# OPTIMIZING MESH RECONSTRUCTION AND TEXTURE MAPPING GENERATED FROM A COMBINED SIDE-VIEW AND OVER-VIEW IMAGERY

Shuang Song[1], Rongjun Qin [1, 2, *]

[1] Dept. of Civil, Environmental and Geodetic Engineering, The Ohio State University, Columbus, USA - (song.1634, qin.324)@osu.edu
[2] Dept. of Electrical and Computer Engineering, The Ohio State University, Columbus, USA - qin.324@osu.edu

**Commission II, WG II/4**

**KEY WORDS:** Surface reconstruction, Visibility, Delaunay tetrahedron, Texture mapping, Data fusion, Digital Surface Models (DSM)

## ABSTRACT:

Image-based 3D modelling are rather mature nowadays with well-acquired images through standard photogrammetric processing pipeline, while fusing 3D dataset generated from images with different views for surface reconstruction remains to be a challenge. Meshing algorithms for image-based 3D dataset requires visibility information for surfaces and such information can be difficult to obtain for 3D point clouds generated from images with different views, sources, resolutions and uncertainties. In this paper, we propose a novel multi-source mesh reconstruction and texture mapping pipeline optimized to address such a challenge. Our key contributions are 1) we extended state-of-the-art image-based surface reconstruction method by incorporating geometric information produced by satellite images to create wide-area surface model. 2) We extended a texture mapping method to accommodate images acquired from different sensors, i.e. side-view perspective images and satellite images. Experiments show that our method creates conforming surface model from these two sources, as well as consistent and well-balanced textures from images with drastically different radiometry (satellite images vs. street-view level images). We compared our proposed pipeline with a typical fusion pipeline - Poisson reconstruction and the results show that our pipeline shows distinctive advantages.

## 1. INTRODUCTION

### 1.1 Introduction

Surface models through meshing point clouds are important presentations in Geomatics and Computer Graphics community. State-of-the-art approaches are well developed for well-acquired and single-source images, which result in many successful software packages and processing pipelines, such as Pix4D (2014), ContextCapture (Antila et al., 2011), Vricon (Landry et al., 2016), RSP (Qin, 2016). However, meshing conflated 3D dataset generated from images of different view and sensors for surface reconstruction is extremely challenging. Nowadays the tasks of 3D modeling and meshing may utilize diversified data sources, such as RGB or depth images from low-cost cameras/depth ranger or expensive LIDAR (Light Detection and Ranging) data. However, the current practice in using multi-source data for meshing is still rather intuitive. For example, data from different sources are often processed separately using well-developed approaches to generate point clouds, and then apply a point cloud based meshing algorithms (Kazhdan et al., 2006), followed by a post texture mapping using available oriented images. It is known that meshing from image-based point clouds often utilizes the visibility information that codes for each point, which image (if available) observes it, as such information implicitly provides the surface information with respect to the camera positions. Meshing from the level of combined point clouds (from different sources) are likely to miss the visibility information, because on one hand, most of the image-based 3D modeling software packages do not output the visibility information, and on the other hand, when multiple cameras are involved, implementing individual camera models for visibility can be trivial processes. In this paper, we take the challenge by proposing a meshing method that integrates point clouds generated from the overview satellite images and side-view perspective images. Instead of re-implementing different camera sensor models into an image-based point cloud meshing pipeline, we assume the point clouds generated from the satellite images are associated with top-view parallel cameras, which is much easier for encoding the visibility information. In addition, we take this idea to further extend an existing texture mapping method to incorporate satellite images with street-view images for seamless texture mapping.

### 1.2 Related Works

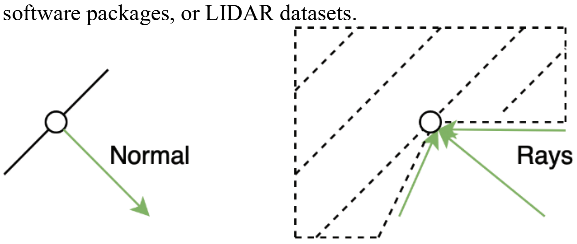
Existing surface reconstruction from point set methods can be categorized into implicit surface methods and Delaunay-based methods (Labatut et al., 2009). Implicit methods construct a function of space from samples, a surface can be implicitly defined as a level-set of the function allowing smooth and approximating surface reconstruction, however sometimes they are sensitive to noise, outliers or non-uniform sampling or even simply by a lack of reliable and consistent normal (Labatut et al., 2009). Among implicit methods, Poisson surface reconstruction (Kazhdan et al., 2006) is regarded one of the most commonly used methods, even though relies on surface normal, and have been implemented in many software packages, including CloudCompare (Girardeau-Montaut, 2016), MeshLab (Cignoni et al., 2008), etc. Another family of Delaunay-based methods that construct tetrahedra that partition the space to compact simplices and eliminate facets of Delaunay tetrahedra according to certain criteria such as being in an inner or outer surface. One of the most widely used techniques for large-scale surface reconstruction was the work of Labatut, et

