Image-based 3D Reconstruction of Generalized Box with User Sketches

Anton Yakubenko, Anton Konouchine, Vladimir Vezhnevets
Moscow State University, Graphics & Media Lab
Moscow, Russia
toh_yakubenko@mail.ru, {ktosh, vvp}@graphics.cs.msu.ru

Abstract

3d modeling is one of the key problems in computer graphics. Creating 3d models of real world objects is among tasks of 3d modeling problem. One of the approaches to solving it is using images of the object of interest. Images are the most natural and easy to obtain source of information about real-world 3d object. So lots of research is focused on image-based 3d modeling.

We propose an algorithm for reconstructing 3d model of object with a specific shape from a sequence of calibrated images with user interaction. The shape of the object is called “generalized box” and describes objects with hexahedron shape, which edges are polylines or curves and faces are curved surfaces. The class of objects described as generalized box is rather wide. It includes technical equipment, machinery or their parts, architectural buildings, a host of common everyday objects.

User outlines some edges on several images with known camera calibration. First 3d edges, corresponding to their outlined projections, are reconstructed. Then missing 3d edges are obtained from known 3d edges. At last curved surface is constructed for each face of the model. After 3d model is constructed, the texture is extracted from input images.

Keywords: Image-based modeling, Sketch, Sketch-Based, 3D Reconstruction*.*

1. INTRODUCTION

Creating of 3d models is a wide and popular problem. Modeling of real world objects is a specific task. Considerable talent, experience and much time are required for creating of high quality 3d models looking like real objects. Images contain much information about real-world 3d objects and are easy to obtain. Using images as the main source of information about object reduces modeling time and allows creating of high quality real looking 3d models.

Fully automatic approaches are not robust and usually do not produce good results. That is why user input is used as additional source of information. Most approaches offer a small choice of very simple primitives [7], full modeling process is routine and takes much time.

We propose an approach for reconstructing of objects with generalized box shape from a sequence of images with just several input user strokes. Technical equipment, machinery or their parts, architectural buildings, a host of common everyday objects are representatives of such objects, etc.

First common problems and existing approaches of image-based modeling will be outline. Then the proposed method will be described and achieved results will be shown. Conclusion and future work end the article.

2. BACKGROUND

To use images for 3d modeling we need to know a relation between 3d scene and 2d image, which is described by projective transformation. Process of estimation parameters of projective transformation is called camera calibration. It is a stand alone problem in computer vision and a bunch of various methods for its solving are already presented. Fully manual matching of corresponding points in several images produces good results but takes too much accurate actions from user. Automatic calibration of image sequence or video requires well textured scene and still is not robust. Calibration technique based on templates [2] assumes that a specific calibration object with known features, presents on each image. We used [10] for pattern-based calibration. If camera calibration is obtained, shape of the object might be reconstructed, 3d model might be built and rendered. Several main approaches for shape reconstruction from a number of calibrated images were already proposed.

One of them is based on visual hull of the object [3]. This approach bears on voxel representation. Voxel is a 3d point in discrete 3d space. If the silhouettes of the object in the image is segmented, visual hull of the object corresponding to this image contains all 3d points, which projections on all images coincides with the original silhouettes. For high quality results views from all directions around the object of interest are often required. The quality of the final model also strongly depends on the accuracy of calibration. This approach usually works well for not very complex and not very concave objects with smooth surface without small details.

Another approach is based on voxel coloring [4]. If object surface is not specular, then each point of the object is observed from different direction in same color. With projecting all points of voxel space onto images and checking colors of the projections a color voxel model of the object might be constructed. This approach is very exacting to time, camera calibration accuracy and images quality. All these requirements are hard to maintain, thus this approach is not widely used.

Among other approaches there is obtaining 3d sparse points and fitting primitives to them [5]. Points correspond to some peculiar features of the object or scene that means that object or scene should have textures with a lot of particular features. The quantity of points for good results should be rather great. Manual matching of points is a routine task, which takes too much time. Automatic points estimation is still not very robust.

