

3D RECONSTRUCTION BASED ON STEREOVISION AND TEXTURE MAPPING

Jingchao Li *, Zhenjiang Miao, Xiangqian Liu, Yanli Wan

Institute of Information Science, Beijing Jiaotong University, 100044, Beijing, China - (08120416)@bjtu.edu.cn

KEY WORDS: 3D Reconstruction; Point Cloud; Patch based Multi-View Stereo; Surface Reconstruction; Texture Mapping

ABSTRACT:

3D reconstruction is one of the most popular research areas in computer vision and computer graphics, it is widely used in many fields, such as video game, animation and so on. It gets 3D model based on 2D images. Using this technology, we can implement scene recurrence, observe the model from any viewpoints stereoscopically and perceive the world well. In this paper, use technologies like point cloud building, surface reconstruction to obtain the visual hull. To make the visual hull looked more vivid and natural, adding texture is necessary. This research proves that this solution plan has some advantages, such as feasibility, easy reconstruction and so on.

1. INTRODUCTION

With the development of computer technology and the increase of digitalizing demand, we hope to break through the computer's information processing ability and turn it to intellectually process multi-dimensional information. In the past 30 years, many researchers work hard to fulfil this goal, most of them treat computer vision and computer graphics as the breakthrough points. 3D reconstruction based on images just uses some principles of computer vision and computer graphics to imitate our eyes. It's also an important research area.

One important branch of vision research areas is Marr model. It disassembles object into little functional units like contour, texture, depth, and then reconstruct this object according to these units. Images can reflect the depth and texture information, so 3D reconstruction based on images is feasible.

Because of 3D technology's importance, many countries have devoted on this area, for example, "Implementing Interaction and Unification between Reality and Unreality" is an important content in European Union's Sixth and Seventh Frameworks. American NIST projects like "Interactive System Technology", "Digital and Multi-Media Technology" focus on this area too. National 973 Key Research Program of China called "Interaction and Merge of Visual Media" approved in 2006 also puts emphasis on solving some key problems in visual media's efficient construction and usage, 3D reconstruction based on images is an important branch of this project.

There are also many researchers making great contributions to this area. Paper 1 and 3 mainly recount two methods of Point Cloud Building: stereovision and depth map. Paper 2 and 9 describe existing measures of surface reconstruction. Paper 4, 6, 8 and 10 summarize 3D reconstruction. The other reference documentations dwell on the concrete technologies.

After referring to a lot of papers, we compare the existing methods and choose stereovision as our solution plan for the goal of reconstructing easily. Use Patch based Multi-View Stereo [1], Poisson Surface Reconstruction [2], texture mapping consisting of texture unfolding and texture synthesis to finish

the reconstruction. We will describe this solution plan in details in the rest sections.

This paper is organized as follows: Part 2 describes 3D reconstruction based on stereovision. Part 3 is the theoretical framework of solution plan. Part 4 is the implementation of solution plan. Part 5 is the conclusion of our work.

2. RECONSTRUCTION BASED ON STEREOVISION

The definition of stereovision [8] is as follows: perceive an object from different viewpoints and obtain a series of images, deduce the spatial shape and position of this object through parallaxes between the corresponding pixels from different images.

3D reconstruction based on stereovision is defined like this: use the principle of stereovision getting a 3D model through operations of 2D images. If find a pair of corresponding pixels from two images, their back-projection rays should intersect at one point in 3D space, we can get the three-dimensional coordinate of this point. If we can get three-dimensional coordinates of all points on the object surface, the spatial shape and position of this object can be determined uniquely.

Figure 1 below shows the principle. P is a spatial point, the image coordinates of P are $(x_i, y_i)(i=1,2)$, we can compute the three-dimensional coordinate (X, Y, Z) of P based on the image coordinates and stereovision.

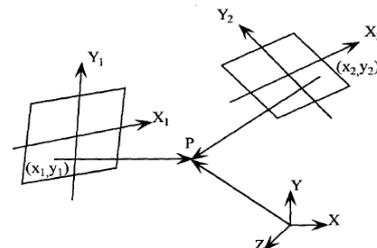


Figure 1. Principle of stereovision

3. THEORETICAL FRAMEWORK OF SOLUTION PLAN

After learning and researching many existing theories and schemes, we find a solution plan of 3D reconstruction. Figure 2 shows the theoretical framework of this solution plan.

