Proceedings of the JURSE 2013, April 21-23, 2013 - São Paulo - Brazil

Model-free Dense Stereo Reconstruction for Creating Realistic 3D City Models

Georg Kuschk
Remote Sensing Institute
German Aerospace Center (DLR)
Wessling, Germany
Email: georg.kuschk@dlr.de

***Abstract*—In this paper we describe a framework for fully automatic and model-free generation of accurate and realistic 3D city models using multiple overlapping aerial images. The underlying DSM is computed by dense image matching, using a robustified Census transform as cost function. To further reduce the noise of mismatches, we afterwards minimize a global energy functional incorporating local smoothness constraints using variational methods. Due to the convexity of the framed problem, the solution is guaranteed to converge towards the global energy minimum and can be efficiently implemented on GPU using primal-dual algorithms. The resulting point cloud is then being triangulated, local planarity constraints are exploited to reduce the number of vertices and finally a multi-view texturing is applied. The quality of the DSM and the 3D Model is evaluated on a complex urban environment, using reference data generated by laser scanning (LiDAR).**

I. Introduction

Digital Surface Models (DSM) are the basic input for a wide range of applications like flood simulation, 3D change detection and radio beam propagation. Additionally for the normal end-consumer, 3D city visualizations are becoming more important for navigation every day. While all of these applications require a high accuracy, the 3D visualization systems additionally require the data to be of modest size, since they often operate with limited ressources (e.g. web-based applications or navigation devices). In this paper we describe a framework which achieves a good trade-off between high accuracy and small data size.

We start with Section II-A, describing the creation of an efficient epipolar geometry between the input images, and use the results for the computation of the cost function in Section II-B. Cost functions in dense stereo matching need to be descriptive and robust on the one hand (e.g. DAISY [7]), and easy to compute on the other hand (e.g. Absolute Difference). The Census transform [8] proved to be a good trade-off between these requirements for remote sensing imagery and is used by us in a robustified version.

Since the raw matching costs are still prone to mismatches and noise, we have to apply an additional regularization, forcing the disparity map to be locally smooth. A common choice is the well-established Semi Global Matching [2], which approximates a truly global optimization by combining different one-dimensional cost aggregations.

However, framing the stereo problem as a convex variational problem [5] and optimize it globally offers some advantages: Being theoretically well-founded, it guarantees to converge towards the global energy minimum. Its implementation being a simple and general framework, adding different image cues or constraints is quite straight forward. Furthermore, it can be efficiently accelerated on modern parallel GPU architecture, which makes it computational appealing again, as it scales directly with the number of parallel processors available (see Section II-C). To finally generate visually appealing 3D models with a low number of polygons needed, we describe our meshing, mesh simplification and multi-view texturing approach in Section II-D and II-E. The quality of the produced DSMs and 3D modelling is finally evaluated in Section III.

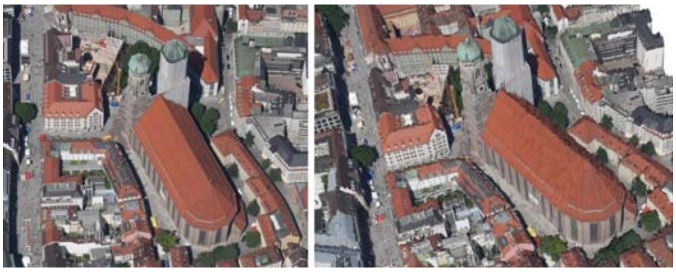

Fig. 1. Left: Example aerial input image. Right: An artificially rendered view of the reconstructed 3D model.

II. Method

Let the image space of a reference image I_1 be denoted as $\Omega \subset \mathbb{R}^2$. For every pixel $\mathbf{x} = (x, y)^T \in \Omega$ and every depth hypothesis $\gamma \in \Gamma = [\gamma_{min}, \gamma_{max}]$, we compute a matching cost $\rho(\mathbf{x}, \gamma)$ with respect to a second image I_2. The matching cost function is defined as

$$\rho(\mathbf{x}, \gamma) = |\ C(I_1, \mathbf{x}) - C(I_2, F_{(1,2)}(\mathbf{x}, \gamma))\ | \qquad (1)$$

with C being an arbitrary local image descriptor (see Section II-B) and $F_{(1,2)}$ a function projecting the pixel $\mathbf{x}$ in I_1 to image I_2 by using the disparity γ (see Section II-A).