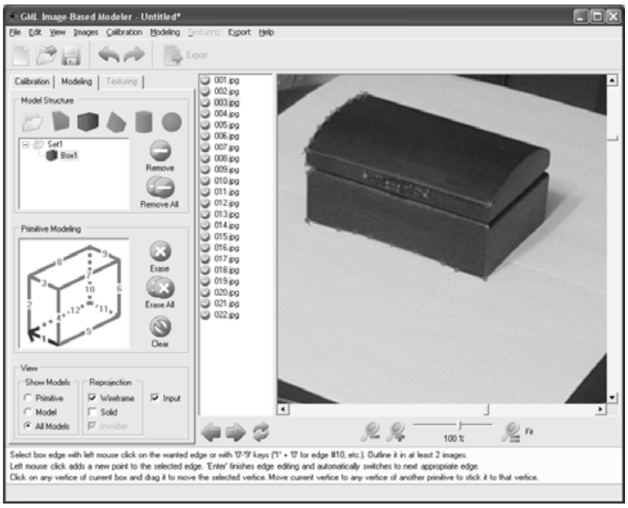
One of the approaches is based on dense matching of all points in image and constructing 3d surface [1].

The only 3d modeling approach that reaches maturity to be implemented in commercial software is manual or automated matching of primitives on images [11].

3. PROPOSED METHOD

3.1 User Interface

We have developed a 3d modeling application, which guides the user through the modeling process.


GML Image-Based Modeler - Untitled*
File Edit View Images Calibration Modeling Texturing Export Help
Export
Calibration Modeling Texturing
Model Structure
Set1
Box1
Remove
Remove All
Primitive Modeling
1 2 3 4 5 6 7 8 9 10 11 12
Erase
Erase All
Clear
View
001.jpg 002.jpg 003.jpg 004.jpg 005.jpg 006.jpg 007.jpg 008.jpg 009.jpg 010.jpg 011.jpg 012.jpg 013.jpg 014.jpg 015.jpg 016.jpg 017.jpg 018.jpg 019.jpg 020.jpg 021.jpg 022.jpg
100 %
Fit

Show Models: ○ Primitive ○ Model ● All Models

Reprojection: ☑ Wireframe □ Solid ☑ Invisible

☑ Input

Select box edge with left mouse click on the wanted edge or with '0'-'9' keys ('1' + '0' for edge #10, etc.). Outline it in at least 2 images.
Left mouse click adds a new point to the selected edge. 'Enter' finishes edge editing and automatically switches to next appropriate edge.
Click on any vertice of current box and drag it to move the selected vertice. Move current vertice to any vertice of another primitive to stick it to that vertice.

Figure 1: Graphical user interface of our application.

The whole modeling framework consists of 3 steps: camera calibration, 3d modeling with user sketches, and texturing. Camera calibration data can be imported or computed by producing automatic pattern-based calibration [10].

In modeling step user should outline several edges of the generalized box. All twelve edges are numbered, thus user should select edge number first. It is done by clicking at the corresponding edge of the help picture.

Each edge is represented as polyline. All editing functions for polylines are provided. A smart auto selection of next edge for editing is developed. It is based on typical order of drawing edges on the current image. It tries to select next edge in the same face as the last edited edge. In case of collision the edge is selected in the face with the largest number of already outlined edges. User should outline at least three edges on at least two images. 3d model is interactively constructed if possible.

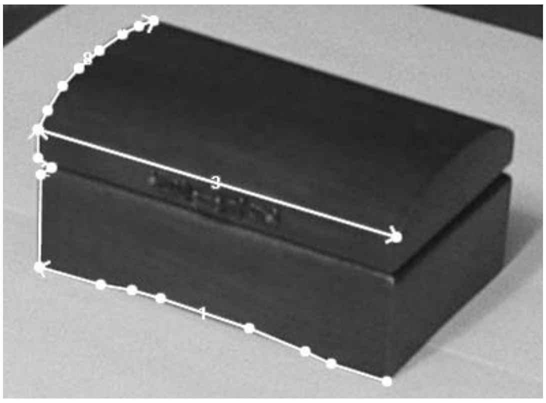


Figure 2: Marked edges of the box - note the curved shape.

3.2 Algorithms

3.2.1 3d Edge Reconstruction

First step of reconstruction step is building a curved 3d edge, corresponding to input 2d edges. Usually when it is needed to construct a 3d edge from two or more its projections, it is also necessary to know correspondence between all the points of the 2d projections. Or at least correspondence between some points should be known, and intermediate points are interpolated. These points are triangulated [6] and 3d polyline is obtained. In our case we know exact correspondence only for end points of the edge.