* Corresponding author

https://doi.org/10.5194/isprs-annals-V-2-2020-403-2020

al. (2009), which was further improved by Jancosek and Pajdla (2014). In this approach, a *s-t* (source-sink, acyclic) graph (Dantzig and Fulkerson, 2003) is derived from Delaunay tetrahedra, the minimal *s-t* cut of graph labels each tetrahedra as either occupied or free. As a result, surfaces located at the interface between source labeled and sink labeled tetrahedron. To construct *s-t* graph, Labatut, et al. (2009) introduced ray visibility information derived from structure-from-motion (SFM) (Moulon et al., 2016) processing to compute weights for each tetrahedron. Our surface reconstruction pipeline is extended from this work, and the visibility information plays an essential role carrying ray information.

As compared to the normal of points, the ray visibility preserves raw observations which helps perform robust surface reconstruction: as shown in Figure 1a, a normal will deterministically pick a specific tangent plane on one point in surface optimization, which can be affected by outliers and noises. In contrast, the rays cover a range of possible orientations that surface may have at this point (Figure 1b), which allows more flexible formulation for surface optimization. However, the visibility information is not always available, such as point clouds generated from commercial software packages, or LIDAR datasets.



(a) (b)

Figure 1. (a): Green arrow is normal of the object point; black line represents surface determined by normal. (b): Green arrows are rays corresponding to the object point. Shadowed area represents possible area of surface implied by ray information.

The image-based 3D mesh reconstruction algorithms are well-practiced in single-sourced data, while processing data from multiple sources is non-trivial. In this paper, taking 3D point clouds generated from satellite images, and 3D point clouds produced by an SFM pipeline (with the associated visibility and poses of the images), we propose a method to reconstruction textured meshes from such data, specifically, in our method,

1. We extend state-of-the-art Delaunay tetrahedron based surface reconstruction method which modelled ray visibility by adapting satellite stereo height map to create large scale, coarse-fine fused surface model.

2. We extend a texture mapping method to accommodate images acquired from different sensors, i.e. side-view perspective images and satellite images.

We will introduce our proposed mesh reconstruction method in Section 2 and texture mapping method in Section 3 in detail. We practice our proposed method on overview satellite point clouds and street-view point clouds and report the results in Section 4; Section 5 concludes this paper and proposes our future works.

## 2. OPTIMIZING MESH RECONSTRUCTION

### 2.1 Data Description

Our method focuses on multi-source fusion specifically for side-view and over-view dataset as shown in Figure 2 and Figure 3. Furthermore, the satellite dataset comes approximately geo-referenced (based on their sensor accuracy) (Barazzetti et al., 2016; Mandanici et al., 2019) in which the Z values of the resulting point clouds refers to the evaluation. The approach we used takes building boundaries as the common feature aligning street-view and overview datasets. The co-registration achieves an RMSE of 1.44 m in error, which are reasonable considering that the satellite point clouds have a resolution of 0.5 m (Qin, et al, 2020).We also assume that side-view images have been oriented with their resulting point clouds co-registered with the satellite point clouds, an example of the dataset is shown in Figure 4.

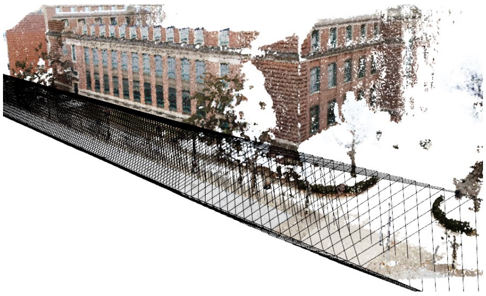


Figure 2. Reconstructed camera poses (black wireframes) and the generated 3D color point clouds from the street-view images.

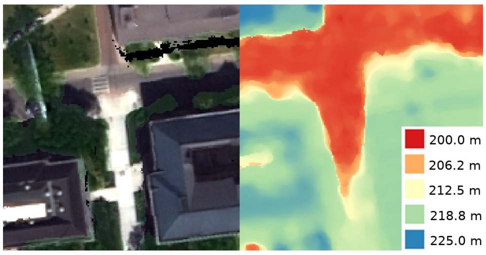


Figure 3. Left: orthophoto generated from satellite images. Right: DSM generated from satellite images.

