

## Method for Reconstructing Functionally Defined Surfaces by Stereo Images of Real Objects

S. I. Vyatkin<sup>1\*</sup> and B. S. Dolgovesov<sup>1\*\*</sup>

<sup>1</sup>*Institute of Automation and Electrometry, Siberian Branch, Russian Academy of Sciences,  
Novosibirsk, 630090 Russia*

Received December 12, 2019; revised September 25, 2020; accepted October 12, 2020

**Abstract**—A method for reconstructing functionally defined surfaces by stereo images of real objects is proposed. A known approach to the computation of 3D models by silhouette images in the object space is used for preprocessing. The depth map is calculated using a well-known algorithm based on area with correlation of image intensity levels. In order to reconstruct a smooth surface with a compact description, a new method of transformation into a functionally defined model is proposed.

**DOI:** 10.3103/S875669902006014X

**Keywords:** *3D object reconstruction, silhouette, stereo pair, depth map, octree, perturbation functions*

### INTRODUCTION

Computing 3D and range data is important for various applications: computer graphics, medicine, multimedia, machine vision, navigation, automotive safety, computer interaction, tracking, image recognition, and many others. Obtaining 3D geometric data from a complex environment has been a subject of research for many years. Computer graphics and image analysis are two methods for processing visual information. Computer graphics uses formal descriptions of objects to create their visual images. Image analysis systems use the images to create formalized models of objects. Recently, there has been a tendency to mutual integration of computer graphics and image analysis. This is due primarily to the development of virtual reality systems, which use 3D modeling of real objects, which is inefficient for complex 3D objects that require significant time and computing resources.

There are active and passive methods for reconstructing information about the depth of the real scene. Active methods, which output depth information, use laser illumination of the working space [1]. However, these methods have limitations with respect to measurement range and cost of hardware components.

Passive methods based on computer vision are implemented with simpler and more inexpensive distance sensors. Such methods can generate information about the depth by an obtained pair of images and parameters of two cameras [2].

Modeling the shape of real-world objects from a series of images has been investigated in recent years. One well-known approach to 3D modeling is shape generation from a silhouette, which reconstructs the shape of objects from their contours. This approach is popular because of its fast computation and high reliability. The construction of 3D models by multiple views was presented in [3]. The method [4] uses orthogonal projection to build 3D models. Numerous studies are devoted to creating shapes from silhouettes to convert visible contours into a visual form [5–7].

---

\*E-mail: [sivser@mail.ru](mailto:sivser@mail.ru)

\*\*E-mail: [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

The aim of this work is to develop a method for reconstructing a 3D model of a real object by means of silhouettes and multiview depth maps, which are generated by means of a stereo pair with the subsequent transformation into a functional model.

In this article, a well-known method for creating shape from silhouettes [8] and a correlation algorithm [9] to calculate a depth map are used.

### PERTURBATION FUNCTIONS

It is proposed to describe complex geometric objects by defining a perturbation function (second order) from a base quadric, which has the form of the following inequality

$$F(x, y, z) = Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + K \geq 0. \quad (1)$$

Free forms are constructed on the basis of quadrics. A free form is a composition of the base quadric  $F(x, y, z)$  and perturbations [10]

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z), \quad (2)$$

where  $R(x, y, z)$  is the perturbation, and  $f_i$  is the form factor:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{if } Q_i(x, y, z) \geq 0; \\ 0, & \text{if } Q_i(x, y, z) < 0, \end{cases} \quad (3)$$

where  $Q(x, y, z)$  is the perturbing quadric.

The resulting surface will be smooth and will require a small number of perturbation functions to create complex surface shapes; the amount of data is reduced by more than 100 times compared to a polygonal description [11].

### OBJECT RECONSTRUCTION

Calibration of stereo cameras with calculation of such parameters as focal lengths, mutual position of cameras in space, optical curvature parameters, which all video cameras have, brightness distortion parameters, and others will not be considered in this work.

The method consists of three calculation steps:

- (1) Computation of silhouettes, voxel model, and intersected nodes of the octree from root to leaves.
- (2) Calculation of multiview depth maps, point clouds, merging of point clouds, construction of a triangle grid, and computation of normals.
- (3) Transformation into a functional model.

