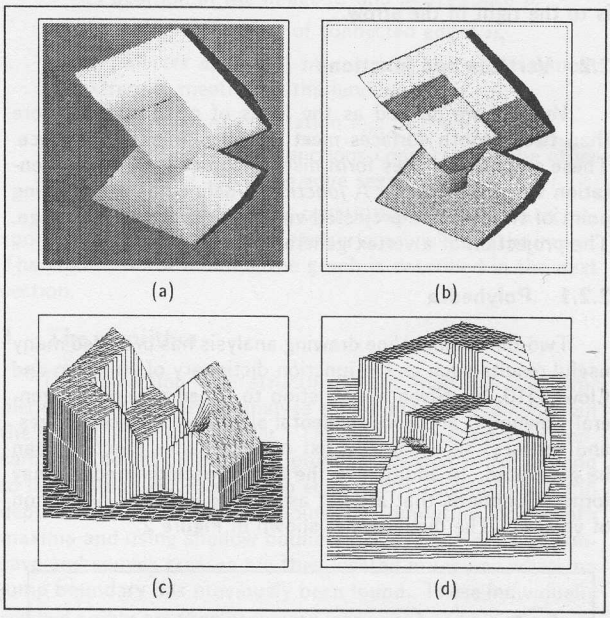

Figure 1 Range image: (a) original image of $z(u, v)$; (b) a shading algorithm is applied to the data; (c) and (d) show two projected surface representations.

The following sections present a description of the object world under consideration and the proposed format for scene description. Then, the algorithms for edge detection, shadow analysis, contour segmentation and junction construction are described and results obtained with real range images are discussed.

2. Edge description of objects

2.1 Objects, surfaces and edges

The family of objects that can be analysed by this approach should be as unrestricted as possible. In our case, since the appearance of junctions is not necessary for edge type inference, restrictions of general viewpoint and planar trihedral vertices are not required. Objects will be assumed to be composed of smooth, possibly curved surfaces that intersect at crease edges that are sharp (to be defined later). The smoothness requirement is implicit in many active range imaging applications since existing sensors do not perform adequately on textured surfaces.

The boundaries between two smooth portions of surfaces form *creases*, either *concave* or *convex*. In the 2D image of a scene, convex edges may appear with only one bounding surface visible. This is called an *occluding* edge and causes a discontinuity in depth. Finally, since curved objects are

also permitted, *limb* edges may occur at positions where a surface curves away from the viewer. Limbs appear as depth discontinuities where the normal on the surface closest to the viewpoint is orthogonal to the line of sight. The existence and position of convex and concave edges on the object are independent of the viewpoint, whereas occluding and limb edges depend on the object pose as well as viewpoint.

By convention, convex edges in the image are labelled "+", concave are labelled "−", occluding edges "→", and limbs "↠". For the latter two, the arrows are oriented such that the surface closest to the viewer (the occluding surface) is to the right of the arrow.

2.2 Vertices and junctions

Vertices are defined as the locus of points where more than two smooth surfaces meet in three-dimensional space. These smooth surfaces form discontinuities in surface orientation when they meet. A *junction* is defined as the meeting point of two or more projected visible edges in the 2D image. The projection of a vertex generally forms a junction.

2.2.1 Polyhedra

Two-dimensional line drawing analysis has provided many useful results such as the junction dictionary of Huffman and Clowes [10, 5]. Given a restriction to trihedral vertices, general viewpoint, and non-accidental alignment of faces, edges, and vertices, an exhaustive list of all possible junctions can be generated. In addition to the above, occluding edges may form T junctions that are not associated with the projection of vertices. The dictionary is shown in Figure 2.

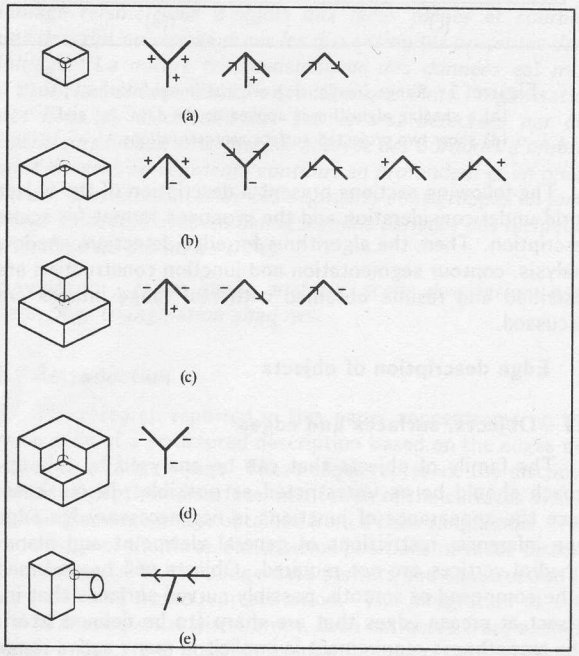

Figure 2 The Huffman-Clowes dictionary of trihedral junctions. The possible corresponding junctions are listed for each trihedral vertex occupying 1 (a), 3(b), 5(c), or 7(d) octants. Occlusions may form T junctions (e), where edges of any type (*) can be occluded.