One way to reconstruct the 3d edge from 2d projections of that edge is to intersect the visual hulls of 2d projections [3]. But due to inaccuracy of calibration the intersection of the visual hulls might be wrong. Also intersection of visual hulls of polylines might work only for two images, because the intersection of 3d curve, obtained from two 2d projections, with 3d surfaces corresponding to other 2d projection usually produces a small set of sparse 3d points. Visual hull is not a good choice for complex polylines or small differences between cameras positions.

Another approach [7] is based on epipolar lines. For instance, let's consider the case of two images. Let a be a point in the first 2d polyline, F is fundamental matrix, then $b = Fa$ is epipolar line. Let c be a point of intersection of b and second polyline, then point c is considered to correspond to point a and the triangulation of these points is a 3d point of the resulting 3d polyline. But there can be more than a single intersection, or even there can be none intersection. The problem of choosing the right corresponding point is ambiguous. The case of none intersection is also a special situation. Moreover some peculiarities of the second polyline might be missed. Also this approach in current formulation is not suitable for more than two images.

We propose a new approach for estimating the correspondence between points of several 2d polylines or curves and constructing the according 3d edge. For instance, let's consider the case of three input polylines.

1. Splitting all input 2d polylines into small segments. As the result we obtain 2d polylines with a large number of points.

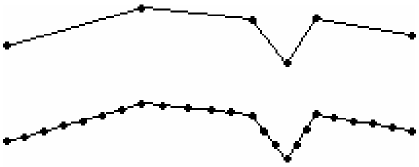


Figure 3: Input polyline (top) and splitted polyline (bottom).

Let's consider that after splitting first polyline has n_1 points with indices $1..n_1$, second polyline has n_2 points with indices $1..n_2$ and third polyline has n_3 points with indices $1..n_3$.

2. The correspondence between points of polylines can be described as a 3d vector of indices. For instance, vector $\{k_1, k_2, k_3\}$ means that some 3d point of the 3d edge can be obtained with triangulation of 3 points in images: point with index k_1 from the first polyline, point with index k_2 from the second polyline and point with index k_3 from the third polyline.

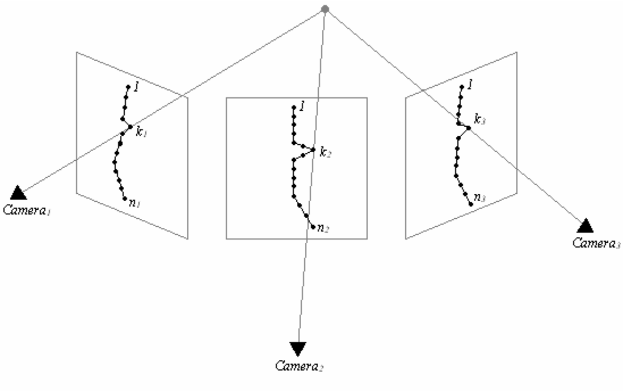


Figure 4: Points correspondence.

We used minimization of reprojection error with gradient descent algorithm for triangulation [6].

3. Starting from the first end point. The correspondence of points for the first end point is known - *{1, 1, 1}*. Thus $k_1=1$, $k_2=1$, k_3=1. The according triangulated point is the first point of the resulting 3d edge.

4. Triangulating points corresponding to vectors *{k_1, k_2, k_3+1}, {k_1, k_2+1, k_3}, {k_1, k_2+1, k_3+1}, {k_1+1, k_2, k_3}, {k_1+1, k_2, k_3+1}, {k_1+1, k_2+1, k_3}, {k_1+1, k_2+1, k_3+1}* – all combinations of points with indices exceeding respective indices of current point no more than *1*. For each triangulated point reprojection error is computed. The point with minimum reprojection error is added to the resulting 3d edge and becomes current.

5. Repeating step 4 until last point is reached – *{n_1, n_2, n_3}*.

6. The resulting 3d polyline would have a great number of points. To reduce it we apply the following post-processing:

6.1. Only points which correspond to any point of the original polyline before splitting are left. During step 1 the indices of original points in the splitted polyline are saved. If any of index in the 3d vector corresponding to the point of the resulting 3d polyline exists in the according set of original indices, then this point is left, otherwise it is removed from the polyline.