The resolution of side-view and over-view sources are significantly different, as the GSD (ground sampling distance) of the satellite point clouds is about 0.5 m, and pixel GSD of side-view image is less than 0.01 m. The resolution differences lead to different uncertainty of the resulting point accuracy. Therefore, it is expected that the resulting meshes can be heterogeneous in terms of their level of details.

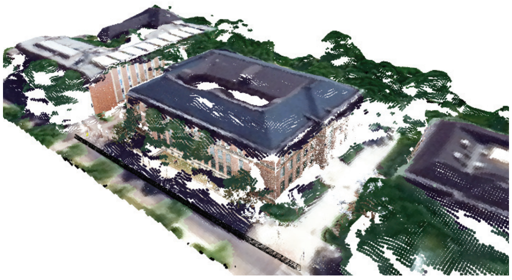


Figure 4. Co-registered street-view reconstructed point clouds and satellite reconstructed point clouds.

## 2.2 The Proposed Multi-source Mesh Reconstruction Pipeline

The base method (Labatut et al., 2009) takes the constructed Delaunay tetrahedra of the point clouds as the input to extract the surface. These tetrahedra can be viewed as a connected graph, in which the tetrahedra are the nodes and shared/common faces are edges. Our pipeline extends from this base algorithm by incorporating point clouds generated from the satellite images.

In a multi-view stereo scenario, each point in 3D space can be determined by at least two rays, which connect the object points and camera centers, and these rays indicate the visibility of this object point on the image connected to it. Based on ray visibility, tetrahedra intersected with rays are evaluated by their probability to be in a free space (outer space), and tetrahedra behind the ray endpoint are evaluated by their probability belonging to the full space (inner space). Labatut (2009)'s work minimized the *s-t* cut approach to solve the labeling problem. The final surfaces are the common faces of the tetrahedra labeled as free and full spaces (Figure 5).

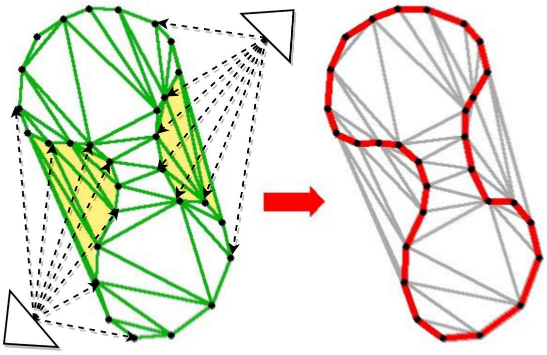


Figure 5. Left: Green network is Delaunay triangulation, yellow region (free space) are tetrahedra which intersected with rays (dash arrows), white region is tetrahedra labelled as full space. Right: Red lines are surfaces between full and free space, which are common faces shared by those tetrahedra.

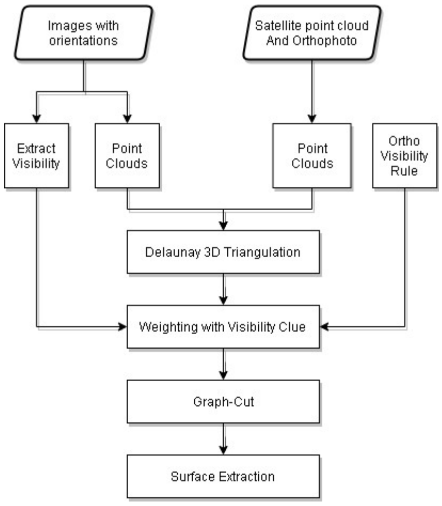


Figure 6. Our pipeline of multi-source mesh reconstruction

Figure 6 shows our proposed pipeline that meshes point clouds generated from street-view images and satellite point clouds. Firstly, we form Delaunay tetrahedra with the combined point set. Next, rays are created from visibility clue through triangulated points from dense matching and orthographic projection of satellite orthophoto to weight the tetrahedra. Finally, solve the minimum *s-t* cut for labeling problem and extract surface.

## 2.3 Delaunay 3D Triangulation

3D Triangulation or tetrahedralization is extended from 2D triangulation, which partitions a polyhedron into non-overlapping basic 3D elements, where the vertices of tetrahedra take the vertices of the original polyhedron. Delaunay tetrahedron reconstruction (Van Kreveld et al., 2000) divides the convex hull of points into compact simplices, where neither extremely long edge nor extremely sharp angle is included. Many well-known commercial packages and open source projects have implemented the algorithm that creates Delaunay tetrahedron from point set, here we use CGAL (Fabri and Pion, 2009) an open source computational geometric algorithm library to construct tetrahedra.