2.2.2 Curved objects

Malik [17] has shown that a junction dictionary for curved objects can be rigorously generated if suitable restrictions are placed on the class of surfaces, general viewpoint is assumed, and particular surfaces are excluded. He has demonstrated that vertices, here defined as points where more than two planar or curved surfaces meet, project as would do their tangent polyhedral approximations. A conical surface with its apex is excluded since it does not possess a tangent plane. Junctions involving limb lines also exist in (the projection of) the curved world. They are located either along a surface orientation discontinuity or isolated on a smooth surface. Creases meet limbs according to a fixed set of possibilities, similar to the ones generated by a cylinder or a cylindrical hole. The connection between limb and crease in the projection shows no tangent discontinuity in the projection, but in general does show discontinuity in curvature [19] . Smooth curved surfaces can be self-occluded, as is the case of a bump seen sidewise. Thus, the notion of junction must also include terminal points of lines. Figure 3 illustrates this new set of junctions for curved objects. The reader is referred to [17] for a detailed discussion of the procedure for creating the dictionary.

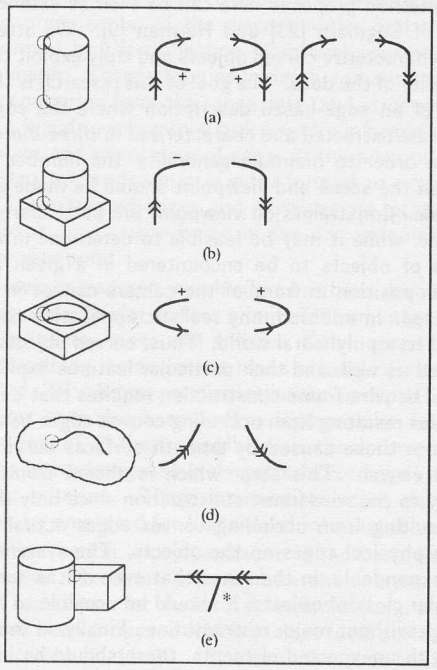

Figure 3 Malik's dictionary of curved junctions. (a)(b)(c) intersections of limbs and edges; (d) surface self-occlusion; (e) T junction involving limb and edge of any type (*).

2.2.3 Extended junction family

The necessity of permitting accidental alignment of faces to occur was emphasized before. This requires modification of the dictionaries. It will be seen that for polyhedral junctions accidental alignment will be acceptable without an explicit listing of all the possibilities. This is achieved by considering the actual three-dimensional spatial configuration producing the junctions. For curved objects, additional junctions caused by accidental alignment are added to the repertory of cylindrical junctions (see Figure 4(a)). Furthermore, new junctions

created by some surface types not part of the object world defined by Malik can be added: for example, the apex of an isolated cone, and a pair of cylinders meeting at an angle (Figure 4(b)). There is no reason to believe that these extensions cover the complete set of possible junctions that can occur in the projection of edges, given the general class of surfaces permitted. But the approach adopted here does not require completeness of the junction dictionary, as with conventional line-drawing labelling problems.

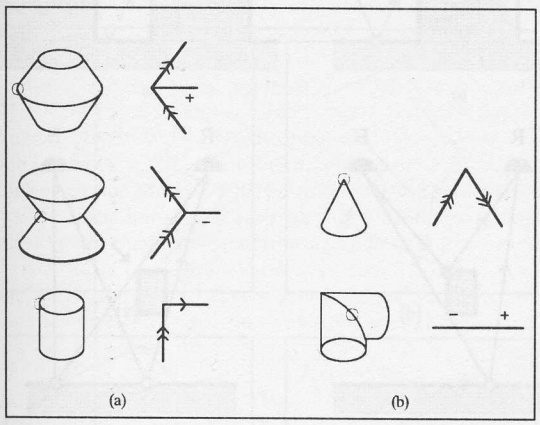

Figure 4 Additional junctions. Supplementary entries corresponding to (a) accidental alignment of curved junctions and (b) types of objects forbidden in Malik's world.

2.3 Edge-junction graph representation

The proposed edge-based scene description is an *edge-junction graph*. This format is similar to the line drawing and offers a structured representation of the edge map obtained from a single-view range image. It embeds quantitative geometric information of position and orientation as well as qualitative descriptors of edges and junctions.