In the resulting *disparity space image* (DSI) we then search for a functional $u(\mathbf{x})$ (the disparity map), which minimizes the energy function arising from the data term E_{data} and the

978-1-4799-0211-8/13/$31.00 ©2013 IEEE

additional smoothness constraints E_{smooth}

$$\begin{aligned} u(\mathbf{x}) &= \underset{u}{\operatorname{argmin}} \left\{ \int_\Omega E_{data} + E_{smooth} \; d\mathbf{x} \right\} \qquad (2) \\ &= \underset{u}{\operatorname{argmin}} \left\{ \int_\Omega \rho(\mathbf{x}, u(\mathbf{x})) + h\left(\nabla(u(\mathbf{x}))\right) \; d\mathbf{x} \right\} \end{aligned}$$

This energy is non-trivial to solve, since the smoothness constraints (implied by the function h) are based on gradients of the disparity map and therefore cannot be optimized pixelwise anymore. In Section II-C we go into detail about our choice of the optimization problem and how to solve it.

A. Epipolar Geometry

In case of multi-image matching, where the images can be arranged arbitrarily, pairwise rectification is cumbersome to implement and introduces additional numerical inaccuries. Also, for satellite images and the corresponding Rational Polynomial Camera (RPC) model, the epipolar lines of an image pair are not straight, but curved [4], increasing the complexity of an image rectification approximation.

We therefore establish the epipolar geometry between two images I_1 and I_2 directly by evaluating the function $F_{(1,2)}(\mathbf{x}, \gamma)$, which projects a pixel $\mathbf{x}$ from I_1 to I_2 using the disparity γ, for a sparse set of grid points in $\Omega \times \Gamma$ space. For all other points we interpolate the projected pixel coordinates by using trilinear interpolation. The lookup-table L is increased and refined iteratively until the reprojection error of the in-between grid points is < 0.001 pixel. To furthermore reduce the need for rotational invariant cost functions we apply a fast plane-sweep approach by warping image I_2 for each disparity $\gamma \in \Gamma$ into the coordinate system of image I_1 and evaluate the cost function at the same position (x, y), using the same local support window, in both I_1 and I_2.

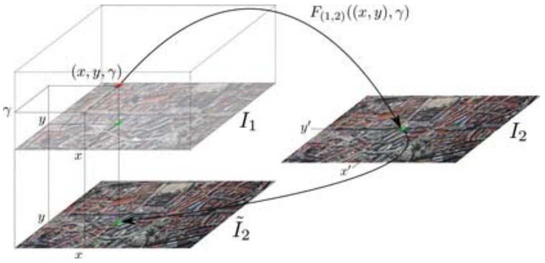

Fig. 2. Plane-sweep based warping of image I_2 into the coordinate system of image I_1 using a disparity γ.

B. Cost Function

The Census transform CT as described in [8] is a non-parametric transform which encodes the local image structure within a small patch around a given pixel. It is defined as an ordered set of comparisons of intensity differences and therefore invariant to monotonic transformations which preserve the local pixel intensity order. Image matching is then performed by comparing the resulting vectors at different image positions. However, the Census transform strongly depends on the center pixel and a slight variation of its intensity can cause the descriptor to vary significantly. We adress this issue by using the following (robustified) modification of the Census transform

$$MCT(I, \mathbf{x}) = \bigotimes_{[i,j] \in D \,\cup\, [0,0]} \xi(\bar{I}(\mathbf{x}), I(\mathbf{x} + [i,j])) \;\; , \qquad (3)$$

where we replaced the intensity of the center pixel by a weighted average of the intensities in its direct neighborhood (see Figure 3). The matching cost of different Census vectors s_1, s_2 is then computed as their Hamming distance $d_H(s_1, s_2)$ and is scaled to the real-valued interval $[0, 1]$

$$\rho_C(\mathbf{x}, \gamma) = \frac{d_H\left(\; MCT(I_1, \mathbf{x}),\; MCT(I_2, F_{(1,2)}(\mathbf{x}, \gamma)) \;\right)}{\max_{i,j}\{d_H(s_i, s_j)\}} \qquad (4)$$

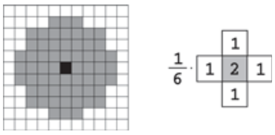

Fig. 3. The Census displacement field D and the weights for computing the center pixel intensity $\bar{I}(\mathbf{x})$.