## 2.4 Visibility

Dense points associated with registered images are the most common source of visibility in our approach. To setup a minimum *s-t* cut problem, we first set the tetrahedra covering camera centers (start point of rays) to be free space and assign *s* label to them. We further initialize the tetrahedron behind object points as full, assign *t* label to them. As for tetrahedra that intersect with visibility rays while not belonging to any cases mentioned above, we assign weights to the edges of the graph (being the faces shared by tetrahedra) as smooth terms. The aforementioned algorithm was implemented by an open-sourced project OpenMVS (Cernea, 2015).

Ideally, we should implement the satellite camera sensor models (e.g. Rational Polynomial Coefficients) (Qin, 2016) to record the visibility information, however that could painstakingly trivial and complex. By considering that the point clouds can be associated with the orthophoto through a parallel projection, we then propose a two-step method to implement this strategy:

1. We first project the satellite points on to a grid, and in this process, for each planimetric location, we only mark the highest points as being visible.

2. Create vertical rays for these visible points.

## 2.5 Assigning Weights for the Graph

Section 2.3 introduces the constructed graph for the surface reconstruction problem. The strategy of weighting the edges considers the visibilities. The idea is intuitive: if a face has been traversed by many rays, it is unlikely to be a surface as the points are all behind this surface (which the ray is shooting to from the camera center). In this process, the confidence value of the triangulated points, if available, can also be incorporated when assigning the weights, otherwise, we keep it as a constant, denote as $\alpha_{vis}$.

Our method follows a so-called soft visibility weighting model that was used by the base algorithm (Labatut et al., 2009): As shown in Figure 7, the left-most tetrahedron which contains camera, gets a scaled $\alpha_{vis}$ link to the source, as discussed in Section 2.2. The green faces that passed through by the ray get scaled $\alpha_{vis}$, and the scale factor serves as the penalty that computed by its distance to the object points, being, the closer the face is to the object point, the less trusty of the ray. As shown by Figure 7, the yellow curve in labelled with oriented facet weights represent scaled $\alpha_{vis}$ along the camera-point ray. The right-most tetrahedron gets scaled $\alpha_{vis}$ and is link to sink.

Once weighting procedure for the edges is done, we use IBFS (Incremental Breadth First Search) (Goldberg et al., 2011) maximum flow algorithm to solve minimum *s-t* cut problem. Finally, the common faces between source and sink tetrahedra are extracted to build up optimum surface model (Figure 8b).

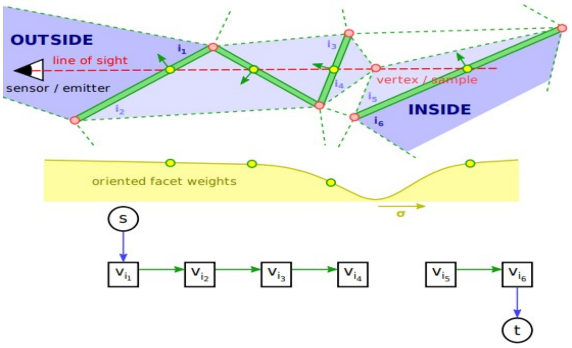
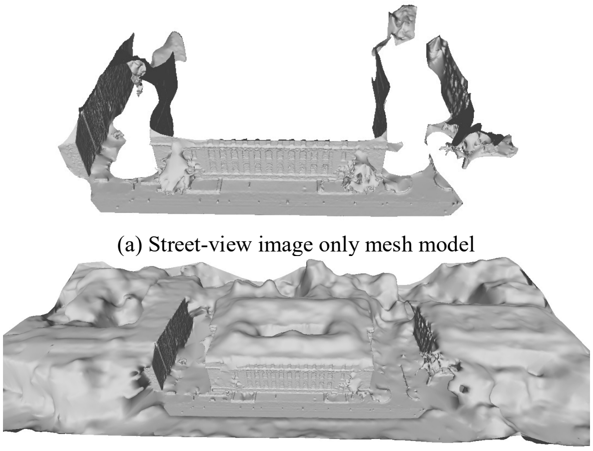


Figure 7. Soft visibility weighting model (Labatut et al., 2009)



(a) Street-view image only mesh model

(b) Mesh model reconstructed using multi-source data

Figure 8. Reconstructed surface model using only the street-view images (a), and surface model using both satellite and street-view images in our reconstruction method (b).