At this point, the notion of a junction needs to be slightly enlarged to include any *significant* events occurring on the edges in the projection (the image). Three categories of junctions can be deduced from the previous discussions. The first category corresponds to the projection of vertices and limb-crease junctions: two or more lines of different type or orientation meet. These are the polyhedral (as in Figure 2) and curved junctions (Figures 3(a,b,c) and 4). A second category corresponds to endpoints of smooth surface self-occlusions (Figure 3(d))and stems of T junctions (Figures 2(e) and 3(e)). The third category is introduced to include closed smooth contours such as the outline of a sphere. It differs from the previous two by not having a fixed location in space, since it is not linked to a particular point of interest on the edge forming a closed contour. Any point on the contour can be chosen as the junction, thereby defining an arbitrary starting and ending point of the edge segment.

The edge-junction graph is a planar graph whose nodes represent junctions and whose arcs are formed by the edges. Formally, the edge-junction graph G is represented by k junctions (nodes) S_i, $(0 \leq i < k)$, and l edge segments (arcs) B_j, $(0 \leq j < l)$. The graph may in fact be composed of several disconnected subgraphs, although it will still be referred to as a single graph.

Each edge segment B_j is described by:

- its type, chosen from the set $\{ +, -, \rightarrow, \twoheadrightarrow \}$
- the 3D and image coordinates of the ordered set of points composing it
- possibly other global descriptors of the edge segment shape

Each junction S_i is described by :

- its category c_i (*polyhedral vertex, curved junction, self-occlusion, T occlusion,* or *closed contour*)
- its configuration f_i, corresponding to the entry in the appropriate dictionary
- its position in the image h_i and in 3D space p_i
- the order, or number of connected edges n_i
- n_i pointers a_{im} $(0 \leq m < n_i)$ to the edge segment extremity meeting at the junction
- the 3D tangent orientation v_{im} of the n_i edges
- pointers z_{im} to the junctions connected to the other extremity of the same edge segments

The edge-junction graph constitutes a structure corresponding to the wire-frame of the surfaces visible in the image. The algorithm for building the graph is described in the next section.

3. The algorithm

The extraction of a structured edge map has been implemented and used to analyse range images obtained from the NRCC laser rangefinder. The first processing step concerns the detection of image points estimated to correspond to discontinuities. First, jump edge points (discontinuities in depth, either $\rightarrow$ or $\twoheadrightarrow$) are found by computing local depth maxima and using shadow boundary analysis. Points on concave and convex creases are then located in regions where no jump boundary has previously been found. These individually marked points are then organized into connected C^0 segments without branch points. In the next step, each segment is decomposed into C^1 subsegments using a 3D corner detection method. Occluding edges and limbs are distinguished by an analysis of the neighbouring surface of jump edges. Corners and endpoints are merged into junctions following tests on their type, position and orientation of edge tangents. Finally, junctions and edges are characterized by the relevant information to constitute the edge graph. At present, this has been implemented for a world of planar and curved objects forming trihedral vertices. The processing stages and data flow appear in Figure 5.

The different modules are briefly described in the following sections. The results of each processing step applied to the range image in Figure 1 are also presented. The image is composed of 256 $\times$ 256 pixels or surface sample points (x, y, z). The original scene measures approximately 120mm$\times$120mm, and the difference between the background and the highest point is about 75mm.

3.1 Understanding range shadows

Active triangulation range shadows occur at positions in the image where the portion of surface illuminated by the emitter is occluded from the receiver by the presence of intervening opaque matter. Shadows appear as black areas in Figure 1. Shadows in triangulation range images are generally seen as a nuisance, particularly to an edge-based process, since the desired edge features are unidimensional while shadows appear as areas in the image. Absence of data from parts of the image may occur because of poor object reflectance or

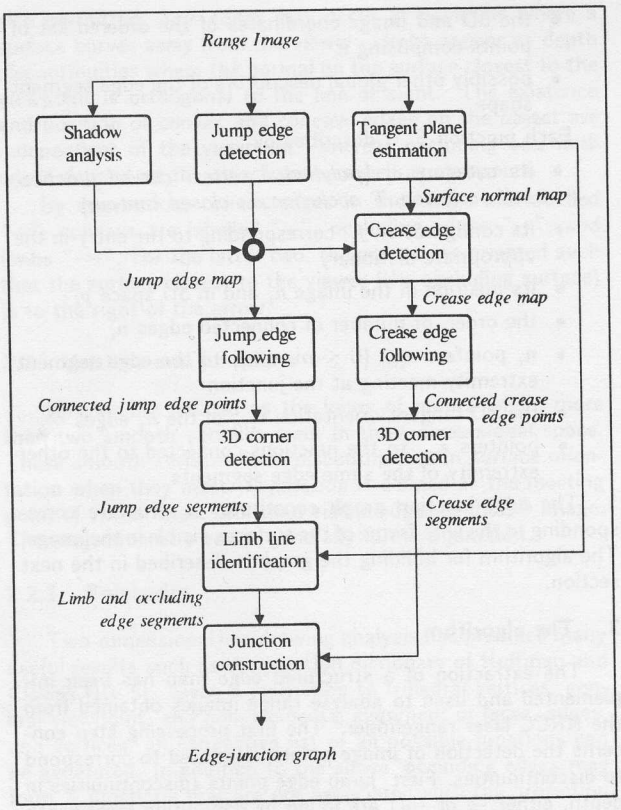

Figure 5 Block diagram of the algorithm.

