Automatic merging of DSMs extracted by image matching algorithms

Verkeenko Mikhail
The Federal State Unitary Enterprise
«State Research Institute of Aviation Systems», Moscow, Russia
envenom.gm@gmail.com

Abstract

In this paper a new approach to building 3D model using existing Digital Surface Models (DSMs) extracted by image matching algorithms is proposed. This problem arises when the 3D object to be reconstructed has a very complicated form and its full surface cannot be restored using only one stereo pair. A new algorithm based on image orientation data and 2D Delaunay triangulation is presented. This method is compared with another widespread approach for 3D surface triangulation. Two types of objects have been used for this article: extended bas-relief and the closed object "Monument to the Heroes of Plevna".

Keywords: *3D model triangulation, DSM, photogrammetry, image matching, Delaunay triangulation, merging of triangulations*

1. INTRODUCTION

At current, digital models of objects are using in many areas: electronic geographic information systems, virtual and augmented reality, mission planning and rehearsal, urban planning etc.. Also, a variety of models is in demand now: typical mass-produced houses of urban development, unique historical buildings, palaces, castles and unique works of architecture, such as the bas-reliefs and sculptures. Automated construction of digital models of objects based on ground-based images and reference data is a science-intensive and highly relevant problem. To date, photogrammetric systems can give the same dense cloud of points, as well as laser scanning [1, 8]. This explains the increasing importance of developing special algorithms and digital photogrammetric systems for data processing ground surveys.

We are not facing problems in case of using only one stereo pair for the reconstruction of the object's surface; since we can use 2D Delaunay triangulation for the corresponding projection of matched point cloud on a plane of stereo pair. Unfortunately, it's unable to reconstruct all closed objects using only one stereo pair. Therefore a problem of merging a number of DSMs arises.

2. RELATED WORKS

Since the problem of surface reconstruction from point cloud is very important, it was proposed many methods over past decades. Several approaches are based on combinatorial structures such as 3D Delaunay triangulations, alpha shapes [5], or Voronoi diagrams [3]. These schemes typically create a triangle mesh that interpolates all or a most of the points. Other schemes directly reconstruct an approximating surface, typically represented in implicit form [4]. A Poisson Surface Reconstruction (PSR) technique [3], developed by Kazhdan et al. (2006), has been adopted for mesh generation. This technique is a novel approach that expresses surface reconstruction as the solution to a Poisson equation. Kazhdan et al. (2006) demonstrate that the Poisson algorithm can facilitate the reconstruction of surfaces with greater detail than previously achieved. This algorithm uses points positions and their normals for surface reconstruction.

Another class of approaches analyzes same triangles in neighboring triangulations and uses them for surfaces transformation and merging. But these methods are unstable in case of different image acquisition conditions, which can cause difference between matched points or reconstructed triangles in the intersection zone [7].

The main feature of our approach is to use an advantage of image orientation data. It can be helpful for moving the problem from 3D space into 2D.

3. INPUT DATA

The input data for the merging procedure is:

- A set of point clouds corresponding to each stereo pair (Figure 1). For each point cloud a triangulation that accurately approximates the actual surface of the object can be computed (Figure 2). Point clouds must have areas of intersection; otherwise the procedure of merging cannot be executed.

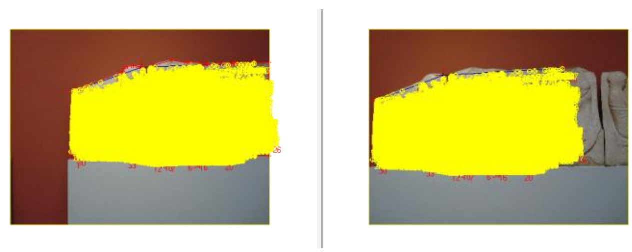


Figure 1: Point cloud for a part of bas-relief

Figure 2: Triangulation for a part of bas-relief

- Possible constraints for triangulation of current point cloud.
- Orientation for each stereo pair, allowing to project points from 3D space onto an image plane, to restore point position in the 3D space based on 2D points from left and right stereo pair images and to determine

position and view vector of camera attached to the current image (Figure 3).

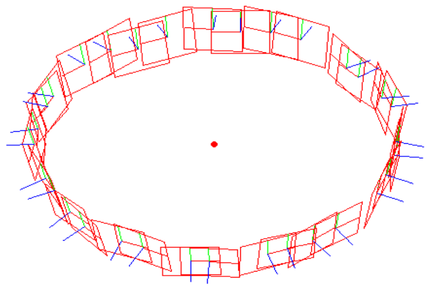


Figure 3: Orientation scheme for the object "Monument to the Heroes of Plevna"

4. THE PROPOSED METHOD

The proposed method consists of few steps:

- Creating the initial triangulation (mesh) for each of the matched stereo pairs and marking one of them as "current".
- Iterative process
    - Searching for adjacent triangulations, which have zones of intersection with the current triangulation.
    - Merging of 2 triangulations.