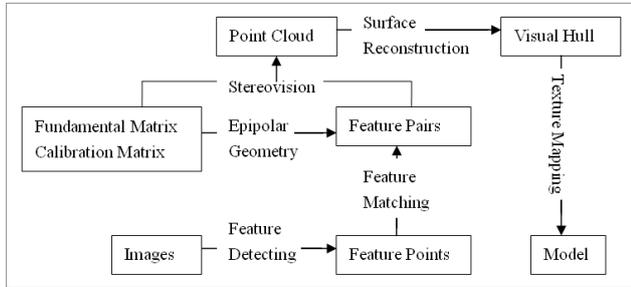


Figure 2. Theoretical framework of solution plan

①Feature Detecting: Get image features that can be used in the next step. There is no widely used image feature detection algorithms, so many kinds of feature exist. In this paper, choose DoG and Harris operators [5] to compute the feature points.

②Feature Matching: Establish the correspondences between the features got in the last step. Its availability depends on the solution of two problems: good features and good matching algorithms. In this paper, use epipolar geometry and widow based matching technology [1] to get the feature pairs.

③ Surface Reconstruction: Find a mathematical equation to describe the object's curved surface accurately and compactly. In this paper, use Poisson Surface Reconstruction [2].

④Texture Mapping: Finish a texture image and use it to render the visual hull. Here, use a texture mapping method [6,10] consisting of texture unfolding and texture synthesis.

4. IMPLEMENTATION OF SOLUTION PLAN

4.1 Point Cloud

4.1.1 Definitions:

①DoG Operator [5]: It is defined as the difference of two Gaussian-Kernels with different scales. It is easy to get and is the approximation of LoG.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

②Harris Operator [5]: For every point in an image, compute the horizontal and vertical first-order derivatives and the product of them. Handle $g_x, g_y, g_x g_y$, using Gaussian Filter. Eigenvalues of matrix M is the first-order curvature of auto-correlative function,

if the curvature value is high at a point, this point is thought to be an angular point:

$$I = \det(M) - ktr^2(M) \quad (2)$$

$$M = G(\bar{s}) \otimes \begin{bmatrix} g_x & g_x g_y \\ g_x g_y & g_y \end{bmatrix}$$

where $G(\bar{s})$ = Gaussian template
 Det = determinant of matrix M
 tr = trace of matrix M
 k = constant

③Patch Model: A patch p is an approximation of one part on the surface. Its geometry is fully determined by its centre $c(p)$, unit normal vector $n(p)$ oriented toward the camera observing it, and a reference image $R(p)$ in which p is visible.

④Photometric Discrepancy Function: Let $V(p)$ denote a set of images in which p is visible. The photometric discrepancy function $g(p)$ for p is defined as follows:

$$g(p) = \frac{1}{|V(p) \setminus R(p)|} \sum_{I \in V(p) \setminus R(p)} h(p, I, R(p)) \quad (3)$$

where $h(p, I_1, I_2)$ = a pair wise photometric discrepancy function between images I_1 and I_2 [3]

Concretely, only images whose pair wise photometric discrepancy score with the reference image $R(p)$ is below a certain threshold α are used for the evaluation:

$$V^*(p) = \{I | I \in V(p), h(p, I, R(p)) \leq \alpha\} \quad (4)$$

$$g^*(p) = \frac{1}{|V^*(p) \setminus R(p)|} \sum_{I \in V^*(p) \setminus R(p)} h(p, I, R(p))$$

⑤Patch Optimization: $c(p)$ and $n(p)$ are optimized by simply minimizing the photometric discrepancy score $g^*(p)$. To simplify computations, constrain $c(p)$ to lie on a ray such that its image projection in the reference image does not change, $n(p)$ is parameterized by Euler Angle [7].

