3D Feature Extraction from Sequences of Range Data

G. Suetz*, G. Sandini**, E. Grazza*** & M. Tistarelli***

* University of Genoa
Department of Communication, Computer and Systems Science
Via Opera Pia 11A - 16145 Genoa, Italy

** Università di Palermo
Dipartimento di Ingegneria Elettrica - Palermo, Italy

In this paper, we illustrate a system designed for extracting relevant superficial features from a sequence of range data. We use depth maps (or, with better results, double maps - see later for further description) because they may constitute a common representation for many different processes, such as depth from motion, depth from scene contours, range finding, and hence facilitate data integration. This kind of information is the starting point for the volumetric integration, which is based on two 3D-voxel-arrays that allow to integrate new range data coming from low level processes and supplies an easy description for surfaces extraction. The 3D integration is based on the concept of merging the density values coming from the maps and it is quite quick.

Unfortunately the results of the 3D integration are not as accurate as we want, and therefore we filter it with a 3x3x3 median operator and with a 2D contours reconstruction (which operates on the surfaces, to improve the efficiency of the algorithm). Poly-Net, the 3D superficial representation, consists of a 3d order graph, whose nodes are polygons patching the objects in the scene, whose edges connect polygons sharing a vertex, an edge and belonging to the same plane. An enhancement of the Marching Cubes algorithm builds up the Poly-Net representation from the volumetric one and then planes are detected from the network by algorithms based on the concept of plane growing. Along this way we can also extract the vertices of the scene through very simple computations.

Some further developments (mainly based on the concept of morphogenesis) are illustrated in the conclusions.

I. Introduction

One of the most challenging topics in artificial vision is that of extracting volumetric information about the observed objects in a scene from a continuous flow of visual data. In humans the task is accomplished using many different sources of information coming from multiple visual processes. This is partially due to the fact that all sensor modalities for range estimation suffer for errors and uncertainties, peculiar to each method. For example, the illumination condition is a weak point in deriving shape from shading [1,2] and the compensation of stereo disparity fails when the matching is performed on edges parallel to the epipolar lines [3,4]. Therefore we decided to select and to integrate, according to a reliability measure, the 3D data coming from different shapes from processes obtaining a unique, volumetric representation of the environment [5,6].

In this report we present an example of the integration of range data computed from stereo matching and optical flow, ending with a 3D representation of the solids in view, in terms of a three-dimensional array of voxels. The experimental set-up is based on two pairs of cameras, with coplanar optical axes directed toward a common fixation point, looking at a set of objects which are translating on a rail, in front of them. An additional camera is located above the rail looking at the top of the objects.

The output of both the stereo and motion algorithms is represented as a dense depth map with an associated reliability map derived from geometric as well as computational considerations [6, 7, 8, 9, 10, 11].

The purpose of the system is to acquire depth information from depth-detection processes (such as stereopsis) and to recognize relevant topological and geometrical superficial features, via a quick 3D reconstruction, for further high level analysis.

The major is to define the high boundary of this analysis. In other words we have to specify what we mean with features: we decided to extract planes, lines and vertices because this analysis is both quite simple and significant for further symbolic operations.

As estimating depth with only one kind of depth-detection processes is not really accurate, we decided to use different modalities of it: the problem is therefore how to integrate efficiently heterogeneous results; we solved it defining a unique output representation for the depth-detection processes (a 3D array of distances), also including data about its own accuracy.

The reconstruction is based only on a sequence of depth maps, without regarding what the origin of each of them is. The main disadvantage of this approach is that it doesn’t take care of the correlation of near points and that each depth map is assumed to have the same accuracy. A more precise description could be obtained using Markov Random Fields [12] to describe the output of depth detection.