Iterative process will change triangulation only in the zone of intersection.

4.1 Preparing initial triangulations

This is a fairly simple step, since it's sufficient to project the points of each cloud on the stereo pair and to triangulate projected points using Delaunay triangulation procedure [2]. We have to take triangulation constraints into consideration, if it's necessary. The resulting triangulation quite accurately reflects the actual geometry of the matched surface, as it's performed on the points extracted from respective images. Also, it solves the problem with overlapping of triangles – all triangles are visible from the corresponding point of view.

4.2 Searching for neighbor triangulations and determination of the mutual intersection zone

4.2.1 Search for neighbors

Firstly, we must introduce a criterion to mark a triangulation as a neighbor. We can check intersections between bounding shapes(Oriented Bounding Box (OBB) for tested triangulation or 3D convex hull for tested triangulation) and current triangulation's points for this purpose (It would be better to utilize OBB if the computational time is critical). If there are few triangulations, that are neighbors, then we can use them together or separately, processing the first triangulation with maximum intersection count. In this work preferences were given to the latest version, as the most evident. The only exception is the closure of the object. We have a high probability that a mutual intersection zone exists in case of the exact orientation and matching.

4.2.2 Determination of the mutual intersection zone

After finding the neighbor triangulation, that is suitable for merging, we should more accurately identify the zone of mutual intersection. We can take intersection points from subsubsection 4.2.1 and identify appropriate triangles from current triangulation. Points and triangles from the neighbor triangulation can be defined similar. Thus we get:

- Point cloud inside the mutual intersection zone (red).
- Triangles outside the intersection zone (green).

In practice, often turns out that this data cannot be used for merging without obtaining various artifacts, because of some "garbage" triangles inside the intersection zone and "jagged and ragged" border between outside triangulation and point cloud inside the intersection zone.

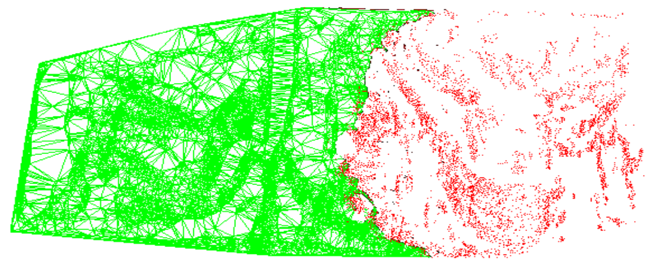


Figure 4: "Outside" triangulation (green) and point cloud inside the intersection zone (red)

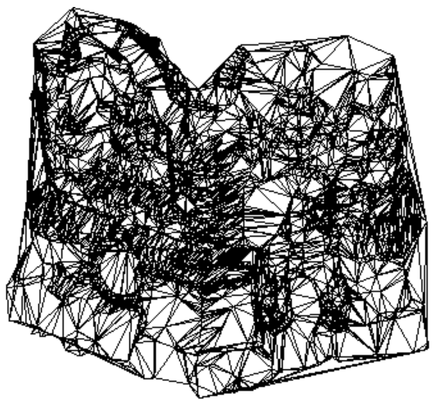
In this way, we have to smooth the border and to remove this garbage. Alternately project point cloud onto 2 planes: neighbor triangulation plane (equal to the image plane from which it was extracted) and another plane which has corresponding camera position near the intersection zone. For each projection we will find 2D bounding box. Additionally we will build graph which uses triangles as nodes for each "outside" triangulation. Using the information about triangles that are exactly inside "old" intersection zone, we can make tracing of triangulation to 2D bounding box boundaries and delete traced and garbage triangles inside the intersection zone. Additionally, we will add new points from deleted triangles into point cloud. As a result, the border between outside triangulation and intersection zone will be more "smoothly", so we can avoid some problems with artifacts in triangulation by merging. After this operation we have to delete repeating points from the intersection zone using some kind of criterion based on distance between points for example. If there are few intersection zones, then we must process them separately.

4.3 Building triangulation in the mutual intersection zone

Now we can make triangulation of target point cloud in the intersection zone. 2D Delaunay triangulation is the simplest way. Therefore we must choose the best projection for our 3D point cloud.

It's obviously, that it's not desirable to use real camera projections described in subsubsection 4.2.2, because the point cloud contains points from opposite borders and these boundaries can be greatly distorted in case of using such projections. So, it's necessary to create any synthetic projection, which will be an average between real projections and will not cause huge distortions in the borders geometry.