⑥Image Model: Associate with each image I_i $\beta \times \beta$ pixels cells $C_i(x, y)$ as in Figure 3. Give a patch p and its visible images $V(p)$, project p into each image in $V(p)$ to identify the corresponding cell. Each cell $C_i(x, y)$ remembers the set of patches $Q_i(x, y)$ that project into it. Use $V^*(p)$ to define $Q_i^*(x, y)$.

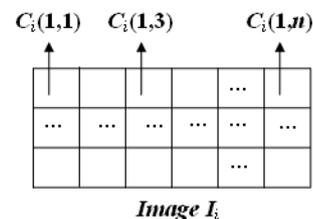


Figure 3. Description of cells

4.1.2 Feature Detecting: To ensure uniform coverage, lay over each image $\beta_1 \times \beta_1$ pixels blocks, and return as features the η local maxima with the strongest responses in each block for each operator. Figure 4 shows the results (DoG is denoted by red. Harris is denoted by green.):

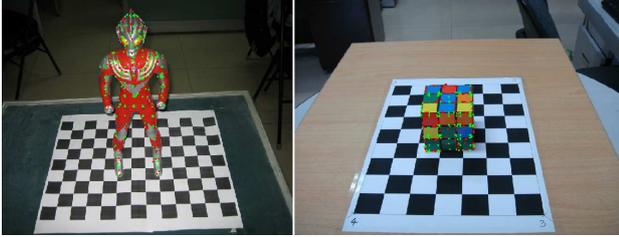


Figure 4. Results of feature detecting

4.1.3 Feature Matching: Consider an image I_i and the optical centre of the corresponding camera denoted by $O(I_i)$. For each feature f detected in I_i , collect in the other images the set F of features f' of the same type that lie within two pixels from the corresponding epipolar lines, and triangulate the 3D points associated with the pairs (f, f') . First construct a patch candidate p with its centre $c(p)$, normal vector $n(p)$ and reference image $R(p)$ initialized as:

$$\begin{aligned} c(p) &\leftarrow \{\text{Triangulation from } f \text{ and } f'\} \\ n(p) &\leftarrow c(p)\vec{O}(I_i) \mid c(p)\vec{O}(I) \mid \\ R(p) &\leftarrow I \end{aligned} \quad (5)$$

Simply assume that the patch is visible in an image I_i when the angle between the patch normal and the direction from the patch centre to the optical centre $O(I_i)$ is below a certain threshold τ :

$$V(p) \leftarrow \{I \mid n(p) \cdot c(p)\vec{O}(I_i) \mid c(p)\vec{O}(I) \mid > \cos(\tau)\} \quad (6)$$

Having initialized all the parameters for the patch candidate p , refine $c(p)$ and $n(p)$, then update the visibility information $V(p)$ and $V^*(p)$ with the refined geometry. If $|V^*(p)|$ is at least γ , the patch generation is deemed a success and p is stored in the corresponding cells of the visible images, then update $Q_i(x, y)$ and $Q_i^*(x, y)$.

4.1.4 Feature Expansion: The goal of the expansion step is to reconstruct at least one patch in every image cell $C_i(x, y)$. More concretely, give a patch p , first identify a set of neighbour empty image cells $C(p)$, then perform a patch expansion procedure for each one of the cells in $C(p)$.

① Identifying Cells for Expansion: Give a patch p , first initialize $C(p)$ by collecting the neighbour image cells in its each visible image:

$$C(p) = \{C_i(x', y') \mid p \in Q_i(x, y), |x - x'| + |y - y'| = 1\} \quad (7)$$

Next, remove image cells from $C(p)$ according to the following two criteria. First, the expansion for an image cell is unnecessary if a patch has been already reconstructed there. Second, a patch should not be expanded for an image cell if there is a depth discontinuity when viewed from the corresponding camera.