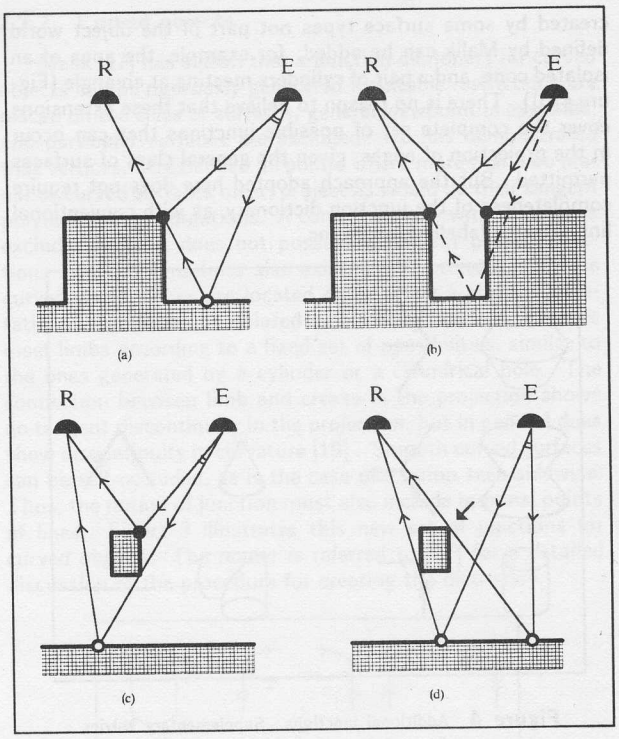

Figure 6 Analysis of shadow formation, with emitter(E), receiver (R), and shadow angle. The relative positions of return light paths are used to infer whether or not jump edges are present (• and ○, respectively) at the boundaries of shadows.

other artifactual behaviour of the sensor. Otherwise, the presence of a triangulation shadow may be considered as a useful indication of a specific surface geometry. Shadows have been studied by Sugihara [23] who has distinguished occluding and occluded linear segments on each side of a shadow by verifying whether both straight line segments are aligned with the camera in space. Smith and Kanade [22] also discuss the analysis of shadow boundaries.

The method proposed here relies on an analysis of the geometry of shadow formation in the triangulation plane formed by the emitter, receiver and illuminated point. The scanning mechanism of the NRCC camera used to obtain the data is arranged such that all points in a single row of the range image share the same triangulation plane. This justifies an analysis of shadow formation in that plane which is a cross-section of the scene. The distribution of opaque matter with regard to the position of both the emitter and receiver determines the locus of the shadows. The existence of jump edges at the surface points that bound the shadow area is deduced from the relative position of the light paths from these points to the receiver.

Figure 6 illustrates some typical configurations. The angular interval of emitted light for which a shadow occurs is indicated. The symbols • and ○ mark whether or not a jump edge should be inferred at the extremities of the shadow interval. Figure 6(a) illustrates a simple *occluding-occluded* relation between the extremities. For this case only, the return paths are collinear and only one side of the shadow corresponds to a jump boundary. In Figure 6(b), the structure on the left cast a shadow that is interrupted on the right side by the presence of another opaque structure. The shadow would otherwise extend to the dashed line. The return paths are not collinear, and both sides of the shadow are marked as jumps. The cases of Figure 6(c) and (d) arise when the cross-section of opaque matter is distributed in disconnected sets. In Figure 6(c), the light beam reaches the lower surface point and travels back to the receiver by a path that is *under* the other shadow boundary point. In this case, the two light paths do not intersect, in distinction to the situation shown in Figure 6(b). Only the surface point corresponding to the highest path is marked as a jump. The same configuration of matter will also give rise to a remote shadow, projected on a surface away from the structure that caused it (Figure 6(d)). In this case, points other than the shadow boundaries are involved in the shadow formation. This case differs from 6(b) by the presence of matter inside the triangle formed by the two shadow boundary points and the receiver. This configuration is recognized if a surface point can be found on the profile that is aligned with the return path on the right, thereby explaining the end of the shadow. This point is indicated by an arrow in Figure 6(d). No edge is marked in this case.

Other configurations can also arise, especially when several disconnected sets of matter are present in the plane. In some cases, no hypothesis concerning the origin of the shadow can be made. Nevertheless, the four configurations shown above cover the most common occurrences of shadow formation in the range images used for our experiments.

The angular interval of shadow appears as a sequence of shadow points along a row of the image. The 3D position

of the data points on each side of the shadow sequence with regard to the receiver's position is used to determine whether the shadow can be explained by one of the configurations of Figure 6. Jump edges are then recorded at their proper positions. Since this analysis can only be performed along rows, the jump boundary must be interpolated between adjacent rows across columns to close the jump edge contours along the shadow area.