## 3. OPTIMIZING TEXTURE FUSION

### 3.1 The Proposed Multi-source Texture Mapping Pipeline

Our texture mapping framework is based on Waechter (2014)'s work which has been well practiced and widely used in many famous open source projects, e.g. TexRecon (Guthe, 2020), OpenMVS (Cernea, 2015), Open Drone Map (Dakota et al., 2017), etc. We consider the street-view images are perspective and the satellite orthophotos are in parallel projection. Our texture mapping considers the orthophoto as one of the images with only few simple modifications: we balanced data term of ortho images to compensate resolution gap and make ortho images as the default sources for texturing. Given these many images serving as potential source of texture, the first step is to pick a best image for each triangular face. Lempitsky and Ivanov (Lempitsky et al., 2006) compute a labeling $l$ that assign a view $l_i$ to be used as texture for each mesh face $F_i$ using a pairwise Markov random field energy formulation (Waechter et al., 2014):

$$E(l) = \sum_{F_i \in Faces} E_{data}(F_i, l_i) + \sum_{(F_i, F_j) \in Edges} E_{smooth}(F_i, F_j, l_i, l_j) \qquad (1)$$

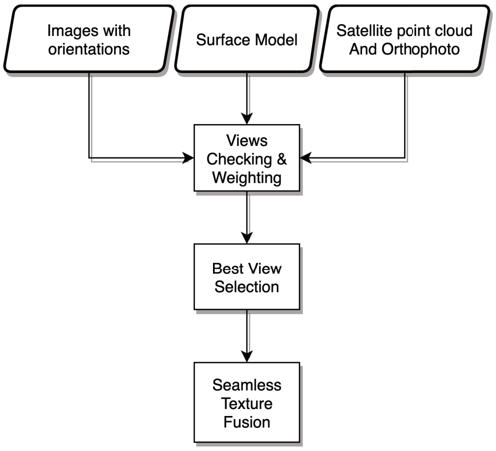
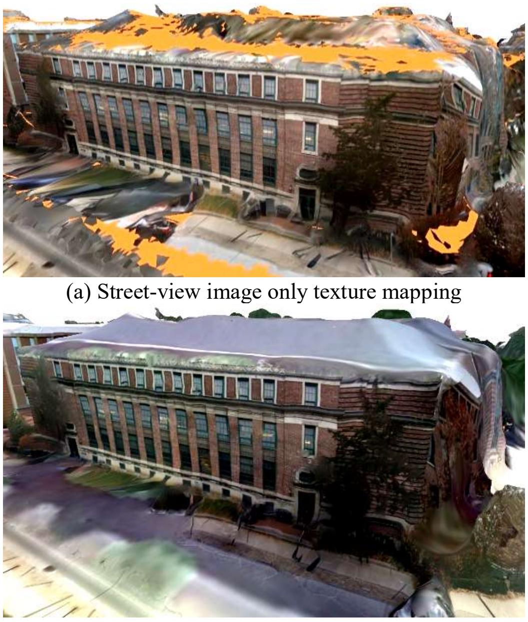


Figure 9. Our pipeline of multi-source texture mapping



(a) Street-view image only texture mapping

(b) Our multi-source texture mapping

Figure 10. Textured models using only the street-view images (a), and textured models using both satellite and street-view images in our texture mapping method (b).

### 3.2 Best-View selection

The base method (Waechter et al., 2014) determines face visibility (distinct from ray visibility) for all combinations of views and faces by first performing back face and view frustum culling, then renders faces onto images, using depth buffer to determine the nearest faces. This base approach takes projective projection as general camera model (Equation 2), but it did not assume images with parallel projections. In our work, we extend face visibility module to make it support orthographic projection (Equation 3) to adapt orthophoto.

Projective projection model:

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix} * \begin{bmatrix}r_{11} & r12 & r13 & t_1\\r_{21} & r22 & r23 & t_2\\r_{31} & r32 & r33 & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (2)$$

Orthographic projection model:

$$\begin{bmatrix}u\\v\\d\end{bmatrix} = \begin{bmatrix}s_x & 0 & c_x\\0 & s_y & c_y\\0 & 0 & 1\end{bmatrix} * \begin{bmatrix}r_{11} & r12 & r13 & t_1\\r_{21} & r22 & r23 & t_2\\r_{31} & r32 & r33 & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (3)$$

**Data Term:** The best view is defined as the closest, orthogonal image with highest-resolution and without imagery quality defects (e.g. blur). Gal et al. (2010) use the gradient magnitude of the image integrated over the face's projection, as Equation 4 shown. The data term $E_{data}(F_i, l_i)$ is assigned only if the given face $F_i$ is visible for street view image $l_i$. The visibility is checked by projecting to a given view and perform depth testing (or z-buffering) like graphics rendering system (Greene et al., 1993). In our case, the challenge is handling data from different sensors and distinct resolutions. First, the normalization factor $\alpha$ is introduced to balance the scale difference which theoretically is the ratio of average GSD of overview images and street view images. By adjusting $\alpha$ manually, we can control the preference between street view and orthophoto. The second challenge we met is sampling issue in z-buffering phase as each pixel in orthophoto might cover multiply triangular faces in practice because the mesh contains high resolution point clouds. Equation 5 shows the data term we used for orthophoto images, the $h_i$ denotes i-th ortho image. If the face is visible for ortho photo $h_i$ , the term will be computed by gradient magnitude; if neither street view nor ortho image can see the face $F_i$, every ortho photo $h_i$ will be assigned a constant value which is less than infinity (infinity is the trivial cost for unlabeled face). The data term (Equation 5) for orthophotos can make sure at least orthophotos are available to provide texture, which is a common practice in photogrammetry digital products generation (Zebedin et al., 2006). If there are more than one ortho images are provided, the smooth term will determine the which orthophoto to use based on face connectivity.