1. The paradigm of active vision is used, in the sense that range data is acquired actively interacting with the environment; therefore preexistent knowledge is used explicitly by the algorithms.
2. Acquisition of range data
2.1 Depth from stereo

The stereo algorithm is based on the computation of the cross-correlation between corresponding square patches of the stereo pair; the images over which the cross correlation is performed are obtained by convolving the originals with a Laplacian of Gaussian (LoG) operator and representing only the sign of the filtered images [3,13]. The correspondence between the left and right image points is established using a coarse-to-fine approach, in three successive steps. At each step the correlation is computed at different spatial frequency bands (i.e., filtering the images with TVG masks of different sizes) going from low to high spatial frequencies. At the end, the maximum disparity is obtained for which a one-to-one correspondence between the left and right image points is found, at the maximum resolution scale. The correlation values computed in a region with uniform brightness (i.e., dark patches, which have highly significant edges) or a low correlation and can produce many errors, which is why the confidence correlation is only over the regions of high contrast, which correspond to the image areas whose energy, measured with a TVG operator, is greater. Overlap constraints are placed on the left image energy value, limiting the overall number of patch sizes used to perform the correlation. In practice the patches can be positioned directly on the edges of the image, or they can be positioned on the zero crossings of the wavelet transformed images. The images of Figure 4a and 4b are the first and the second of a sequence of 11 images, so that the middle images were position estimated from the four corner points using a set of corresponding object points. The images were then co-aligned using a linear approximation. The image orientation and scale are computed using the method described in the previous section.

The distance of the moving objects is determined from the optical flow and the known object motion parameters. The distance of the object from the camera is a function of six independent parameters:

\[ Z_0 = Z_0(x, y, \phi, R, L, A) \]  

where \( Z_0 \) is the distance of the world points from the camera along the direction of the optic axis and \( Z_0 \) is the focal length of the camera.

The information acquired at the contour points is not a depth map; in the experiments we compute a dense depth map by interpolating the values at the contour points in a linear manner. In Figure 4d the depth map of the analyzed scene is presented. It was obtained for all image points, by a linear interpolation of the depth values computed at the contour points. The gray level is proportional to depth.

The uncertainty relative to the depth map shown in Figure 4c is presented in Figure 4d; the intensity is proportional to the uncertainty of the measured depth.

3. Uncertainty analysis

In the last paragraph we have addressed the computation of visual motion, aimed at determining depth maps from views of a dynamic scene. The proposed approach is subject to errors due to the finite accuracy of the measured parameters and, particularly, to the computational schema. In order to evaluate, and eventually reduce, the amount of error in the measured parameters (optical flow and depth), a statistical analysis of the computational process has been performed.

The basic idea is that of considering the measurement process as stochastic, where the states, variables are Gaussian with known or measurable variance and covariance corresponding to their actual values. The depth computed from motion parallel can be expressed as a function of known parameters, as stated by (2):

\[ Z_m = Z_a + V_x V_y V_z W X W F \]  

where \( z \) and \( y \) are the coordinates of the considered point on the image plane and \( z \) is the object point on the camera.

Considering all the state variables at Gaussian and uncorrelated, then the mean value of depth is assumed equal to \( Z_x \), while its variance is expressed using a linear approximation [17]:

\[ \sigma^2 = \left( V_x^2 + V_y^2 + V_z^2 \right) \sigma_z^2 \]  

where \( \sigma_x \) is the variance of the depth.

Taking into account these considerations, the variances of the \( x \) and \( y \) components of the flow field are determined in the following way:

\[ \sigma_x^2 = \left( V_x^2 + V_y^2 \right) \sigma_z^2 \]  

\[ \sigma_y^2 = \left( V_x^2 + V_y^2 \right) \sigma_z^2 \]  

where \( \sigma_z \) is the variance of the measured object velocity and \( \sigma_z^2 \) is the variance of the computed object velocity.
the uncertainty of the image-derived parameters. An uncertainty estimate is possible, in this case, only with some precise specifications and assumptions.