② Expansion Procedure: For each image cell $C_i(x, y)$ in $C(p)$, generate a new patch p' like this: first initialize $n(p')$, $R(p')$ and $V(p')$ by the corresponding values of p . $c(p')$ is initialized as the point where the viewing ray passing through the centre of $C_i(x, y)$ intersects the plane containing the patch p . After computing $V^*(p')$, refine $c(p')$ and $n(p')$. Then add to $V(p')$ a set of images in which the patch should be visible, then update $V^*(p')$. Finally, if $|V^*(p')| \geq \gamma$, accept the patch as a success and update $Q_i(x, y)$ and $Q_i^*(x, y)$.

4.1.5 Definitions: The following three filters are used to remove erroneous patches. The first filter relies on visibility consistency. Let $U(p)$ denote the set of patches p' that are inconsistent with the current visibility information—that is, p and p' are not neighbours, but are stored in the same cell of one of the images where p is visible. Then, p is filtered out as an outlier if the Eq. 8 holds. The second filter also enforces visibility consistency, but more strictly: for each patch p , compute the number of images in $V^*(p)$ where p is visible. If the number is less than γ , p is filtered out as an outlier. In the third filter, enforce a weak form of regularization: for each patch p , collect the patches lying in its own and adjacent cells in all images of $V(p)$. If the proportion of patches that are neighbours of p in this set is lower than λ , p is removed as an outlier.

$$|V^*(p)| (1 - g^*(p)) < \sum_{p_i \in U(p)} 1 - g^*(p_i) \quad (8)$$

4.2 Surface Reconstruction

We approach the problem of surface reconstruction using an implicit function framework [9]. Compute a 3D indicator function χ (1 denotes points inside the model, 0 denotes points outside), then obtain the reconstructed surface by extracting an appropriate isosurface.

The input data S is a set of sample $s \in S$, each consisting of a position $s.p$ and an inward-facing normal $s.N$, assumed to lie on or near the surface ∂M of an unknown model M . Our goal is to reconstruct a triangulated approximation to the surface by approximating the indicator function of the model and extracting the isosurface.

The key insight is that there is an integral relationship between oriented points sampled from the surface of a model and the indicator function of the model. The oriented point samples can be viewed as samples of the gradient of the model's indicator function. Find the scalar function χ whose gradient best approximates a vector field \vec{V} defined by the samples, i.e. $\min_{\chi} \|\nabla \chi - \vec{V}\|$. If apply the divergence operator, this problem transforms into a standard Poisson problem: compute the scalar function χ whose divergence of gradient equals the divergence of the vector field \vec{V} :

$$\Delta\chi = \nabla \cdot \nabla\chi = \nabla \cdot \vec{V} \quad (9)$$

In order to obtain a reconstructed surface ∂M , it is necessary to first select an isovalue and then extract the corresponding isosurface from the computed indicator function. We do this by evaluating χ at the sample positions and use the average of the values for isosurface extraction:

$$\partial M \equiv \{q \in R^3 \mid \chi(q) = \gamma\} \quad \gamma = \frac{1}{|S|} \sum_{s \in S} \chi(s.p) \quad (10)$$

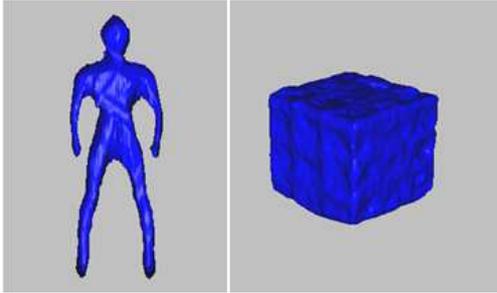


Figure 5. Results of surface reconstruction

4.3 Texture Mapping

Texture mapping is defined as follows: define a texture image in advance, then based on some mapping algorithms, establish the correspondences between points in the texture image and points on the visual hull, finally, render the visual hull using the texture image. We must notice that: the visual hull is three dimensional, and the texture image is two dimensional. Figure 6 shows the principle of texture mapping.