$$E_{data}(F_i, l_i) = -\int_{\phi(F_i,l_i)} ||\nabla(I_{l_i}(p))||_2 dp \quad (4)$$

$$E_{data}(F_i, h_i) = \begin{cases} -\alpha \int_{\phi(F_i,h_i)} ||\nabla(I_{h_i}(p))||_2 dp & \text{visible in ortho image} \\ Constant & \text{no visible image} \end{cases} \quad (5)$$

where $F_i$ = i-th triangular face
$l_i$ = index of i-th street view image
$h_i$ = index of i-th ortho image
$I_{l_i}$, $I_{h_i}$ = image of i-th street / ortho view
$Constant$= any constant value that less than infinity
$\alpha$ = average GSD of orthophoto divided by average GSD of the street-view images.

**Smoothness Term**: Waechter (2014) proposed a smoothness term based on Potts model (Wu, 1982): $E_{smooth} = [l_i \neq l_j]$, which works well and very fast.

### 3.3 Seamless Texture Fusion

Waechter (2014) proposed a global and local color adjustment method to blur the seams, which extended Lempitsky and Ivanov's (Lempitsky et al., 2006) color adjustment approach. The original approach only accounts for color difference on vertices to measure color difference along the seam line, called global adjustment. The extended method added a local adjustment with Poisson editing (Pérez et al., 2003), affect border strip of image patches.

In our case, since the resolution of orthophoto is way lower than the street-view images, prior to applying the fusion of image patches, we up-sampled orthophoto to the same resolution as that of the street-view images. After color balancing and Poisson editing, color differences can be well adjusted and seams are successfully been blurred, as shown in Figure 11.

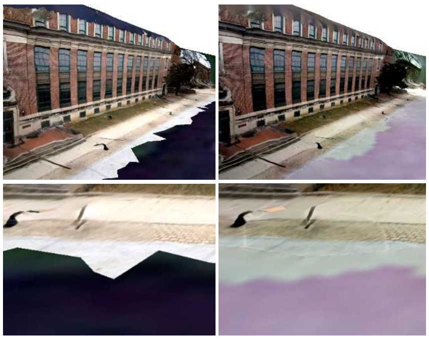


Figure 11. Seamless texture fusion with (right column) and without fusion (left column)

## 4. EVALUATION

We collected 6 World-View 2 images fully covering the campus region of The Ohio State University and processed with RSP satellite imagery stereo reconstruction package (Qin, 2016) The resulting DSM (or point clouds) and orthophoto cover 5 km x 3 km, with 0.5m GSD. As for close-range side view dataset, we extract a sequence of ~3000 images taken with GoPro camera mounted on a car. GoPro images are registered with state-of-the-art SFM algorithm (in-house developed approach, following standard SFM / photogrammetry pipeline (Moulon et al., 2016)), and the dense matching were performed using OpenMVS through multiple view 3d reconstruction algorithm (Shen, 2013). The datasets are then co-registered with the free SFM bundle with satellite DSM points (Figure 12) using approach described in (Qin, et al, 2020).

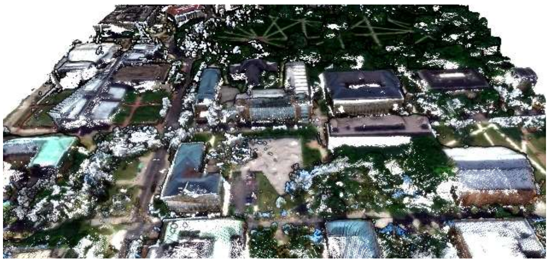
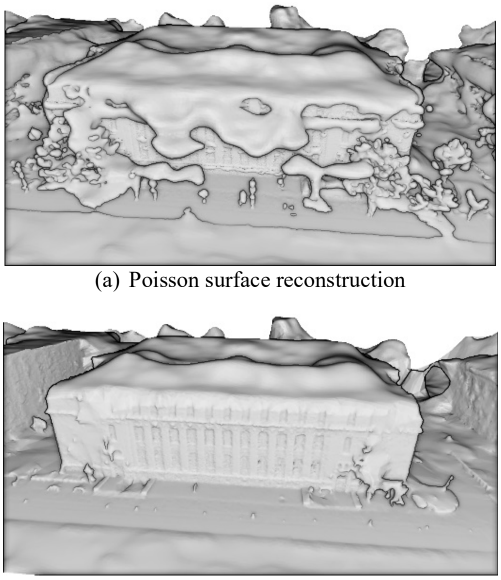


Figure 12. Co-registered street-view and satellite view dense match points.