An image is a spatially non-stationary random process; we can still use a simple description such as:

\[ R(l, j) \approx [f(l) \ast r(j)] (0 \leq l, j \leq N - 1) \]

Here, \( R(l, j) \) is an high frequency component which has stationary statistics with a mean value equal to zero. In more detail, \( R(l, j) \) is uncorrelated Gaussian variables in the range \((0, 1, 0, \ldots, N - 1)\).

\[ R(l, j) = \exp \left[ -\frac{1}{2} (l^2 + j^2) \right] \]

The correlation function, which depends from disparity \( d \), is modeled in exponential form \([3]\). If a value of disparity \( d \) corresponds to the peak of the correlation function we can write:

\[ R(d) = A \exp [-d/(1 + d^2)] \]

\[ A = \max \{1, \frac{1}{2} a \} \]

\[ a = \text{the maximum intensity value of the image while } b \text{ determines the peak amplitude.} \]

Under these assumptions it is possible to evaluate the disparity variance in relation to the statistical parameters stored in the stereo algorithm. We obtain:

\[ \sigma^2 \approx A \sigma^2 \]

\[ \sigma = \text{the width of the peak of the function, } A \text{ is the maximum intensity value of the images, } a \text{ is the amplitude of the central lobe of the } \gamma \text{ mask used for the filtering of the stereo pair and } \sigma \text{ is the variance of image noise.} \]

4. 3D representation and integration of range data

As the choice of the data representation is crucial in every computation scheme, we give a description of the volumetric integration.

Our best desired representation should be:

- easy for the detection of volumetric information (such as the volume, the surface area, ...);
- with explicit spatial information.
- memory saving.
- fast.

We choose a volumetric form of 3D representation, based on two 3D arrays of voxels: in such a way we make explicit both locality and intrinsic density of volumetric data. As each voxel is an eight bits cell it can describe the state of full, empty, and 99 intermediate density situations, and moreover it can show some particular additional situation (e.g., a point never visited, ...). Nevertheless we could have choosen a larger cell, but this leads to a much greater memory occupancy and, since the low resolution of the depth detection processes, couldn't improve significantly the performance.

While the first cube describes the density values, the cells of the second one contain the estimated variance of the corresponding voxel; this allows to exploit a recurrent analysis, in fact, an approximation of the reliability of the calculated density of each point is always available. This feature is really remarkable for the possible application of the system to navigation tasks of a mobile robot exploring unknown environments.

The 3D reconstruction algorithm acts in the following way:

- for each depth map available do:
- for each pixel visible from the depth map do:
  - calculate the estimated density;
  - if it was never seen before take this value as the new one, else use the following formula, in all adjacent media between the old estimation and the newly computed value:
    \[ M_{i+1} = M_{i+1} + M_{i+1} + \frac{1}{2} (M_{i+1} + M_{i+1}) \]
    \[ M_{i+1} = \frac{1}{2} (M_{i+1} + M_{i+1}) \]
    \[ M_{i+1} = \text{value computed at step} \]
    \[ M_{i+1} = \text{best estimation at step} \]

in other words, simplifying:

\[ \frac{1}{2} M_{i+1} + \frac{1}{2} M_{i+1} = \frac{1}{2} M_{i+1} + \frac{1}{2} M_{i+1} \]

note that this expression coincides with the optimal estimate of constant parameters of the noise,

\[ \text{end; } \]

\[ \text{end. } \]

The linearly approach makes this algorithm simple and quiet, his main advantage is however the opportunity of applying new map, integrating new maps, applying the above procedure to an already built dual cube.

On the other side the results are not too accurate and precise, as before, going on with the superficial analysis, we need to improve the results of the integration: two solution are given: either we find a way to enhance the 3D integration, or we design some filters that clean the volumetric image. While the first idea is conceptually better and improves the knowledge of the scene, it is quite impracticable both because of the limited computational power of today computers and the intrinsic imperfection of the data acquisition systems and finite accuracy of the depth detection processes. Consequently we turned to the second one.

3. Volumetric enhancements

The aim of these processes is to eliminate imperfections from the scene cleaning and regularizing the global shape; the first idea was therefore to use an average operator. Unfortunately this approach, even though generates smooth and easy to see object, corrupts the global shape (also in its smallest dimension, i.e. 3\times3 voxels) and consequently it was rejected. We decided therefore to use the median filter, which in its smallest configuration (i.e. 3\times3 voxels), cleans perfectly the space without removing too much the scene. Also this approach has some drawbacks, the most critical of which is that it smooths the voxels: for this reason we have designed a procedure to repair these imperfections. Also, to extract the iso-surface, we use a marching cubes algorithm, which tries to copy the truth of the surface, according to an anisotropic criterion. The result of the sequential application of these two algorithms is remarkable and it will be illustrated in the experimental part of this work.

