This ICCV Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

3D Texturing From Multi-date Satellite Images

Marie d'Autume, Enric Meinhardt-Llopis
CMLA, ENS Paris-Saclay, France
marie.de-masson-d-autume@ens-paris-saclay.fr

Abstract

This paper addresses the problem of point cloud texturation of urban areas from multiple satellite images. Our algorithm is well-suited for the case where the images are obtained from different dates, where the dynamic ranges are incompatible and the position of the shadows is different. The method relies on a robust, PDE-based, fusion of the multiple candidate textures for each surface patch, and optionally on a geometric criterion to remove the shadows of the known objects. We showcase the results by building a 3D model of some areas of Boulogne Sur Mer (Argentine) such that all facades are correctly textured, with uniform colours and without shadows, even if for each individual input image only one side of the buildings was visible.

1. Introduction

We propose a method to assign a texture to a given digital elevation model from several satellite images of the same site. The main novelty of the proposed method is its conceptual simplicity: the texture at each point is obtained by a weighted fusion of *all* the available images instead of a stitched piece-wise selection of the best image [27]. The smooth weights assure seamless transitions between the different parts of the texture. Instead of merging the colours, we use an illumination invariant feature, its drift field [28, 5], so that images of very different dynamic ranges are combined correctly together. Since the position of the shadows is typically different at each image it is important to be able to obtain a shadow-less final texture, they disappear after the robust fusion based on the geometric median.

1.1. Previous work

In the last decade extensive research has been done in the computer vision community on 3D reconstruction of large-scale surfaces from multi-view images. For ground and aerial images auto-calibration, Structure-from-Motion (SfM) and Multi-view stereo techniques [20, 25, 29] can recover with impressive accuracy the geometry for many

Figure 1: Top: detail of the Lidar provided in the IARPA dataset. Bottom: textured mesh obtained using the Lidar and the 47 satellite images of the dataset. The texturation adds a new dimension and allows a better understanding of the reconstructed area.

problems. Texturation, the final step of the reconstruction, has seemed neglected at first but recent years have seen a steady increase of publications on this topic, in particular since the paper of Waechter et al. [27].

Most texturation methods nowadays use high-resolution texture atlases [4, 26, 13] and coarse meshes. We chose to use per-vertex colours defined on a fine mesh of the same resolution as the input panchromatic images. This method is halfway between classic texturation with atlases and the colourisation of point clouds [19, 18]. Our choice is made possible by the relatively low resolution of satellite images compared to aerial and terrestrial images and allows for a higher flexibility for the fusion, namely using PDE (Partial Differential Equations).

Of particular interest among large-scale surfaces are ur-

ban areas [1]. Higher quality smartphone cameras and the growth of photo-sharing websites has lead to a large number of various views of famous buildings and can allow high quality reconstruction. Terrestrial mobile mapping systems (MMS) are widely used to obtain street-view data of whole cities. They offer images acquired simultaneously ensuring a texture with no seams and global colour adjustment [3]. The development of aerial oblique photogrammetry based on aircraft or unmanned aerial vehicles also provide new venues of exploration for successful urban areas 3D modelling [21]. Actually both are complementary as they offer different view points which is useful to reconstruct both roofs and building facades. [30].

Another important source of input images for multi-view stereo reconstruction of building areas is optical satellite imagery [2, 11, 14]. Digital Surface Models (DSMs) obtained from these multi-view stereo algorithms are useful in particular for the automatic detection of elevation changes [17, 16]. However little work has been done on texturation using satellite images and most of it for large-scale terrain reconstruction [24]. In the following we tackle the problem of urban areas texturation using multi-date Worldview3 images and show that current satellite camera resolution (panchromatic: 0.31m and multi-spectral: 1.24m nadir) can already provide interesting results.

1.2. Specifics of satellite images

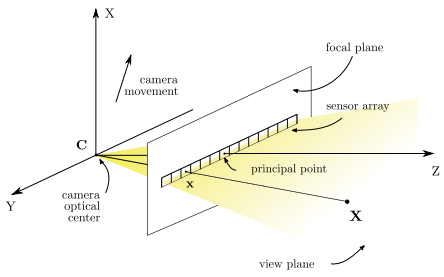


Figure 2: Linear push-broom camera model. The instantaneous camera optical centre C moves at a constant speed. The view plane sweeps out the whole 3-space as the camera moves forward. The 3D point $\mathbf{X}$ is imaged by the camera to the focal plane location x [6, 20].