As shown in Figure 13, our method outperforms Poisson reconstruction with our dataset, when processing dataset coming from multiple sources. Our method benefits from visibility information comparing with Poisson surface reconstruction (Kazhdan et al., 2006). Since Poisson reconstruction requires oriented points (with normal) but our dataset does not provide such information, we preprocessed the dataset with normal estimator (Mitra and Nguyen, 2003). The Poisson surface reconstruction we used is implemented in CloudCompare (Girardeau-Montaut, 2016), and the voxel size of the algorithm is configured to match with our average point distance.

Figure 14 shows results of the texture mapping method we proposed with our reconstructed mesh models. Our method can eliminate less confident vertices from satellite (e.g. tree crowns in Figure 13a), and create smooth, nice looking model with color adjustment.

Here we summarized run-time statistics and the model complexities of the generated model using different methods. As shown in Table 1, two methods have similar processing time, but it turns out our method can generate more concise model without appearance loss.


(a) Poisson surface reconstruction

(b) Our reconstruction method

Figure 13. A comparison of Poisson reconstruction and our method.

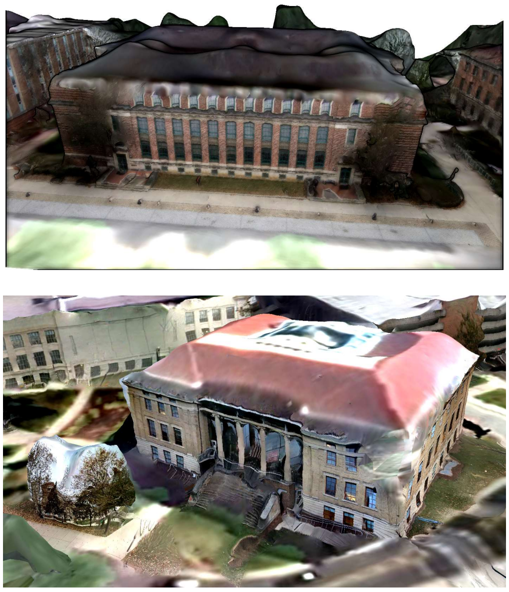


Figure 14. Our multi-source texture mapping method applied on the generated surface model.

| Model Name | Vertices | Faces | Usage |
|---|---|---|---|
| Ours #1 | 422,726 | 845,328 | 4 min |
| Poisson Recon #1 | 1,782,581 | 3,581,926 | 5 min |
| Ours #2 | 1,134,241 | 2,268,167 | 7 min |
| Poisson Recon #2 | 13,078,242 | 26,270,296 | 6 min |

Table 1. Number of vertices of the reconstructed mesh and the processing time

## 5. CONCLUSION

We present an approach to mesh heterogeneous 3D data by using Delaunay tetrahedralization based mesh reconstruction with ray visibility weighting. The surface reconstruction is cast as an energy minimization problem that can be globally optimized by solving minimum *s-t* cut. Our method recovers visibility from DSM data which acquired by matching satellite images and scaled to adapt to street-view images with orientations. It turns out that combining complementary data sources could enhance surface representative significantly if all information is well processed. The proposed multi-source meshing pipeline which extends state-of-the-art Delaunay tetrahedron based surface reconstruction method and multi-source texturing mapping pipeline which accommodates images acquired from different sensors (i.e. side-view perspective images and satellite images) can create large scale, coarse-fine fused, and seamless textured surface model.

The experiments show that our method creates visually consistent surface model from these two drastically different sources, as we as textures with well-balanced color although the source data are with very different radiometry (satellite images vs. street-view level images). We compared our proposed pipeline with a typical fusion pipeline - Poisson surface reconstruction and the results show that our pipeline shows distinctive advantages.

Our approach assumes that the satellite point clouds are associated with orthophotos, which does exclude the possible points from facades, as well as other scenarios in which different types of sensors from different views are used. Our planned future work includes methods that deal with 3D point clouds generated from other sources, such as MLS (Mobile Mapping System), ALS (Airborne LIDAR System).