4. Surface representation

The first step towards a surface representation of the 3D scene is to define a surface representation model, whose main features should be:

- easy construction from a volumetric representation model; it is particularly important in our case as the raw data is provided by a 3D array of voxels;
- based on the concept of matching the scene, as we want to try to group together patches in planes, to detect planar surfaces;
- preserving explicitly locality information, to allow a fast analysis;
- good common properties, such as memory saving and simple to handle;
- belonging to the same plane;
- is implemented with a double linked list of nodes, if, being a subetation of i, with a predicate over the elements of I, and iii, with a slot in the structure describing polygons.

Our superficial representation scheme, called Poly-net, is a 3d order graph, which:

- nodes: polynomial patches enveloping the scene;
- edges: defining three kind of relationships among nodes:
  1. having at least a vertex in common,
  2. sharing an edge,
  3. belonging to the same plane;
- is implemented with a double linked list of nodes i., if, being a subetation of i, with a predicate over the elements of I, and iii, with a slot in the structure describing polygons.

It is simple to understand how well Poly-net fits our purposes; this is also enforced by the fact that optional features are added, imposing, for example, that all vertices of the scene existing on the same point share the same memory location, so that the topological properties are preserved.

To easily extract a surface representation from a dense voxel representation it is necessary to define a threshold over densities which states what is full and what is empty.

Experiments based on human binocular vision show that the best value is 35%; this is not a truly scientific way to define a threshold, however, we accepted this value because this parameter is not critical for our purposes.

Patching with polygons a scene based on binary voxels is very simple, as we can use the squares of the border cubes, but this leads to an unpleasant representation. Much better results are obtained using the marching cubes algorithm [18], which, given a volumetric, binary representation of a scene, efficiently generates a list of planar triangles smoothly enveloping the scene. We enhanced this algorithm to handle also density values (e.g., not simply binarizing the input image but with a linear interpolation: the output triangles are no more equilateral), and then we integrated this list in Poly-net.

The integration with Poly-net was obviously simple and Moreover we obtained a more regular structure, as the patching polygons are all triangles.

5. Surface segmentation

As concerns scene segmentation, our aim is to design symbolic analyzers with topological and geometrical data coming from the scene. Therefore we decided to detect planes, vertices and edges, both because they are significant for further inspection and also simple to extract. Moreover, choosing this kind of properties, human vision at whatever kind of high level analysis could hope to perform them; they are quite general, in fact.
At first sight, one may try a global approach for the detection of planes, e.g., minimizing a cost function which relates the depth of the voxels inside the objects in the scene (and outside) and the number of planes, for the possible plane set configuration; at closer inspection, however, this way appears unrealistic for its computational complexity.

Our approach is based on the concept of plane growing, which means: starting from a node rotated around the z-axis (i.e., trying to project on it as much other plane as possible), we proceed with a slight move in a z direction, until a cost function is below a priori threshold.

In this section it is worth specifying some concepts:

- how to select the starting node: this is performed through a simple heuristics.
- how the approximation plane varies: as the regression plane approach is not suitable for our purposes (it doesn’t have, in our situation, a simple, recurrent and linear formula), we define, inductively, a very simple plane which produces nice results:

  \[ x_0 = \text{plane of the starting triangle} \]

- the cost function: the maximum distance of the points of the triangles from the approximation plane.

- the threshold: it is defined a priori after an accurate analysis.

Even though this algorithm is not stable, at an a priori analysis it was satisfactory, as it is illustrated in the next section.

8 Experimental results

Three image sequences were acquired from two stereo cameras and another camera looking at a set of objects. Each camera pair was located facing one side of a wall, one stereo pair in front of the wall in the middle, another camera was on a mobile platform mounted on the wall, capable of being moved by the wall in the middle, another camera was on a mobile platform mounted on the wall, capable of being moved by the mobile platform. The motion was a translation of the objects, capturing sequences of 100 frames from above. The displacement of the platform was 5 mm/frame.