Texturation using multi-date satellite images presents several specific difficulties. In general 3D reconstructions problems, the first step is to estimate camera parameters. Typically the internal (focal length, pixel width and principal point) and external parameters (position and orientation) are estimated using auto-calibration and Structure-from-Motion. While aerial or terrestrial imagery is obtained by a projective camera, satellite images are acquired by linear push broom sensors. The sensor sweeps a region of space capturing a single line at a time, hence its name. Thus in the direction of the sensor motion the image is an orthographic projection, preventing the use of most 3D reconstruction pipelines

Instead satellite images metadata provide us with a lot of information. On modern high resolution satellite we can assume that the internal parameters are perfectly known, having been precisely calibrated before launch. The external parameters are measured on board in real time and given with the images but a pointing error still remains. For Worldview 3 satellite the geolocation accuracy is inferior to 3.5m, an error that remains too large for the texturation without a registration step.

Figure 3: Crops of some of the input images. Here we can see colour variations, saturation, various cast shadows and states of the trees depending on the season and on the bottom right image the blur resulting from a cirrus.

Another important difficulty comes from working with multi-date images. In aerial views, shadows are often be avoided by flying under clouds so that the shadows are diffuse; while for satellite images only cloudless views are exploitable. Nowadays, commercial satellites are designed to take picture of a zone always around the same time, around 2pm in our dataset. Thus the shadows are always attached to the same side of the objects, only the cast shadows change direction with the seasons. Few oblique views show the shadowed side of the surface and the shadows in satellite imagery are very dark and noisy. In one direction the texturation always fail.

Reflections, on the roof or the windows in particular, lead to saturation, and the images have very different and high dynamic range due to the seasons and the cirrus cover. Also these various atmospheric conditions lead to very different colours that necessitate a global colour adjustment. See Figure 3.

Finally very slanted surface with respect to the point of view such as building facades are challenging and some blurring cannot be avoided.

2. IARPA dataset and S2P algorithm

For our experiments, we used the public benchmark dataset for multiple view stereo mapping using multi-date satellite images. This dataset, which supported the IARPA Multi-View Stereo 3D Mapping Challenge, includes 47 DigitalGlobe WorldView-3 images of a 100 square kilometre area near San Fernando, Argentina [2] (see Figure 4). The images were acquired over a period of 14 months. Most of them were taken at different dates. Nearly all the images are clear sky. However, the quality is not consistent: the winter images are considerably noisier, and the images with large incidence angles suffer from a loss of resolution in the range direction. The dataset also includes 30 cm resolution airborne lidar ground truth for a 20 square kilometre subset of the covered area.

Each satellite image is provided with a Rational Polynomial Coefficients (RPC) camera model [10], and other metadata such as the exact acquisition date or the direction of the sun. The RPC model combines the internal and external parameters of the push-broom system in a pair of rational polynomial functions that approximate the mapping from 3D space points given as (latitude, longitude, height) to 2D image pixels $(i, j) = P(\lambda, \mu, h)$ (named *projection*), and its inverse $L : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3$, *localisation* (see Figure 6. Both rational functions have degree 3 (for a total of 160 coefficients per image). Figure 6.

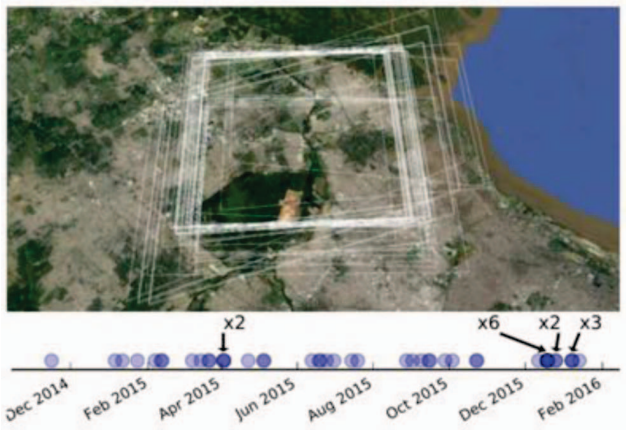


Figure 4: Footprints and dates of the 47 images of the IARPA challenge dataset [2]. The images cover the North part of Buenos Aires and were acquired over a period of 14 months. Only four groups of images were taken during the same orbit: two pairs, one triplet, one sextuplet.