Figure 7(a) illustrates the results of the shadow analysis.

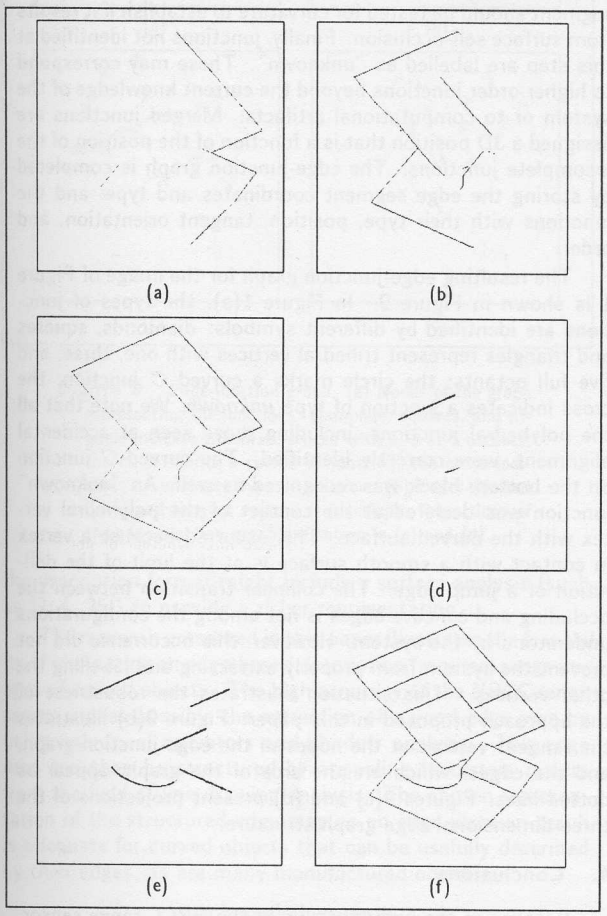

Figure 7 Edges detected in the data shown in Figure 1. Jump edges are obtained from shadow analysis (a) and local differences (b). These are combined in (c). Crease edges may be concave (d) or convex(e). The combined edge map, after thinning of crease edges, appears in (f).

3.2 Edge detection and tracking

In a shadow-free neighbourhood, jump edge detection is performed by detecting local maxima of depth difference that exceed a fixed threshold. The z coordinates of each sequence of four adjacent data points on a row of the image are tested along rows and columns. The differences are computed only along rows and columns to yield points on the 8-connected contour of the boundaries on the highest side between adjacent surfaces separated by a depth discontinuity. The side of the pixel (up, down, left, right) on which the edge transition has been detected is also recorded, and will be used in the edge following procedure. The jump edge map computed from local differences is shown in Figure 7(b). The combined jump edge map appears in Figure 7(c) .

Discontinuities in surface orientation, here called creases, must be detected in the next step. The method employs estimates of the tangent plane at a point by fitting a small plane in a least-squares sense (here over a 3×3 patch). A crease is detected if the angular difference between a pair of points in at least one of the four directions (row, column and diagonals) is larger than a threshold, and if the two planes intersect properly at the middle point to form a dihedral partition. The type of crease detected, either concave or convex, is noted. Crease edge detection is attempted only at locations sufficiently distant from a previously detected jump edge. The resulting crease edge map is thinned to obtain 8-connected pixel-wide edge boundaries. Before thinning, the concave and convex edge maps appear as indicated in Figure 7(d) and (e) for an angular threshold of 30°.

Figure 7(f) illustrates the complete edge map. Each type (concave, convex, jump) of detected edge element is organized into segments that are C^0, that is, continuous in depth. The segments are stored as a set of ordered lists of points. A contour tracking procedure connects jump edge points if no jump discontinuity exists between them. The jump edge segments are ordered such that the surface closest to the sensor is to the right of the boundary as it is followed. Crease edges are also tracked. There can be no depth discontinuity between connected crease edge pixels, otherwise a jump edge would have been detected there. However, branch points may be detected at vertices and must therefore be decomposed into simple segments during the tracking process.

Segments of length below a fixed threshold are considered to be insignificant and are removed. A preliminary edge-junction graph is formed at the end of this step: it comprises the l C^0 edge segments produced by the tracking and the associated $2l$ junction nodes representing the endpoints of these segments. The edge segments are labelled as being either *convex*, *concave* or *jump*.