The motion algorithm was supposed to the sequence captured from the single camera, building a depth map relative to the 3D image. The stereo algorithm was supposed to the sequence of the stereo cameras, obtaining a total of 4 depth images. In Figure 2a and 2b, two stereo pairs, each on opposite sides with respect to the moving scene, are presented. The computed disparity maps are depicted in Figure 3a and Figure 3b, and the depth maps and the associated variables are shown. In Figure 4 the results of a plane algorithm are presented for the monocular sequence.

5 to 8 are visualizations of the reconstructed objects on an HP 9000 computer with graphic accelerator; the scene is artificially illuminated with red, green and blue lights coming from orthogonal positions, so that connected regions of the same colour belong to the same plane. Figure 5 and 6 present the output of the 3D integration process with the threshold set to 85% and 95% respectively; note that, while the threshold grows up, the noise and artefacts decrease. Figure 6 shows the results of the application of the 33x33 median filter to the 3D array of objects and Figure 7 the output of the contour reconstruction. The objects extracted from the scene are shown in Figure 8, the threshold used is 2.4 voxels.

9. Conclusions

The integration of different visual processes has been investigated. We have developed a system for the 3D reconstruction of objects ending with a volumetric representation, which is used to extract significant 3D features. The integration process accumulates several depth maps provided from ranging sensor modalities like depth-from-stereo and depth-from-motion; depth maps are merged according to their uncertainty. Surfaces are extracted from the volumetric representation as a graph of polygons; then planes are selected using a plane-growing technique. Other significant features which can be handled by the surface description are vertices and edges of the solids. At present we are testing the algorithms for the extraction of vertices and edges. They are based on the fact that, after the projection of the index on the planes, the network maintains some initial topological information. Therefore vertices can be detected at polygon vertices belonging to more planes, while edges are extracted analyzing planes, vertices and the graph.

The Morphostructural Analysis [10] deals with finding a parameter function which, from a topological entity (the object) describes the scene of the entities. This approach assumes that the morphogenesis carries more information than the mere object. Given a scene, on the approximated morphogenesis is obtained filtering it with media operators at different resolution scales. We are now further using the systems, trying to face the unsatisfactory problem of an analysis of the approximated morphogenesis, which is useful until now, i.e., in any case, for high level recognition.

We are also developing algorithms for the visualization of geometrical features (edges or color) in the 3D structure. Extending this information to the graph is really simple (just one more field in the structure) and may be useful for two purposes: to improve the plane detection algorithm (e.g., via a more elements in the threshold (the pictorial data), and for higher level symbolic reasoner.

References


Acknowledgements: The research described in this paper was carried out by the ESPRIT project PAIR: Image and Movement Understanding and by the National Research Council.


Figure 2. Two stereo pairs used in the experiment. The resolution of the images is 256x256 pixels.

Figure 3. Results of the stereo algorithm relative to the stereo views of Figure 2. a) Results of the regional correlation step; the grey level codes image disparity; b) depth maps obtained by linear interpolation of the contour values (depth is proportional to grey levels); c) the associated uncertainty maps (uncertainty is proportional to grey levels).
Figure 4: a) First and last image of the monocular sequence of 11 images used in the motion experiments. The object was moving along a direction parallel to the X (horizontal) axis of the camera coordinate system. The resolution of the images is 256 x 256 pixels. b) Optical flow; c) depth map obtained by linear interpolation of the depth values computed from the optical flow (depth is proportional to gray level); d) the associated uncertainty (consistency) is proportional to gray level.

Figure 5: 3D volumetric integration of the depth maps. Perspective representation of the voxel-based accumulator obtained integrating both stereo and motion-derived information. a) perspective view of the solids with probability equal to 0.8; b) some of a) with probability equal to 0.5.

Figure 6: Result of the application of the 7x7x3 median filter.

Figure 7: Application of the surface representation.