2.1. Geolocation error

Although the RPC are accurate, the model they encode is subject to measurement errors which translate into geopositioning errors of the localised points. For high-resolution satellite images one can assume that the internal parameters are known perfectly and that the external parameters are known with a high precision.

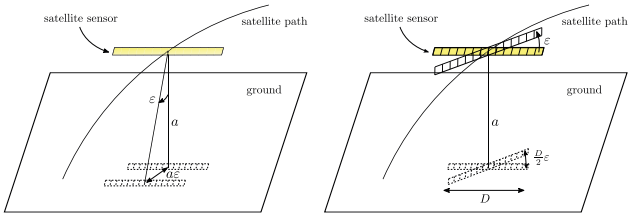


Figure 5: Effect of attitude errors on the localisation function. The figure on the left shows a pitch error of ε. To first order, this induces a ground displacement of the sensor projection of $a\varepsilon$, where a is the flying altitude of the sensor. The figure on the right shows a yaw error of ε. It induces a maximal ground displacement of $\frac{D}{2}\varepsilon$, obtained for the sensor endpoints, where D denotes the swath width [6].

An error of 1cm of the camera centre position results in an error of at most 1cm on the ground and is negligible at our working resolution. Errors relative to the sensor orientation however have more serious consequences. This orientation is computed from external parameters, the attitude parameters: yaw, roll and pitch. As illustrated in Figure 5 while a yaw error has little effect, a pitch or a roll error is much more problematic. Indeed for a satellite orbiting at 600km, a small error of a μrad on the roll or pitch leads to a displacement of about 60cm on the ground. This is the main cause of geolocation inaccuracy. Locally this error can be approximated as a 3D translation of the focal plane for scenes of size up to 50×50 km [6]. The pointing errors can be of the order of tens of pixels in the image domain.

The geolocation error can be highlighted using the output DSMs of the Satellite Stereo Pipeline (S2P) [1] [8, 7, 11]. The S2P pipeline takes as input a pair of satellite images A and B and its associated RPC and gives as output a 3D point cloud. This point cloud is then projected on a geographic grid with the same resolution as the satellite nadir GSD (ground sampling distance). Both input images have different pointing errors but the output DSM, denoted DSM^A in Figure 6, is coherent with the first input image with respect to geolocation. To each pixel (i, j) of this DSM can be associated a 3D point with Universal Transverse Mercator (UTM) coordinates (e, n, z) by the function denoted $\varphi_2 : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3$ in Figure 6:

$$(e, n, z) = \varphi_2((i, j), \mathrm{DSM}^A(i, j)) \quad (1)$$

$$= (s_e^A i + o_e^A, s_n^A + o_n^A, \mathrm{DSM}^A(i, j)). \quad (2)$$

The scale (s_e^A, s_n^A) and the offset (o_e^A, o_n^A) needed to obtain the UTM coordinates are extracted from the metadata of the satellite image.

[1] https://github.com/cmla/s2p

It is thus possible to obtain for each satellite image of our dataset an associated DSM with the same geolocation error as the image by running S2P several time. Our method takes as input a reference DSM used to create a mesh, the satellite images with their metadata and the associated DSMs from S2P associated to each image. For the experiments presented here, the reference DSM is the Lidar.

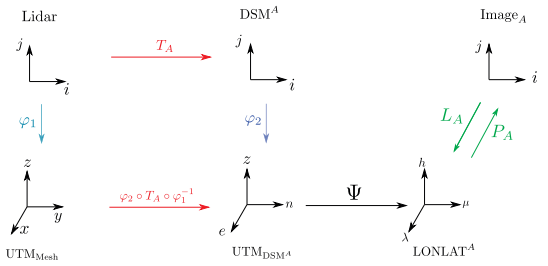


Figure 6: This figure illustrates all the coordinates system we use for the pipeline. The localisation and projection functions, L^A and P^A, are found in the metadata of the satellite image A. The functions φ_1 and φ_2 are the inverses of the functions used to project the georeferenced point clouds onto the DSMs. The vector T_A corresponds to the geolocation error. Ψ translates UTM coordinates into longitude-latitude.