3.3 3D Corner detection

At each point on a C^0 segment of type *jump* or *crease*, a 3D corner detector is applied to ascertain whether it should be partitioned into C^1 subsegments. Discontinuities in the spatial orientation of the boundary (3D corners) delimiting the subsegments must be detected. The corner detector works as follows. Two 3D lines are fitted in a least-squares sense to the n consecutive points on each side of the point in the segment sequence being considered. The value of n should remain small so that curved boundaries are adequately approximated. Typically, a value of 8 was used. Points that exhibit a local minimum of the total fit error on the left and right sides are detected as corners if the angle between the two lines exceeds a threshold. Corner detection in 3D provides more robustness than 2D methods (as in [23]) because the 3D corner angles are always greater or equal to the angle of their projection in 2D. The method is also immune to accidental alignments, since a corner in three-space is detected even though the edge at the corner may be collinear in the projection.

The graph G is now updated by splitting the edge segments at 3D corners and adding new junctions of order $n_i = 2$ at the split points. The tangent orientation on each side of a corner, as well as at an endpoint, is estimated by the direction of the fitted 3D line and recorded as the vector v_{im} of the corresponding junction.

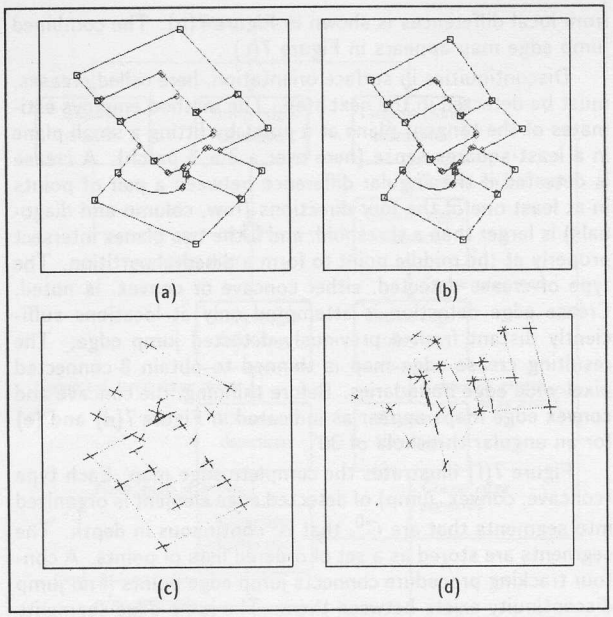

Figure 8 Corners and endpoints bounding the edge subsegments. (a) Corners (squares) and endpoints (diamonds); (b) projection of the edge segments on the xy plane; (c) edges (dotted lines) and tangent vectors projected on the xy plane; (d) another view of the vectors.

Figure 8(a) shows the corners detected on the edges (squares) and the endpoints (diamonds). Corners on jump boundaries that appear as straight lines in the image, due to accidental alignment, were successfully detected. The edge segments are projected on the xy plane in Figure 8(b). The tangent vectors at corners and endpoints appear in Figure 8(c) along with the edges (dotted lines). A projection along another viewing direction (Figure 8(d)) makes clear the three-dimensional nature of the edge positions and tangent orientations.

3.4 Limb and occluding edge detection

Each portion of a C^1 *jump* edge segment must be labelled as being either an occluding convex edge "$\rightarrow$" or a limb edge "$\twoheadrightarrow$". The proposed method is to analyse the surface normal along the side of the edge closest to the sensor. The normal should be almost perpendicular to the line of sight. The type labels of jump edge segments in the graph are adjusted accordingly to either $\rightarrow$ or $\twoheadrightarrow$.

No limb edges are present in our example.

3.5 Construction of the junctions

At this point, the arcs of the edge-junction graph have all been extracted and the nodes represent corners and endpoints of the edge segments. The nodes can then be merged into complete junctions if their relative positions, types, and tangent orientations are in accordance with the known configurations. Junctions of unknown type may also be produced.

First, endpoints of segments with identical labels are connected if they are sufficiently close and their tangent directions indicate that they are collinear. This step fills gaps that might remain after edge detection and following. Then, junction building is attempted between the junctions within a certain distance from each other. Polyhedral junctions are built by verifying the angles in three-space between the (two or three visible) lines meeting at the junctions to be merged. These angles, along with the line labels, help to ascertain whether the junction represents one of the four possible trihedral vertices (Figure 2). This is done by determining the number of occupied octants corresponding to the visible structure of the junction. Curved junctions are then built by consulting the appropriate junction dictionaries (Figures 3 and 4). Occlusion (T) and smooth self-occlusions are verified for the remaining endpoints. T junctions are declared if the edge ends close to a jump edge or a shadow. If isolated, the endpoint of a limb segment should be tested for curvature to establish if it results from surface self-occlusion. Finally, junctions not identified at this step are labelled as "unknown". These may correspond to higher order junctions beyond the current knowledge of the system or to computational artifacts. Merged junctions are assigned a 3D position that is a function of the position of the incomplete junctions. The edge-junction graph is completed by storing the edge segment coordinates and type, and the junctions with their type, position, tangent orientation, and order.