In order to solve the problem of reconstruction of objects, the well-known method of 3D reconstruction using voxels [8], stereo reconstruction with the determination of depth and surface normals [12, 9], and transformation into a functionally defined description of the octree structure and calculated normals were used.

1. In the 3D reconstruction method using voxels, a rotating disk is rotated, video images are captured, and the contour of a real object is extracted (Fig. 1).

When the camera geometry is known, a bounding pyramid can be constructed for each image. This is done by sighting lines from the camera focus through all the contour points of the object silhouette. The shape is computed in 3D by a cut-out method to remove all voxels that are outside the projected silhouette cone. The intersection of all silhouette cones from several images defines the geometry of the object, which is called the visual shape [8].

If the image coordinate defines a background pixel, the voxel is marked as to be removed. After processing all voxels, the object is rotated and the algorithm goes to the next image.

The method is optimized based on voxel calculations using an octree structure. Voxels at the current level can be obtained from the results of the previous level by applying a modified algorithm. Constraints



Fig. 1. Real object with shadows in different angles.

on the areas containing the corresponding information object significantly shorten the computation time. The use of such preprocessing greatly reduces the probability of depth calculation error.

Calculating normals to voxel surfaces is a nontrivial task. The scalar field of a voxel model is considered in its current state, i.e., its time dependence is not taken into account. The gradient of the scalar field  $\nabla f(x)$  is always perpendicular to the level surfaces, the set of points in space at which the scalar field takes the same value. Therefore, the normal to the voxel surface at a given point  $\mathbf{n}(\mathbf{x})$  can be calculated as a scalar field gradient, which is a vector of partial derivatives of the scalar field along the directions of the coordinate axes. The partial derivative  $\partial f(\mathbf{x})/\partial x$  of the considered scalar field toward the  $i$ th axis of the coordinate system can be found as a convolution of the discrete function  $f[\mathbf{k}]$  with a filter kernel of a special form. The sequence of operations is performed in one pass for one block of the voxel model. In order to obtain an image of the whole voxel model or its separate part from the camera, the procedure should be repeated for each block in the sample. The operation speed depends on the number and size of output blocks, pixel projection area, and camera orientation relative to the 3D grid (the number of voxel samples increases if projection beams pass through the blocks diagonally). The disadvantages of the method include noninteger processing of blocks. As a result, normals are calculated for block boundaries, while the integrity of the normal map of the monolithic body is not observed.

Therefore, in this work another method was used to calculate normals.

2. In addition to voxel shape generation, the method of reconstruction of the 3D scene from an image sequence is used for depth map computation.

For the calibrated stereo pair, we have two projective matrices [12]:

$$\begin{pmatrix} u_i s_i \\ v_i s_i \\ s_i \end{pmatrix} = M_i \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (4)$$

where  $X$ ,  $Y$ , and  $Z$  are 3D coordinates of the point,  $u_i$  and  $v_i$  are their projections on the image and  $i$ ;  $s_i$ , and is the scale factor. The stereo pair  $I = (u, v)$  is the point of the image plane and  $P = (x, y, z)$  is the point of the world coordinate system. From the calibrated stereo pair, the depth map of one perspective of an image is calculated using the correlation algorithm. In this work, the area-based algorithm with correlation of image intensity levels (grey levels) is used [9].

The number of perspectives is generated according to the number of silhouettes calculated earlier. Depth maps are projected back into the 3D space of the object. When reconstructing a 3D shape from multiview silhouettes and depth maps, a cloud of 3D points is first generated from each depth image with an appropriate camera setup. The combination of these point clouds is treated as an initial shape estimate. It is then refined by applying the previously calculated silhouettes to filter out the noise points.

The point will be preserved only if all its multiview 2D-projections are inside the silhouettes. Then, a grid of triangles is constructed and the normals are calculated as described in [13].

3. The final shape based on perturbation functions is highly dependent on the accuracy of the normals, because the computation begins by traversing the leaves of the octree where they are stored. After calculating the voxel model and normals, all intersected nodes of the octree are marked at each point on the surface from the root to the leaves to create a functionally specified object (the leaves store the calculated normals).

The surface voxels (leaves of the octree) are matched with the image input signal. Functional objects consist of base quadrics and perturbation functions.