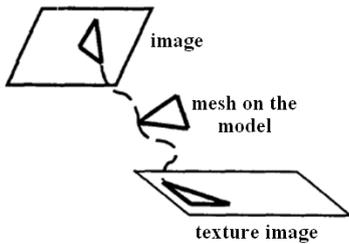


Figure 6. Principle of texture mapping

4.3.1 Texture Unfolding: Texture unfolding is to establish the correspondences between the 3D visual hull and the 2D texture image, in a sense, it is the parameterization of the visual hull. Visual hull parameterization can be summarized as this: give a 3D visual hull M consisting of points $V \in R^3$ and a 2D parameter field Ω consisting of points $V^* \in R^2$, find a one-to-one mapping φ between points $V^* \in \Omega$ and points $V \in M$, making sure that meshes in the parameter field and meshes on the visual hull are topological isomorphism.

Frequently used parameterization methods include flat surface parameterization, cylindrical surface parameterization and spherical surface parameterization. We use flat surface parameterization in this paper.

Intuitively, flat surface parameterization unfolds the 3D visual hull into a 2D texture image. Compute the visual hull’s AABB bounding box – a hexahedron, whose every face is parallel to one of the faces decided by X, Y, Z axis, so computing direction vector of every face is possible. Project the visual hull to the six faces, unfold the hexahedron, we can get the correspondences between the 3D visual hull and the 2D flat surface texture image.

Because there are six projection faces, selecting the correct projection face for every mesh on the visual hull is important to the accuracy of parameterization. Give normal vector of one mesh on the visual hull \vec{N} and direction vector of one projection face \vec{F} , we can use Eq. (11) to choose the appropriate projection face.

$$\vec{N} * \vec{F} \geq 0 \quad (11)$$

4.3.2 Texture Synthesis: Now, we have got the correspondences between the visual hull and the texture image. Next, we should compute the texture image’s pixel values. We can get the image coordinates of every vertex on the visual hull based on the projection matrixes, but there are still two problems: first, all vertexes can’t be just projected in only one image, in other words, there must be some points hiding other points, use mesh visibility estimation to solve it, here, mesh represents the mesh on the visual hull; second, one vertex may correspond to more than one image, because the vertex can be taken from different viewpoints, so it can appear in more than one image. Weighed mean can solve this problem.

⊙ Mesh Visibility Estimation: Under normal circumstances, whether a mesh is visible or not in an image can be estimated by the angle between normal vector of the mesh and viewpoint vector of the image.

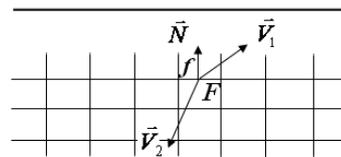


Figure 7. Sketch-map of mesh visibility estimation

In Figure 7, \vec{N} is the normal vector of mesh f , \vec{V}_1 and \vec{V}_2 are two viewpoint vectors, F is the centre of f . $\vec{N} \cdot \vec{V}_1 > 0$, so f may be visible to V_1 , $\vec{N} \cdot \vec{V}_2 \leq 0$, so f is not visible to V_2 . If one mesh is not visible to an image, then we don’t consider this image when compute texture information of the mesh.

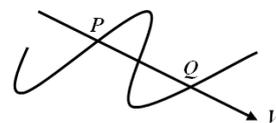


Figure 8. Sketch-map of occlusion problem

In Figure 8, when $\vec{N}_p \vec{V} > 0$ and $\vec{N}_q \vec{V} > 0$, in other words, two vertexes P and Q on the visual hull are both visible to the viewpoint V , and P, Q, V lie on the same straight line. Because $|PV| > |QV|$, we can deduce that P is not visible to V , that is to say, P is hidden by Q , then when compute texture information of P , don't consider the image with viewpoint V , but for the computation of Q , should consider it.

② Weighted Mean: Every vertex on the visual hull may correspond to many images, we need to summarize these images to get the best texture image. We use weighed mean here, give a vertex V and an image l , the angle between V and l is k_l , this angle is defined by the optical axis and the straight line determined by the projection point and the viewpoint, for the set of images in which this vertex is visible, the biggest angle is k_m , image l 's weight is defined as $1-k_l/k_m$. We define pixel value of vertex V like this:

$$T_v = \sum_{i=1}^N k_i \times t_i (i \neq V) \quad (12)$$

where $T_v = V$'s pixel value
 $k_i = V$'s weight referring to image i
 $t_i =$ pixel value of the projection point in image i corresponding to V
 $N =$ number of images where V is visible

Now, we just know the pixel values of the vertexes on the visual hull, they are not enough to fill the texture image, so we use interpolation to finish it.