The resulting edge-junction graph for the image of Figure 1 is shown in Figure 9. In Figure 1(a), the types of junctions are identified by different symbols: diamonds, squares and triangles represent trihedral vertices with one, three, and five full octants; the circle marks a curved C junction; the cross indicates a junction of type *unknown*. We note that all the polyhedral junctions, including those seen at accidental alignment, were correctly identified. The curved C junction on the bottom block was recognized as well. An "unknown" junction was declared at the contact of the polyhedral vertex with the curved surface. This occurred because a vertex in contact with a smooth surface is at the limit of the definition of a jump edge. The collinear transition between the occluding and concave edges is not among the configurations understood by the system. However, this occurrence did not prevent the system from properly extracting and labelling the other vertices. This situation illustrates the robustness of the approach proposed in this paper. Figure 9(b) illustrates the tangent vectors at the nodes of the edge-junction graph, and the edges, which are the arcs of the graph, appear as dotted lines. Figures 9(c) and (d) present projections of the three-dimensional edge-graph structure.

4. Conclusions

Because of the high precision of the NRCC range sensor, no prefiltering of the data was necessary to obtain a virtually noise-free edge map using simply local operators. The analysis of shadows has proven to be useful for inferring the presence of jump edges. The elimination of the general viewpoint requirement by the use of three-dimensional information is a significant improvement over previous approaches. The object world we have studied was limited to trihedral vertices on planar and curved objects, but a wider class could be interpreted with a suitable expansion of the junction grouping rules. Junctions of a type unknown to the system could still occur and would simply be ignored. Because of the local nature of junction building, the lack of completeness of the junction dictionary would not hinder image understanding.

An edge-oriented feature detection and scene representation method is primarily useful for objects that are well-described by their edges. On the other hand, objects without any surface crease edges would only be represented by the viewpoint-dependent limb edges and have no associated viewpoint-independent wire-frame. Thus, a generalization of

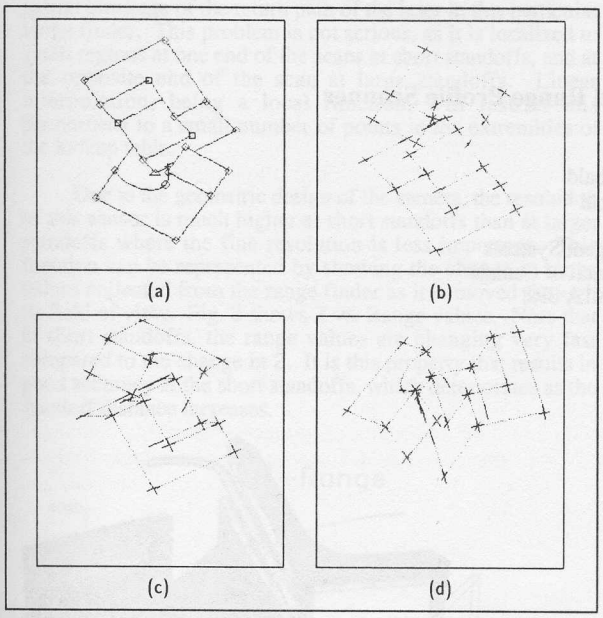

Figure 9 Edge-junction graph. (a) Nodes of the graph representing the junctions: diamonds, squares, and triangle indicate trihedral vertices with 1, 3 or 5 octants; the circle marks a curved C junction; the cross indicates an unknown type of junction. (b) Edges (dotted lines) and tangent vectors at junctions. Projections of the edge-junction graph are shown in (c) and (d).

the description format might include a surface analysis (such as [2, 4, 25]) to provide a richer representation.

The research presented in this paper illustrates the feasability of computing an edge-based description from real range images of curved objects. The high accuracy of the NRCC range sensor makes possible the extraction of a good edge map by using only local operators and shadow analysis. This edge map can then be structured by appealing to certain results from the field of line drawing analysis. The chosen representation of the structured edge map as an edge-junction graph is adequate for curved objects that can be usefully described by their edges, as are many manufactured objects.

Acknowledgements

The authors would like to thank M. Rioux and J. Domey of the National Research Council of Canada for providing the range images used in this research. This research was partially supported by an NSERC Postgraduate Scholarship to G. Godin and the NSERC Operating Grant A1733 to M.D. Levine. M.D. Levine would like to thank the Canadian Institute for Advanced Research for its suppport.

References

[1] S. Aubry, and V. Hayward, " Range Image Analysis Using Level Curves," McRCIM-TR-CIM-87-1, McGill Research Centre for Intelligent Machines, McGill University, February 1987.

[2] P. J. Besl, and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Anal. and Machine Intell.*, vol. 10, no. 2, March 1988, pp. 167-192.

[3] R. C. Bolles, and P. Horaud, "3DPO: a three-dimensional part orientation system," *Inter. J. of Robotics Res.*, vol. 5, no. 3, Fall 1986, pp. 3-26.