6.2. If the array of 3d vectors corresponding to the polyline acquired on the previous step contains more than one vector with the same index from the set of the original indices, for instance, {k_1, k_2, k_3} and {k_1, k_4, k_5} and k_1 is in the set of original indices, then only one of the 3d points corresponding to these vectors is left. The first criteria is the number of indices from the set of original indices in these vectors. For instance, if k_2, k_3, and k_4 do not exist in the sets of original indices and k_5 appears in the set of original indices corresponding to the third polyline, then the point corresponding to {k_1, k_2, k_3} would be removed and the point corresponding to {k_1, k_4, k_5} would be left. If all the candidate vectors have the same number of original indices, then the point with the minimum reprojection error is left and others are removed from the resulting 3d polyline.

The advantages of the proposed method of reconstructing of 3d polyline from a number of 2d projections are:

- Needs only correspondence for end points of the 2d polylines.

- Does not depend on the complexity of the input polyline.

- Is suitable for any number of images.

- Works well on not very precise calibration.

3.2.2 Missing Edges Construction

On the previous step we have obtained 3d edges, corresponding to their 2d projections on images.

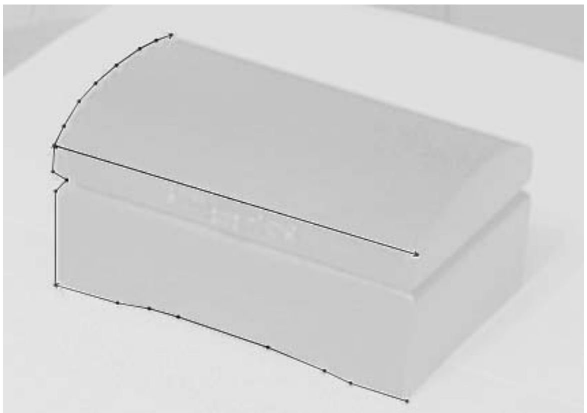


Figure 5: Obtained 3d edges.

But if the user outlined only several edges, we should construct missing edges. We analyze each face that has two non-parallel or three already constructed edges. We continue analyzing faces until we build all missing edges or if it is not possible.

If face has two adjacent edges we obtain two missing edges by shifting one 3d edge from one end point to the other. Let's consider that end points of the first edge are *{x_1, y_1, z_1}* and *{x_2, y_2, z_2}*, and end points of the second edge are {x_2, y_2, z_2} and *{x_3, y_3, z_3}*. Then third edge can be obtained with adding *{x_3-x_2, y_3-y_2, z_3-z_2}* to coordinates of points of the first edge. The forth edge can be obtained with adding *{x_1-x_2, y_1-y_2, z_1-z_2}* to coordinates of points of the second edge.

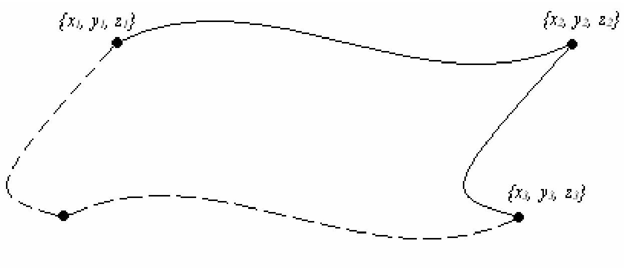


Figure 6: Missing edges construction – two adjacent edges.

If face has three edges then the missing edge can be obtained with shifting and interpolation of the middle edge. For instance, first edge end points are *{x_1, y_1, z_1}* and *{x_2, y_2, z_2}*, second edge end points are *{x_2, y_2, z_2}* and *{x_3, y_3, z_3}* and third edge end points are *{x_3, y_3, z_3}* and *{x_4, y_4, z_4}*. Let's consider that second edge has *N* point with indices *1..N*. Then missing edge would also have *N* points and point with index *i* is obtained with adding *({x_1-x_2, y_1-y_2, z_1-z_2}*i + {x_4-x_3, y_4-y_3, z_4-z_3}*(N-i))* to coordinates of point of the second edge with index *i*.

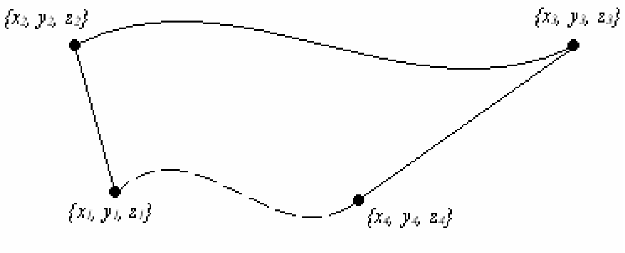


Figure 7: Missing edges construction – three edges.