## ACKNOWLEDGEMENT

This work is partially supported by the Office of Naval Research (Award No. N000141712928). The satellite datasets are provided by DigitalGlobe.

## REFERENCES

Barazzetti, L., Roncoroni, F., Brumana, R., Previtali, M., 2016. Georeferencing Accuracy Analysis of A Single Worldview-3 Image Collected Over Milan. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLI-B1, 429-434. doi.org/10.5194/isprsarchives-XLI-B1-429-2016

Antila, V. and Polet, J., 2011, September. ContextCapture: using context-based awareness cues to create narrative events for status updates. *In Proceedings of the 13th international conference on Ubiquitous computing* (pp. 585-586).

Cernea, D., 2015. OpenMVS: Open Multiple View Stereovision.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G., 2008. Meshlab: an open-source mesh processing tool, *Eurographics Italian chapter conference*, pp. 129-136.

Dakota, B., Fitzsimmons, S. and Toffanin, P., 2017. Open Drone Map.

Dantzig, G., Fulkerson, D.R., 2003. On the max flow min cut theorem of networks. *Linear inequalities and related systems* 38, 225-231.

Fabri, A. and Pion, S., 2009, November. CGAL: The computational geometry algorithms library. *In Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 538-539).

Gal, R., Wexler, Y., Ofek, E., Hoppe, H. and Cohen‐Or, D., 2010, May. Seamless montage for texturing models. *In Computer Graphics Forum* (Vol. 29, No. 2, pp. 479-486). Oxford, UK: Blackwell Publishing Ltd.

Goldberg, A.V., Hed, S., Kaplan, H., Tarjan, R.E., Werneck, R.F., 2011. Maximum Flows by Incremental Breadth-First Search. *Algorithms – ESA 2011*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 457-468.

Greene, N., Kass, M. and Miller, G., 1993, September. Hierarchical Z-buffer visibility. *In Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (pp. 231-238).

Guthe, S., 2020. TexRecon – Graphics, Capture and Massively Parallel Computing – Technische Universität Darmstadt, https://www.gcc.tu-darmstadt.de/home/proj/texrecon/ (5 February 2020).

Girardeau-Montaut, D., 2016. CloudCompare..

Jancosek, M., Pajdla, T., 2014. Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *International scholarly research notices 2014*. doi.org/10.1155/2014/798595

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction, *Proceedings of the fourth Eurographics symposium on Geometry processing*. Eurographics Association, Cagliari, Sardinia, Italy, pp. 61–70.

Labatut, P., Pons, J.-P., Keriven, R., 2009. Robust and Efficient Surface Reconstruction From Range Data. *Computer Graphics Forum 28*, 2275-2290. doi.org/10.1111/j.1467-8659.2009.01530.x

Landry, L. and Arbo, R., 2016. Vricon: The World in 3D.

Lempitsky, V., Boykov, Y., Ivanov, D., 2006. Oriented Visibility for Multiview Reconstruction. *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 226-238.

Mandanici, E., Girelli, V.A., Poluzzi, L., 2019. Metric Accuracy of Digital Elevation Models from WorldView-3 Stereo-Pairs in Urban Areas. *Remote Sensing* 11, 878.

Mitra, N.J. and Nguyen, A., 2003, June. Estimating surface normals in noisy point cloud data. *In Proceedings of the nineteenth annual symposium on Computational geometry* (pp. 322-328).

Moulon, P., Monasse, P., Perrot, R., Marlet, R., 2016. Openmvg: Open multiple view geometry, *International Workshop on Reproducible Research in Pattern Recognition*. Springer, pp. 60-74.

Pix4D, S.A., 2014. Pix4Dmapper..

Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing, *ACM SIGGRAPH 2003 Papers*. Association for Computing Machinery, San Diego, California, pp. 313–318.

Qin, R., 2016. Rpc stereo processor (rsp)–a software package for digital surface model and orthophoto generation from satellite stereo imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3, 77.

Qin, R., Lu, X., Song, S., Huang, X., 2020. 3D data generation using low-cost cross-view images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Shen, S., 2013. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE transactions on image processing* 22, 1901-1914.

Van Kreveld, M., Schwarzkopf, O., de Berg, M., Overmars, M., 2000. *Computational geometry algorithms and applications*. Springer.

Waechter, M., Moehrle, N., Goesele, M., 2014. Let There Be Color! Large-Scale Texturing of 3D Reconstructions. *Computer Vision – ECCV 2014*. Springer International Publishing, Cham, pp. 836-850.

Wu, F.Y., 1982. The Potts model. *Reviews of Modern Physics* 54, 235-268.

Zebedin, L., Klaus, A., Gruber-Geymayer, B. and Karner, K., 2006. Towards 3D map generation from digital aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6), pp.413-427.