The solution is based on the next approach. We have to take two nearby positions of cameras and have to calculate any point A, lying on the view ray $\overline{Vec1}$ one of them (Figure 5). Then we make additional plane using three points: Cam1, Cam2 and A; and project view ray of camera 2 $\overline{Vec2}$ on this plane. Point B is an intersection of this projection and $\overline{Vec1}$. This point is also a center of circle with radius $R = \left|\overline{(Cam1, B)}\right|$. The synthetic position of camera is a point C, which is lying in the middle of circular arc between points Cam1 and Cam2'. The view vector of a new camera is the vector between points B and C, and the up vector is the normal to the plane (Cam1, Cam2, A).



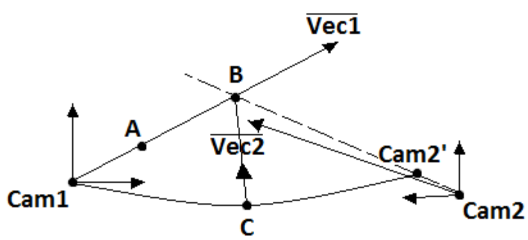


Figure 5: Calculating synthetic camera's position and view vector

After choosing a new projection, we can get 2D coords of point cloud to implement 2D Delaunay triangulation. It should be mentioned, that we have to take borders between outside triangulations and an intersection zone into consideration. In this case, we can use constraint Delaunay triangulation to consider these borders [2]. There is still one remaining issue – Delaunay triangulation is building triangles into convex hull, so there can be some intersections between edges of “old” and “new” triangles. To avoid this problem the intersection of built triangles with part of outside triangulation can be checked.

5. COMPARSION WITH PSR APPROACH

In the section 2 Poisson Surface Reconstruction approach for reconstructing surface from oriented point cloud was mentioned [3]. This approach is implemented for example in a well-known Computational Geometry Algorithms Library CGAL (www.cgal.org). As already mentioned, to perform this algorithm is required to have only points positions and normals, which are easily obtained from our orientation and matching data.

The results of two algorithms are presented in the figure 6 and figure 7. As you can see, our approach preserves the geometry in its initial condition, including the number of polygons, in addition, it executes a little faster (at 50K points proposed algorithm takes 15 s and PSR takes 20 s). To obtain acceptable results by merging DSMs is desirable to have a blunder-free surface. Also, this method does not require dense point cloud.

6. CONCLUSION

An approach for merging DSMs extracted by image matching was proposed in this paper. The proposed method takes advantages of the image orientation and produces accurate and complete surface of an object without artifacts and holes. The proposed method was compared to a PSR approach; advantages in terms of accuracy and initial condition of the reconstructed surface, as well as the computational time have been shown. There is also an example of merging surfaces for the rather complicated closed object (figure 8).

7. FUTURE WORK

It is planned to test this method on a large number of objects, to optimize it and to develop an algorithm for automatic creation of photorealistic textures. This application will be a part of a photogrammetric system, containing modules for automatic orientation of images, automatic matching and reconstruction of surfaces, merging of extracted DSMs and texturing. Consequently, we can develop an effective workflow for automatic building of realistic and precise digital models of existing objects.

8. REFERENCES

[1] Блохинов Ю.Б., Веркеенко М.С.. Алгоритмы построения цифровых трехмерных моделей уникальных объектов, Известия РАН. Теория и системы управления, М., №4, 2011, с.xxx-xxx

[2] Скворцов А.В.. Триангуляция Делоне и её применение, Издательство Томского университета, 2002. 127 с.

[3] Amenta N., Bern M., Kamvysselis M.. A new Voronoi-based surface reconstruction algorithm. Computer Graphics (SIGGRAPH ’98) (1998), 415–21.

[4] Carr J., Beatson R., Cherrie H., Mitchel T., Fright W., Mccallum B., Evans T.. Reconstruction and representation of 3D objects with radial basis functions. SIGGRAPH (2001), 67–76.

[5] Edelsbrunner H., Mucke E.. Three-dimensional alpha shapes. TOG (1994), 43–72.

[6] Michael Kazhdan, Matthew Bolitho, Hugues Hoppe. Poisson Surface Reconstruction, Eurographics Symposium on Geometry Processing, 2006.

[7] Thomas R. Hudson. Merging VRML models: extending the use of photomodeller, University of Virginia, 1998.

[8] Zhang L.. Automatic Digital Surface Model Generation from Linear Array Images // DISS.ETH NO. 16078. Zurich, 2005. 219 p.

About the author

Verkeenko Mikhail – Ph.D. student at State Research Institute of Aviation Systems.