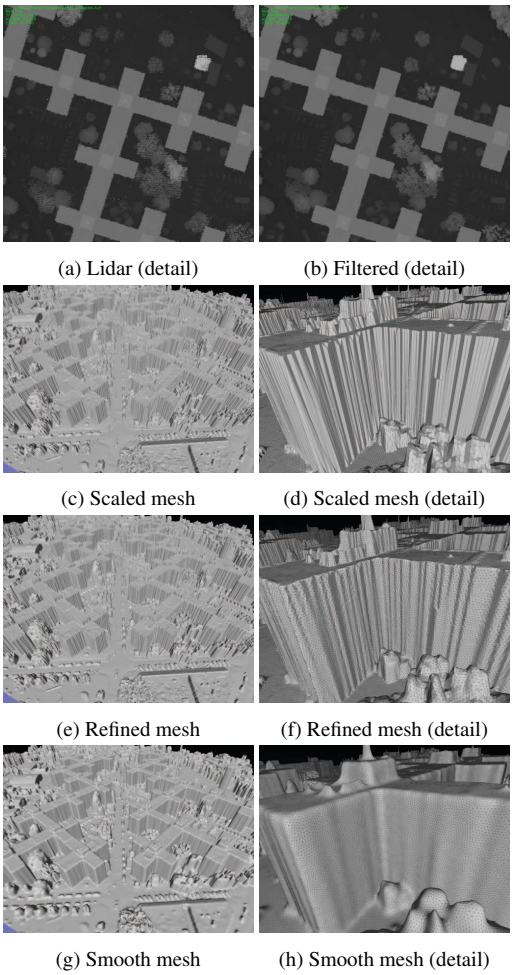
3. Texturation pipeline

In this section we detail the proposed pipeline to obtain a textured mesh. The input of the algorithm is the original set of N satellite images, and the associated set of DSMs computed by the S2P software (or a similar 3D reconstruction software). The output of the algorithm is a textured mesh.

The texturation has five steps, two of them optional. First, a high resolution mesh is created from a single DSM that is taken as reference (section 3.1). Second, all the datasets are accurately registered (section 3.2) to the reference one; this step is only necessary because S2P does not include a bundle adjustment, which is nontrivial for the degenerate satellite case. Third, the colors of each image are projected to the same reference 3D model, giving a set of N textures over the same mesh (section 3.3). Fourth, the shadows on each image are identified (section 3.4); this step is optional, only required if we want the "shadowless" images. Fifth and last, the N textures are merged into a single one (section 3.5).

3.1. Mesh Creation

The first step consists in creating a fine and watertight mesh from the reference DSM, which is a 2.5D model. As noted in [9], the problem of reconstructing facade geometry from a DSM is a challenge. At our working resolution however we can use a very basic approach. This is done in four steps illustrated in Figure 7:



(a) Lidar (detail) (b) Filtered (detail)

(c) Scaled mesh (d) Scaled mesh (detail)

(e) Refined mesh (f) Refined mesh (detail)

(g) Smooth mesh (h) Smooth mesh (detail)

Figure 7: Steps described in Subsection 3.1 for the creation of a smooth mesh from a reference DSM.

First the reference DSM is filtered to eliminate aberrant points, smooth building facade and fill in the trees in the particular case of a Lidar. This step is done by median filtering. A intermediary mesh is created by triangulation, each vertex corresponding to a pixel of the filtered reference DSM. The vertices have UTM coordinates in meter obtained by the function called $\varphi_1 : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3$ in Figure 6.

$$(e, n, z) = \varphi_1((i, j), \mathrm{DSM}^{ref}(i, j)) \quad (3)$$

$$= (s_e i + o_e, s_n + o_n, \mathrm{DSM}^{ref}(i, j)). \quad (4)$$

The scale and offset are obtained from the metadata of the reference DSM. This leads to elongated triangles on the facades with a zigzag pattern in respect to their orientation. The mesh is then refined using the CGAL library [22] so that each point all vertices are equidistant. The length of the

edges is chosen a little smaller than the size of a panchromatic pixel. Finally a Laplacian smoothing (using the umbrella operator) is applied to regularise the orientation of the faces. For this step we used the `trimesh2` library available online [2].