In order to distinguish between the base quadric and the perturbation functions, the coefficients of its equation are preliminary determined. To do this, the quadric should be written into the resulting structure of the octree model, and the coefficients of the base quadric should be determined by the 9 dimensional points of the surface voxels. We have the coefficients of perturbation function equations (1)  $A, B, C, D, E, F, G, H, I$ , and  $K$  and coefficients of equations of the base quadric  $q$ . At the next defining of the quadrics,

$$Q = \begin{pmatrix} q_{xx} & q_{xy}/2 & q_{xz}/2 & q_x/2 \\ q_{xy}/2 & q_{yy} & q_{yz}/2 & q_y/2 \\ q_{xz}/2 & q_{yz}/2 & q_{zz} & q_z/2 \\ q_x/2 & q_y/2 & q_z/2 & q \end{pmatrix}. \quad (5)$$

The value of the function given by (5) at an arbitrary point  $P[x, y, z]$  will be written as follows

$$Q(P[x, y, z]) = q_{xx}x^2 + q_{yy}y^2 + q_{zz}z^2 + q_{xy}xy + q_{xz}xz + q_{yz}yz + q_x x + q_y y + q_z z + q. \quad (6)$$

Since the value of  $Q$  on the surface is zero, we have a system of linear equations to find the quadric coefficients by the 9 points  $P_1[x_1, y_1, z_1] - P_9[x_9, y_9, z_9]$

$$\left\{ \begin{array}{l} Q(P_1) = 0 \\ Q(P_2) = 0 \\ \vdots \\ Q(P_i) = 0 \\ \vdots \\ Q(P_9) = 0 \end{array} \right\}. \quad (7)$$

Substituting (6), we obtain

$$\left\{ \begin{array}{l} q_{xx}x_1^2 + q_{yy}y_1^2 + q_{zz}z_1^2 + q_{xy}x_1y_1 + q_{xz}x_1z_1 + q_{yz}y_1z_1 + q_x x_1 + q_y y_1 + q_z z_1 + q = 0; \\ q_{xx}x_2^2 + q_{yy}y_2^2 + q_{zz}z_2^2 + q_{xy}x_2y_2 + q_{xz}x_2z_2 + q_{yz}y_2z_2 + q_x x_2 + q_y y_2 + q_z z_2 + q = 0; \\ \vdots \\ q_{xx}x_i^2 + q_{yy}y_i^2 + q_{zz}z_i^2 + q_{xy}x_iy_i + q_{xz}x_i z_i + q_{yz}y_i z_i + q_x x_i + q_y y_i + q_z z_i + q = 0; \\ \vdots \\ q_{xx}x_9^2 + q_{yy}y_9^2 + q_{zz}z_9^2 + q_{xy}x_9y_9 + q_{xz}x_9z_9 + q_{yz}y_9z_9 + q_x x_9 + q_y y_9 + q_z z_9 + q = 0. \end{array} \right\}. \quad (8)$$

The solution of the system of equations with the free term  $q$  (let us put it equal to  $K$ ) and given  $P_1[x_1, y_1, z_1] - P_9[x_9, y_9, z_9]$  will be 9 required coefficients:  $q_{xx}, q_{yy}, q_{zz}, q_{xy}, q_{xz}, q_{yz}, q_x, q_y, q_z$ .

Next, the coefficients of this quadric will be used when transforming to a functional model and calculated using the Cramer's method:

$$\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 \\ x_3^2 & y_3^2 & z_3^2 & x_3y_3 & x_3z_3 & y_3z_3 & x_3 & y_3 & z_3 \\ x_4^2 & y_4^2 & z_4^2 & x_4y_4 & x_4z_4 & y_4z_4 & x_4 & y_4 & z_4 \\ x_5^2 & y_5^2 & z_5^2 & x_5y_5 & x_5z_5 & y_5z_5 & x_5 & y_5 & z_5 \\ x_6^2 & y_6^2 & z_6^2 & x_6y_6 & x_6z_6 & y_6z_6 & x_6 & y_6 & z_6 \\ x_7^2 & y_7^2 & z_7^2 & x_7y_7 & x_7z_7 & y_7z_7 & x_7 & y_7 & z_7 \\ x_8^2 & y_8^2 & z_8^2 & x_8y_8 & x_8z_8 & y_8z_8 & x_8 & y_8 & z_8 \\ x_9^2 & y_9^2 & z_9^2 & x_9y_9 & x_9z_9 & y_9z_9 & x_9 & y_9 & z_9 \end{bmatrix} \begin{bmatrix} q_{xx} \\ q_{yy} \\ q_{zz} \\ q_{xy} \\ q_{xz} \\ q_{yz} \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \end{bmatrix} . \tag{9}$$