[4] P. Boulanger, and P. Cohen, "Stable estimation of a topographic primal sketch," in *Proc. IAPR Workshop on Comput. Vision - Special Hardware and Industrial Applicat.*, Tokyo, Japan, October 12-14, 1988, pp. 436-440.

[5] M. B. Clowes, "On seeing things," *Artif. Intell.*, vol. 2, no. 1, Spring 1971, pp. 79-116.

[6] R. O. Duda, and D. Nitzan, "Low-level processing of registered intensity and range data," in *Proc. of 3rd Inter. Joint Conf. on Pattern Recognition*, Coronado, California, November 8-11, 1976, pp. 598-601.

[7] T.-J. Fan, G. Medioni, and R. Nevatia, "Segmented description of 3-D surfaces," *IEEE J. of Robotics and Automat.*, vol. RA-3, no. 6, December 1987, pp. 527-538.

[8] B. Gil, A. Mitiche, and J. Aggarwal, "Experiments in combining intensity and range edge maps," *Comput. Vision, Graphics, and Image Processing*, vol. 21, no. 3, March 1983, pp. 395-411.

[9] M. Herman, "Generating detailed scene descriptions from range images," in *Proc. of Inter. Conf. on Robotics and Automat.*, St-Louis, Missouri, March 25-31, 1985, pp. 426-431.

[10] D. A. Huffman, "Impossible objects as nonsense sentences" in *Machine Intelligence 6*, edited by B. Meltzer and D. Michie, Edinburgh: Edinburgh University Press, 1971, pp. 295-323.

[11] S. Inokuchi, and R. Nevatia, "Boundary detection in range pictures," in *Proc. of 5th Inter. Conf. on Pattern Recognition*, Miami Beach, Florida, December 1-4, 1980, pp. 1301-1303.

[12] S. Inokuchi, T. Nita, F. Matsuda, and Y. Sakurai, "A three dimensional edge-region operator for range pictures," in *Proc. 6th Inter. Conf. on Pattern Recognition*, Munich, Germany, October 19-22, 1982, pp. 918-920.

[13] T. Kasvand, "Extraction of edges in 3D range images to subpixel accuracy," in *9th Inter. Conf. on Pattern Recognition*, Rome, Italy, November 14-17, 1988, pp. 93-98.

[14] D. Laurendeau, and D. Poussart, "Model building of three-dimensional polyhedral objects using 3D edge information and hemispheric histogram," *IEEE J. of Robotics and Automat.*, vol. RA-3, no. 5, October 1987, pp. 459-470.

[15] J.-G. Leu, I. K. Sethi, and G. Yu, "Range image segmentation from equidistant contours," in *Proc. IEEE Comput. Soc. Workshop on Comput. Vision*, Miami Beach, Florida, November 30-December 2, 1987, pp. 339-341.

[16] M. J. Magee, B. A. Boyter, C.-H. Chien, and J. K. Aggarwal, "Experiments in intensity guided range sensing recognition of three-dimensional objects," *IEEE Transactions Pattern Anal. and Machine Intell.*, vol. PAMI-7, no. 6, November 1985, pp. 629-637.

[17] J. Malik, "Interpreting line drawings of curved objects," *Inter. J. of Comput. Vision*, vol. 1, no. 1, 1987, pp. 73-103.

[18] A. Mitiche, and J. Aggarwal, "Detection of edges using range information," *IEEE Transactions on Pattern Anal. and Machine Intell.*, vol. PAMI-5, no. 2, March 1983, pp. 174-178.

[19] V. Nalwa, "Line-drawing interpretation: a mathematical framework," in *Proc. of IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognition 1988*, Ann Arbor, Michigan, June 5-9, 1988, pp. 18-31.

[20] J. Ponce, and M. Brady, "Toward a surface primal sketch," in *Proc. of IEEE Inter. Conf. on Robotics and Automat.*, St-Louis, Missouri, March 25-28, 1985, pp. 420-425.

[21] M. Rioux, "Laser range finder based on synchronized scanners," *Appl. Optics*, vol. 23, no. 21, November 1984, pp. 3837-3844.

[22] D. R. Smith, and T. Kanade, "Autonomous scene description with range imagery," *Comput. Vision, Graphics, and Image Processing*, vol. 31, no. 3, September 1985, pp. 322-334.

[23] K. Sugihara, "Range-data analysis guided by a junction dictionary," *Artif. Intell.*, vol. 12, no. 1, May 1979, pp. 41-69.

[24] F. Tomita, and T. Kanade, "A 3D vision system: generating and matching shape descriptions in range images," in *Proc. of 2nd Inter. Symp. on Robotics Res.*, Kyoto, Japan, August 20-23, 1984, pp. 35-42.

[25] N. Yokoya, and M. D. Levine, " Range image segmentation based on differential geometry: a hybrid approach," McRCIM-TR-CIM 87-16, McGill Research Center for Intelligent Machines, McGill University, September 1987. *A shortened version will appear in IEEE Transactions on Pattern Anal. and Machine Intell.*.