Email: envenom.gm@gmail.com

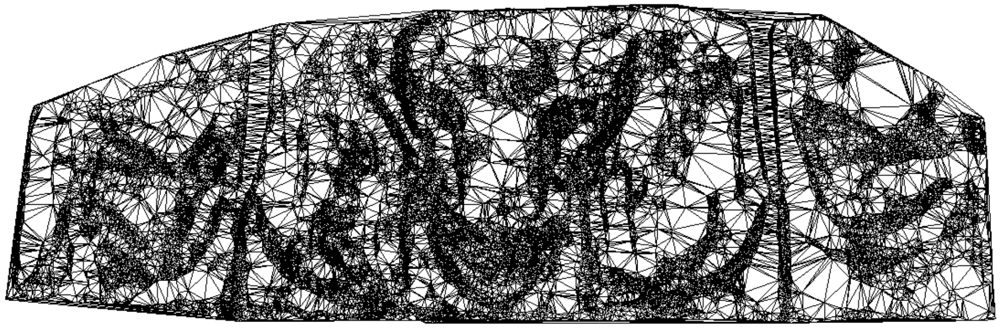


Figure 6: Result of proposed approach

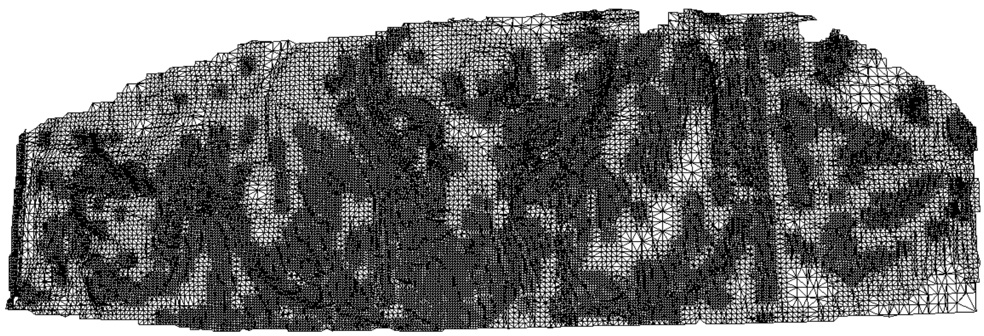


Figure 7: Result of Poisson surface reconstruction

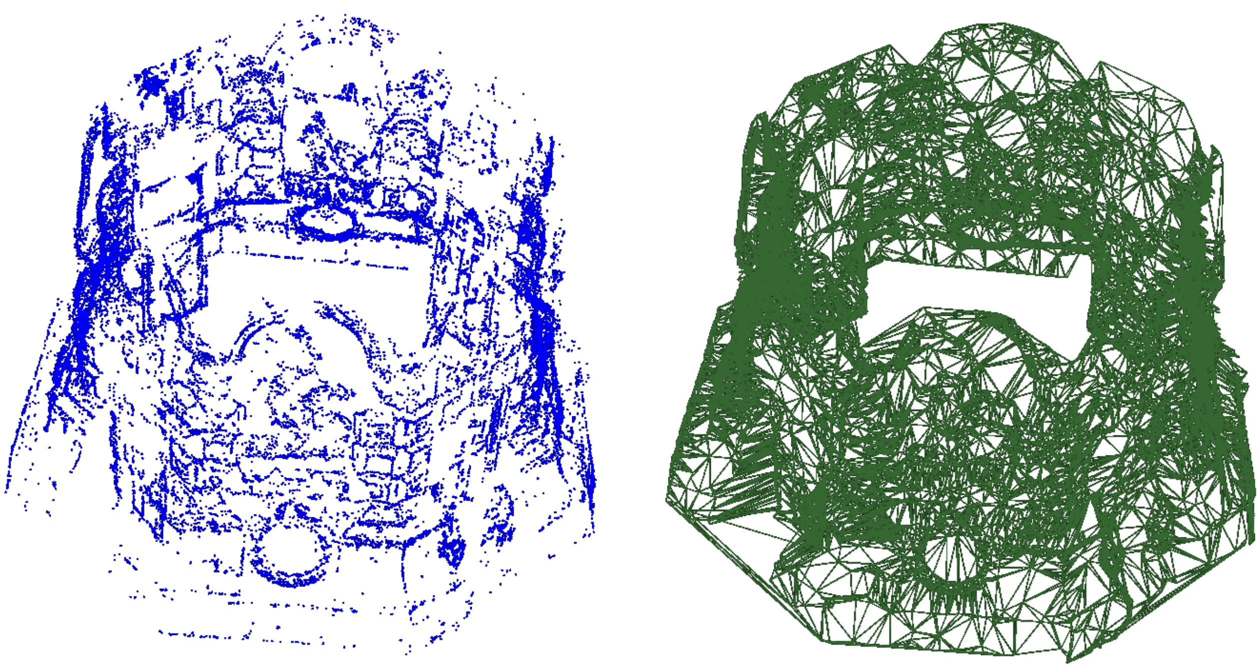


Figure 8: Point cloud and triangulation of the object "Monument to the Heroes of Plevna"