3.2.3 *Face triangulation*

After all edges are constructed the full polygonal model can be built. It is done independently for each of six faces by discrete coons patches algorithm [8].

Due to low number of points in 3d edges (see step 6 in 3d edge reconstruction) and inprecision of user sketches the faces of the reconstructed model can be ragged. So additional mesh smoothing is required [9].

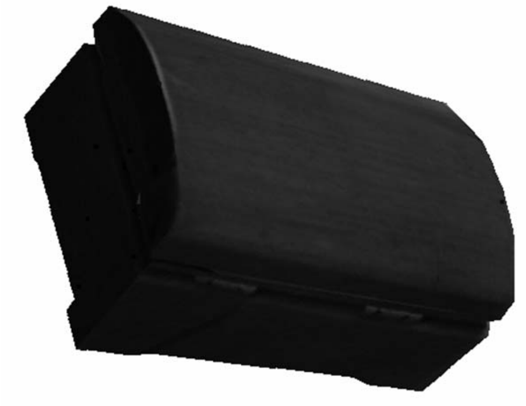


Figure 8: Rendered 3d model.

4. CONCLUSION

In this paper a new method for 3d reconstruction of objects with generalized box shape with user interaction was proposed. Using this approach 3d models of a wide class of objects may be obtained. As a part of the method an new algorithm for 3d polyline estrimation from set of 2d projection on multiple images was proposed. This method offers several advantages with comparison with other methods. It can be used for reconstruction of other types of primitives, like pyramid, etc.

The future directions of our work are the following:

- Implement additional mesh smoothing [9].

- Develop an instrument for fast and approximate marking of 2d edges with automatic refinement.

- Use proposed algorithm of 3d edge reconstruction for other types of generalized primitives.

5. REFERENCES

[1] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, *“Automatic Generation of 3D Models from Photographs”*, Proceedings Virtual Systems and MultiMedia, 1998.

[2] Z. Zhang *“Flexible Camera Calibration by Viewing a Plane from Unknown Orientations”*, International Conference on Computer Vision (ICCV'99), Corfu, Greece, pp. 666-673, September 1999.

[3] A. Laurentini, *“The Visual Hull Concept for Silhouette-Based Image Understanding”*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 2, Feb. 1994.

[4] S.M. Seitz, C.R. Dyer. *“Photorealistic Scene Reconstruction by Voxel Coloring”*, Proc. IEEE CVPR, pp. 1067-1073, June 1997.

[5] C. Robertson, R. B. Fisher, N.Werghi, A. P. Ashbrook *“Fitting of Constrained Feature Models to Poor 3D Data”*, Proceedings Adaptive Computing in Design and Manufacture, Plymouth, pp. 149-160, 2000.

[6] R. I. Hartley, P. Sturm *“Triangulation”*, ARPA Image Understanding Workshop, Monterey, California, pp. 957-966, November 1994.

[7] Pan, C., Yan, H., Medioni, G., Ma, S. *“Parametric reconstruction of generalized cylinders from limb edges”*, IP(14), No. 8, pp. 1202-1214, August 2005.

[8] G. Farin, D. Hansford *“Discrete Coons patche”*, Computer Aided Geometric Design 16, pp. 691–700, 1999.

[9] H. Zhang E. Fiume *“Mesh Smoothing with Shape or Feature Preservation”*, Advances in Modeling, Animation, and Rendering, pp. 167-182, 2002.

[10] GML C++ Camera Calibration Toolbox http://research.graphicon.ru/calibration/gml-c++-camera-calibration-toolbox-3.html.

[11] REALVIZ ImageModeler, http://realviz.com/products/im/index.php.

About the author

Anton Yakubenko is 4-th year student in Computer Science department, MSU. His email address is toh_yakubenko@mail.ru

Anton Konouchine, PhD, has received his specialist degree in computer science in 1997 in MSU. He received his PhD in 2005 in Keldysh Institute of Applied Mathematics. His email address is ktosh@graphics.cs.msu.ru

Vladimir Vezhnevets, PhD, has received his specialist degree in Moscow State University. He received his PhD in 2002 in Moscow State University. His email address is vvp@graphics.cs.msu.ru