Give a triangle whose vertex coordinates are known, for every point in this triangle, its coordinate can be decided uniquely by vertex coordinates and its areal coordinate.

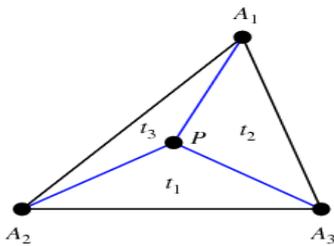


Figure 9. Illustration of areal coordinate

For point P , the way of computing its areal coordinate is like this:

$$t_1 = T_1/T, t_2 = T_2/T, t_3 = T_3/T \quad (13)$$

where $T_1 =$ area of ΔA_2PA_3
 $T_2 =$ area of ΔA_1PA_3

$$T_3 = \text{area of } \Delta A_2PA_1$$

$$T = \text{area of } \Delta A_1A_2A_3$$

For point P , its coordinate can be computed like this:

$$\vec{V} = t_1 \times \vec{V}_1 + t_2 \times \vec{V}_2 + t_3 \times \vec{V}_3 \quad (14)$$

where $\vec{V}_1, \vec{V}_2, \vec{V}_3 =$ vertex coordinates
 $\vec{V} =$ point P 's coordinate
 $(t_1, t_2, t_3) =$ point P 's areal coordinate

Through above-mentioned computation, we can know every pixel value in the texture image, use it to render the visual hull as shown in Figure 10.



Figure 10. Results of rendering

5. CONCLUSION

On the basis of many researchers in this area, we use the technologies like patch-based multi-view stereo, Poisson surface reconstruction, texture mapping to finish the work, but our work just gets off the ground, we need to go further.

3D reconstruction is an important research area in computer vision. It can be used in many areas from visual navigation of robot to 3D games, digital library, visual communication, virtual reality, internet travelling and so on. Though many researchers have made a great improvement and have had many achievements in this area, it still has a long distance between theory research and practical application [4].

6. ACKNOWLEDGEMENTS

This work is supported by National 973 Key Research Program of China (2006CB303105) and Innovative Team Program of Ministry of Education of China (IRT0707).

7. REFERENCES

Yasutaka Furukawa, Jean Ponce, 2007. Accurate, dense and robust multi-view stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 17-22 June, pp. 1-8.

Michael Kazhdan, Matthew Bolitho, Hugues Hoppe, 2006. Poisson surface reconstruction. In: Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June, pp. 61-70.

Michael Goesele, Brian Curless, Steven M. Seitz, 2006. Multi-view Stereo Revisited. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 2402-2409.

Chunli Wang, Libo Liang, Baoyu Wang, 2003. Development and application of computer 3D reconstruction. Journal of Shenyang University, Vol. 15, No. 2.

Jian Gao, Xinhua Huang, Gang Peng, Min Wang, Zuyu Wu, 2008. A Feature Detection Method Based on Harris Corner and Difference of Gaussian. Journal of Pattern Recognition and Artificial Intelligence, Vol. 21, No. 2.

Yanping Wu, Zhenjiang Miao, 2009. Three-dimensional Reconstruction Based on Images. Master degree thesis, Beijing Jiaotong University.

Shuhui Jia, 1991. General Euler Angles and Its Application. Journal of Mechanics and Engineering, Vol. 13, No. 4, pp. 54-58.

Hua Duan, Dongbiao Zhao, 2003. 3D reconstruction of the computer vision based on binocular vision systems. Master degree thesis, Nanjing University of Aeronautics and Astronautics.

Lixia Guo, Baosheng Gang, 2006. On Point-cloud data Reconstruction Using RBF. Master degree thesis, Northwest University.

Yuru Pei, Yue Zhao, 2003. Fast polyhedral surface mesh generation algorithm for 3D objects based on 2D photo images. Journal of Zhejiang University(Engineering Science), September, Vol. 37, No.5.