Functionally, during rasterization the model is set in a cube with center (0, 0, 0) and coordinates from -1 to 1 along *x*, *y*, and *z* in the object coordinate system. In octal division, the coefficients of the equation *A*, *B*, *C*, *D*, *E*, and *F* are divided by 4 and the coefficients *XYZ*, by 2 (reducing the size of the cube). After that, a shift to vector (±0.5; ±0.5; ±0.5) (10) is made, where unprimed coefficients are taken from the previous recursion step:

$$\begin{aligned} A' &= A/4, & B' &= B/4, & C' &= C/4, & D' &= D/4, & E' &= E/4, & F' &= F/4, \\ G' &= G/2 + iA/2 + jD/4 + kE/4, & H' &= H/2 + iD/4 + jB/2 + kF/4, \\ I' &= I/2 + iE/4 + jF/4 + kC/2, & K' &= K/2 + iG/4 + jH/4 + kI/4, \\ & & K'' &= K'/2 + iG'/2 + jH'/2 + kI'/2. \end{aligned} \tag{10}$$

All surface points (surface voxels) are known and all branches and leaves of the tree where the model intersected with cubes of different levels of the octree representation of the object space took place as well as all intersected leaves of the tree. The main idea is as follows. If by means of recursive partitioning of the object space, we can visualize functionally defined objects (calculate surface points, normals at these points, illumination, etc.), then we can solve the following inverse problem:

$$N_x = \frac{G'}{\sqrt{G'^2 + H'^2 + I'^2}}; \tag{11}$$

$$N_y = \frac{H'}{\sqrt{G'^2 + H'^2 + I'^2}}; \tag{12}$$



Fig. 2. Virtual functionally defined object in different perspectives.

**Table 1.** Computation time reduction using hierarchical processing

CPU E8400	Levels	CPU E8400	Levels
52	1	2.93	5
9.35	2	2.88	6
4.06	3	2.8	7
3.01	4	2.66	8

Calculation time per minute.

$$N_z = \frac{I'}{\sqrt{G'^2 + H'^2 + I'^2}}, \quad (13)$$

i.e., find functions describing a given object by the given points and normals (11), (12), and (13).

Systems of linear equations at each level of recursion are solved. The same coefficients of equations are determined (taking into account the set threshold of approximation accuracy), and the number of functions at each level is minimized. Only after all equations of a given level have been processed does the process move to the next upper level. This process is repeated up to the octree root and as a result obtains the required minimum of functions representing the given object in the format of description of functionally defined objects on the basis of quadrics with analytical perturbation functions (Fig. 2).

## METHOD ANALYSIS AND RESULTS

A voxel model of 3D shape reconstruction using 30 images and  $512^3$  voxels was obtained from a real object. It does not have a good degree of smoothness to look realistic. Therefore, a stereo pair reconstruction method was used to calculate the normals more accurately. For the sake of brevity of description and smoothness of the surface, the model was transformed into a functional description. It required 104 perturbation functions. Hierarchical processing using the octree structure significantly shortens the computation time (Table 1).

The compression coefficient for transforming polygonal models into functional ones is more than two orders of magnitude, and for voxel models, more than three orders of magnitude. The method was tested for several real objects. This method can be used to transform polygonal and voxel models into functional objects without losses. Moreover, for smooth curvilinear surfaces, the image quality will improve. The objects are visualized using the method described in [14] and using graphics processors. It took from several milliseconds to one second to transform voxel models into functional ones. The testing was performed on an Intel Core2 CPU E8400 3.0 GHz. The known approaches to transform 3D models into implicit surfaces use iterative methods, which are approximate and slow. In the proposed method, the transformation to a functionally based model is carried out by a rigorous mathematical calculation without iterations.