C. Convex Optimization

To optimize Equation 2 globally, minimization of the Total Variation based on the L_1 norm (TV_{L_1}) proved to be a good regularizer in image applications due to its discontinuity preserving property [6]. With $h(\nabla u) = TV_{L_1}(\nabla u) = \int_\Omega |\nabla u(\mathbf{x})| \; d\mathbf{x}$, the stereo problem then becomes

$$u(\mathbf{x}) = \underset{u}{\operatorname{argmin}} \left\{ \int_\Omega \rho(\mathbf{x}, u(\mathbf{x})) + |\nabla(u(\mathbf{x}))| \; d\mathbf{x} \right\} \qquad (5)$$

whose minimization itself is hard to compute. Solving it, most algorithms depend on gradient descent, which often gets stuck in local minima and in general needs a very good initialization. To overcome these problems, [5] proposed to transform the general (non-convex) problem to a (higher dimensional) convex problem, whose solution is guaranteed to converge towards its global optimum. Additionally they developed an efficient numerical algorithm for solving this problem, by using a primal-dual algorithm. In the following we give a short outline of how to transform and frame the stereo problem of Equation 5 according to [5].

Forcing the energy of Equation 5 to be convex the original problem is lifted to a higher-dimensional space (from $\min_\Omega$ to $\min_{\Omega \times \Gamma}$) by representing u in terms of its superlevel sets: $\phi : [\Omega \times \Gamma] \rightarrow \{0, 1\}$, with

$$\phi(\mathbf{x}, \gamma) = \begin{cases} 1 & \text{if } \; u(\mathbf{x}) > \gamma \\ 0 & \text{otherwise} \end{cases}$$

where ϕ is an (initially) binary function, but in the following is allowed to vary smoothly in the interval $[0, 1]$. Together with the implied level-set properties, the set of feasible solutions for ϕ is given by

$$D = \{\phi : \Sigma \to \{0,1\} | \phi(\mathbf{x}, \gamma_{min}) = 1, \phi(\mathbf{x}, \gamma_{max}) = 0\} \quad (6)$$

using the short notation $\Sigma = [\Omega \times \Gamma]$. Now, the solution of Equation 5 can be formulated as

$$\min_{\phi \in D} \left\{ \int_{\Sigma} \rho(\mathbf{x}, \gamma) \cdot |\partial_\gamma \phi(\mathbf{x}, \gamma)| + |\nabla_2 \phi(\mathbf{x}, \gamma)| \, d\mathbf{x} d\gamma \right\} \quad (7)$$

and by using its dual formulation, we arrive at the primal-dual problem

$$\min_{\phi \in D} \left\{ \max_{\mathbf{p} \in C} \left\{ \int_{\Sigma} \nabla_3 \phi \cdot \mathbf{p} \, d\Sigma \right\} \right\} \quad (8)$$

with the set of feasible solutions in dual space constrained to

$$C = \Big\{ \mathbf{p} : \Sigma \to \mathbb{R}^3 \mid \sqrt{p_1(\mathbf{x}, \gamma)^2 + p_2(\mathbf{x}, \gamma)^2} \leq 1 \, , \quad |p_3(\mathbf{x}, \gamma)| \leq \rho(\mathbf{x}, \gamma) \; \Big\} \quad (9)$$

This problem can now finally be solved by alternatingly updating the primal and dual solution:

$$\phi^{k+1} = \mathcal{P}_D \left(\phi^k + \tau_p \cdot \mathrm{div}_3 \mathbf{p}^k \right) \quad (10)$$

$$\mathbf{p}^{k+1} = \mathcal{P}_C \left(\mathbf{p}^k + \tau_d \cdot w \cdot \nabla_2 \phi^{k+1} \right) \quad (11)$$

where τ_p and τ_d are the primal and dual step size, ∇_2 and div_3 are the divergence and gradient operators in the primal and dual space, $\mathcal{P}_D$ denotes the projection onto the set D (a simple truncation of ϕ^{k+1} to the interval $[0, 1]$) and $\mathcal{P}_C$ denotes the Euclidean projection onto the set C. In Equation 11, we introduced an additional weighting of the smoothness constraint of ϕ with a function $w(\mathbf{x})$ depending on the image gradient at position $\mathbf{x}$. The weighting function $w(\mathbf{x})$ is given by

$$w(\mathbf{x}) = e^{-\alpha \cdot ||\nabla_2 I_1(\mathbf{x})||_2} \quad , \quad w(\mathbf{x}) \in [0, 1] \quad (12)$$

resulting in large regularization weights for low intensity changes and small regularization weights for large intensity changes. The parameter α is only used for the normalization of the image gradient, and for 8Bit images is set to $1/255$.

The iterative primal-dual algorithm is stopped if the energy of Equation 5 does not change much anymore and the function u can be recovered from the final level sets ϕ, by summing them up for each pixel. After obtaining the disparity maps (one per image pair), we project them to UTM coordinates, merge them using a median filter and and interpolate the missing data (resulting from the outlier removal and the projection itself).

D. Meshing and Simplification

Since our resulting DSM is a dense 2.5D representation of the scene on a regular grid, we can create a mesh by simply connecting the 4 incident vertices of a square into two triangles. Of the two possible triangulations (Figure 4), we adaptively choose the one which minimizes the second derivative of the height surface in the neighborhood of the square, as proposed in [1]. This is done by computing the sum of the second derivative along two line segments for each of the two choices of the diagonals w.r.t. the height information

$$\begin{aligned} d_1 &= |v_{20} - 2v_{11} + v_{02}| \; + \; |v_{31} - 2v_{22} + v_{13}| \\ d_2 &= |v_{23} - 2v_{12} + v_{01}| \; + \; |v_{32} - 2v_{21} + v_{10}| \quad . \end{aligned} \quad (13)$$

If $d_1 < d_2$, the square $(v_{11}, v_{12}, v_{21}, v_{22})$ is triangulated using the diagonal (v_{21}, v_{12}), otherwise by the diagonal (v_{11}, v_{22}).

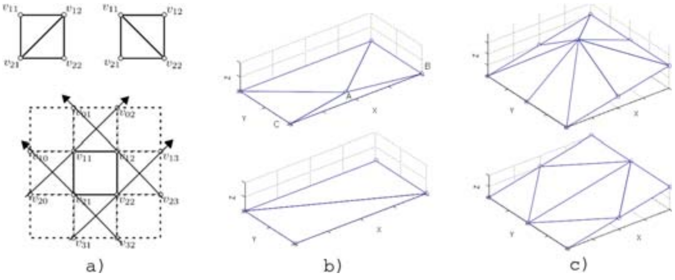

Fig. 4. a) The two possible triangulations of a square and the vertices involved in computing the optimal diagonal, b) Planar mesh simplification, c) Collinear mesh simplification

Because the DSMs are represented by one height value per pixel, meshing and texturing for visualization purposes is not practical using such a dense 3D model. We therefore have to reduce the amount of triangles needed to represent the 3D model, while at the same time preserving its dominant features and surface properties.

In a first step, we simplify planar structures by iterating over all vertices and fit a 3D plane through its neighbors using least squares method. If the minimum distance of the vertex to the fitted plane is $< \Delta_{plan}$, the vertex is removed (see Figure 4). As this would sometimes remove the corners of buildings, we add a further constraint that the vertex gets only removed, if the height difference to all of its adjacent vertices is $< \Delta_{disc}$. These two parameters depend on the grid resolution δ of the DSM and are set to $\Delta_{plan} = \delta$ and $\Delta_{disc} = 10\delta$.

The second step of our mesh simplification is removing nearly collinear triangles. If for any triangle (A, B, C), $\overline{AB} + \overline{AC} < \overline{BC} \cdot \Delta_{coll}$ (with $\Delta_{coll} > 1$) the vertex A will be removed. We chose to remove only very collinear triangles ($\Delta_{coll} = 1.01$).

E. Multi-view texturing

When aiming for a natural looking 3D model, we have to assign 2D texture coordinates to the corners of each triangle. Images of the scene taken from different viewpoints allow us to extract the texture of parts of the scene hidden from a single view, like for example the facades of buildings (see Figure 5). In that case we have to devise a quality measure Q for the projection $\pi(t_i, I_k)$ of a triangle t_i into each image I_k available for texturing. Of all these K projections, we then choose the one with the best quality measure for texturing the triangle t_i

$$k = \underset{k}{\mathrm{argmax}} \, \{ \; Q(\; \pi(t_i, I_k) \;) \; \} \quad (14)$$

Since our image data was taken from roughly the same distance to the scene, we choose the image I_k for texturing