## CONCLUSIONS

This article proposed a method for reconstructing real objects from stereo images using the voxel structure of an octree and its transformation into a functional description based on perturbation functions. The transformation is performed by mathematical calculation rather than iteratively and with approximations which lead to a partial loss of information. Hierarchical processing significantly reduces the computation time of the voxel model. The proposed method of defining 3D objects and the method of geometric data transformation have advantages over known approaches.

The main advantages of the proposed method and technique include: the simplicity of transformation of voxel objects into a functional description with a fast search for describing functions as well as a significant reduction in the amount of data for defining curvilinear objects.

## REFERENCES

1. J. Butime, I. Gutierrez, L. G. Corzo, and C. F. Espronceda, "3D Reconstruction Methods, a Survey," in *Proc. First Int. Conf. on Computer Vision Theory and Applications, Setúbal, Portugal, 2006*, vol. 2, pp. 457–463.  
<https://doi.org/10.5220/0001369704570463>.  
<https://pdfs.semanticscholar.org/1a63/9c74c0327c7982714b3f964fd09417ec4e9f.pdf>. Cited December 12, 2019
2. M. Siudak and P. Rokita, "A Survey of Passive 3D Reconstruction Methods on the Basis of More than One Image," *Machine Graph. Vision* **23** (3/4), 57–117 (2014).  
[http://mgv.wzim.sggw.pl/MGV23\\_3-4\\_057-117.pdf](http://mgv.wzim.sggw.pl/MGV23_3-4_057-117.pdf). Cited December 12, 2019.
3. W. N. Martin and J. K. Aggarwal, "Volumetric descriptions of objects from multiple views," *IEEE Trans. Pattern Analysis Machine Intell.* **PAMI-5**, 150–158 (1983). <https://doi.org/10.1109/TPAMI.1983.4767367>
4. C. H. Chien and J. K. Aggarwal, "Identification of 3D objects from multiple silhouettes using quadrees octrees," *Comput. Vision Graph. Image Process.* **36**, 256–273 (1986).  
[https://doi.org/10.1016/0734-189X\(86\)90078-2](https://doi.org/10.1016/0734-189X(86)90078-2)
5. W. Niem, "Robust and fast modeling of 3D natural objects from multiple views," *SPIE Proc.* **2182**, 388–397.  
<https://doi.org/10.1117/12.171088>
6. R. Szeliski, "Shape from rotation," in *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition, Maui, USA, 1991*, pp. 625–630. <https://doi.org/10.1109/CVPR.1991.139764>
7. S. Vedula, P. Rander, H. Saito, and T. Kanade, "Modeling, combining, and rendering dynamic real-world events from image sequences," in *Proc. 4th Conf. on Virtual Systems and Multimedia (VSMM98), Gifu, Japan, 1998*, pp. 326–332.
8. Ya. Kuzu and V. Rodehorst, Volumetric modeling using shape from silhouette.  
[https://www.cv.tu-berlin.de/fileadmin/fg140/Volumetric\\_Modeling\\_using.pdf](https://www.cv.tu-berlin.de/fileadmin/fg140/Volumetric_Modeling_using.pdf). Cited December 12, 2019.
9. S. I. Vyatkin, "Complex surface modeling using perturbation functions," *Optoelectron., Instrum. Data Process.* **43**, 226–231 (2007).
10. S. I. Vyatkin and B. S. Dolgovesov, "Compression of geometric data with the use of perturbation functions," *Optoelectron., Instrum. Data Process.* **54**, 334–339 (2018). <https://doi.org/10.3103/S8756699018040039>
11. S. I. Vyatkin, "Method of face recognition with the use of scalar perturbation functions and a set-theoretic operation of subtraction," *Optoelectron., Instrum. Data Process.* **52**, 43–48 (2016).  
<https://doi.org/10.3103/S8756699016010076>
12. P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," *Machine Vision Appl.* **6**, 35–49 (1993). <https://doi.org/10.1007/BF01212430>
13. S. I. Vyatkin, "Polygonization method for functionally defined objects," *Int. J. Autom., Control Intell. Syst.* **1**, 1–8 (2015).
14. S. I. Vyatkin, "Recursive search method for the image elements of functionally defined surfaces," *Optoelectron., Instrum. Data Process.* **53**, 245–249 (2017). <https://doi.org/10.3103/S8756699017030074>

*Translated by O. Pismenov*