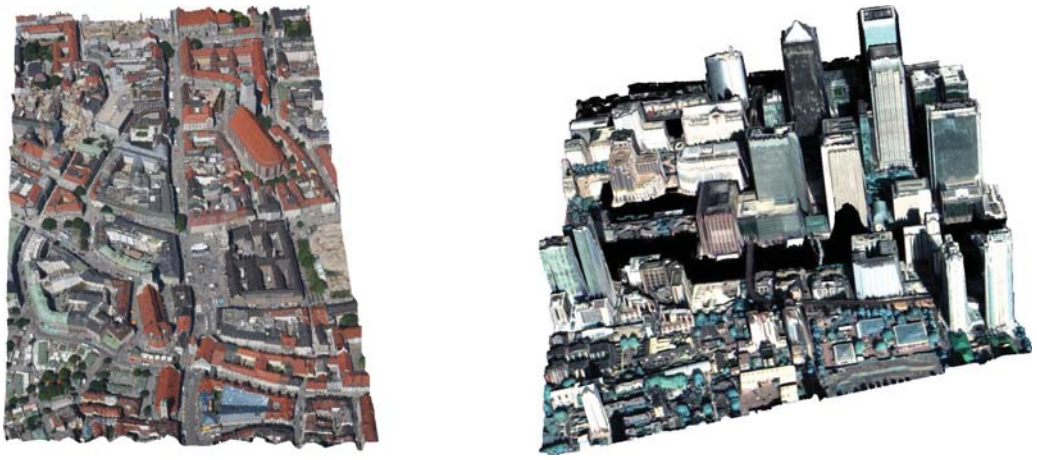

Fig. 5. Textured 3D visualization of the test areas. Left: Munich obtained by aerial 3K+ camera, Right: London obtained by WorldView-2.

a triangle t_i, where the 2D projection of t_i has maximal size (to capture fine details) and is least occluded by other triangles (especially important for urban areas containing large buildings and narrow streets). The texturing workflow is then:

- Sort all 3D triangles of the model according to the distance to the camera center (z-buffering)
- Project the triangles onto the image plane and render them using a unique identifier
- Sweep over the rendered image and compute the quality of each triangle in term of its remaining visible pixels
- Assign texture coordinates for each triangle from the corresponding best input image

III. Results

Evaluation of the proposed algorithms on remote sensing images is done on an aerial image set of the inner city of Munich, obtained by the 3K+ camera system [3] and on satellite images from the inner city of London. For both test areas we have reference data obtained by airborne laser scanning (LiDAR) at hand. Due to the different resolution of the DSMs as well as the LiDAR point cloud, we compute the error metrics as Euclidean distance between the points in the reference data to the triangulated full DSM or simplified 3D model (see Table I). Additionally we show the textured 3D visualization of both test areas in Figure 5).

IV. Conclusion

It has been shown that using the robustified Census transform together with the convex optimization of the Total Variation TV_{L1} produces accurate DSMs for remote sensing images. For creating visually appealing 3D city models, a model-free approach for meshing, simplifying and multi-view texturing was presented. In future work, we will focus on incorporating additional image cues like edges and planar structures into the optimization framework.

Acknowledgment

The author would like to thank European Space Imaging (EUSI) for providing the WorldView-2 data of London.

TABLE I

Data properties and accuracy of the proposed algorithms - Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Normalized Median Absolute Deviation (NMAD)

	Munich 3K+	London WV2
Area [m]	750×450	800×800
GSD [m]	0.2	0.5
Area [pixel]	$8.4 \cdot 10^6$	$2.5 \cdot 10^6$
Vertices in 3D model	$246,000$	$155,000$
Vertices / m^2	0.73	0.24
Vertices / pixel	0.03	0.06
DSM - MAE [m]	0.71	1.17
DSM - RMSE [m]	1.45	2.07
DSM - NMAD [m]	0.56	0.77
3D Model - MAE [m]	1.12	2.10
3D Model - RMSE [m]	2.09	2.88
3D Model - NMAD [m]	0.71	1.84

References

[1] M. Grabner. Compressed adaptive multiresolution encoding. *Journal of WSCG*, 10(1):195–202, 2002.

[2] H. Hirschmueller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.

[3] F. Kurz, R. Mueller, M. Stephani, P. Reinartz, and M. Schroeder. Calibration of a wide-angle digital camera system for near real time scenarios. In *Proc. ISPRS Hannover Workshop 2007-High Resolution Earth Imaging for Geospatial Information*, pages 1682–1777, 2007.

[4] J. Oh. *Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images*. PhD thesis, The Ohio State University, 2011.

[5] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. *Computer Vision - ECCV 2008*, pages 792–805, 2008.

[6] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[7] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, pages 815–830, 2009.

[8] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. *Computer Vision - ECCV 1994*, pages 151–158, 1994. census transform, rank transform.