3.2. Correction of Geolocation errors

After the first step it, is already possible to project each satellite image on the mesh. However the geolocation error (explained in section 2.1) renders the results inconsistent as illustrated in Figure 8. The roofs are partly projected on the facades while some ground texture appears on the roofs. To mitigate this problem we need to find the 3D displacement vectors resulting from the geolocation errors inherent to each input image.

The S2P pipeline does not perform a bundle adjustment; instead, it selects a list of pairs of input images, computes a DSM for each pair independently, and then merges all the obtained DSM. Thus, each of these intermediate DSM is naturally registered to the first image of the corresponding pair. We chose the best DSM as reference, and then we compute the 3D translations that register all the other DSM to this one. Registering 3D models turns out to be more stable than registering image features, especially when there are a lot of shadows and occlusions, as is the case.

For the registration of the DSM, we use an algorithm performing subpixellic gradient phase correlation [15]. The goal of this algorithm was to register aerial and satellite images. As they are taken at different times, under distinct heights and sometimes even on different spectral bands it is a difficult problem. Gradient phase correlation is invariant to illumination changes, handles appearing and disappearing objects, large displacements, noise and compression artefacts and small intersections between images. When registering DSMs there are no noise or compression artefacts. However we still have to handle large displacements, appearing and disappearing objects. And the height translation can be visualised for this application as an illumination change.

After registration, we find that the error is generally less than 50cm on the ground. The more complete the reconstructed DSM, the more precise the translation parameters obtained. See Figure 8 for an illustration of this step. Now we can assume that each point of the mesh can be mapped precisely into any of the images from where it is visible. The final texture will be obtained by combining all these colours in a consistent way. At the end of this step, we are in possession of the vector T_A represented in Figure 6.

[2]https://gfx.cs.princeton.edu/proj/trimesh2/, Copyright ©2004-2018 Szymon Rusinkiewicz.

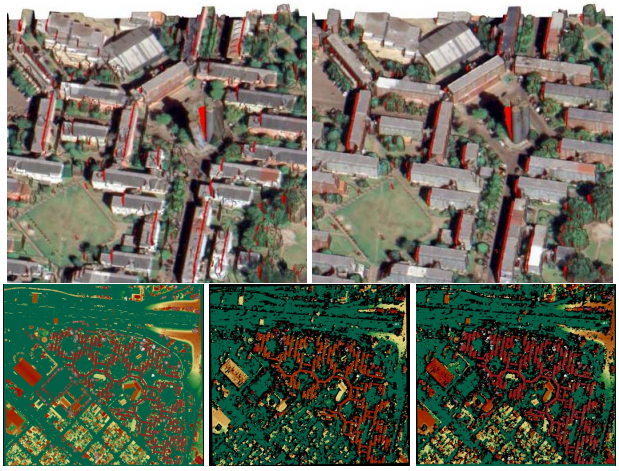


Figure 8: Top: Projection on the mesh obtained from the Lidar of image I before and after the registration step. Bottom: Lidar, DSM from S2P associated with image I before and after registration, all represented with the same colormap. The translation vector obtained from gradient correlation is $T_A = (-6.94985$ pix, -20.2493 pix,-3.06814 m).

3.3. Data projection

After the registration step, it is now possible to project accurately each vertex of the mesh onto the image plane, allowing to assign a color to this vertex. However not all of the vertices are actually visible by the camera. We now have all the data needed to project the mesh on the image plane. The image pixel (i, j) associated to a vertex with UTM coordinates (e, n, z) is:

$$(i, j) = P_A \circ \Psi \circ \varphi_2 \circ T_A \circ \varphi_1(e, n, z)). \qquad (5)$$

As illustrated on Figure 9, all points of the mesh on the same line of sight are projected on the same pixel in the image plane: the points A_i, B_i and C_i are all projected respectively on the pixels A, B and C of the image plane.

A classic tool of computer graphics, z-buffering [12], allows us to determine which of these vertices are really visible in the satellite image. In our case the z-buffer is easily built thanks to the fact as the camera is situated far above the objects considered, the point visible in the image is the highest one. The Figure 9 illustrates the z-buffer associated to a detail of one of a satellite image.

We define the normal to a vertex as the 3D average of the normals of the triangles to which it belongs.

For each vertex v of the mesh, and for each image, we store:

- $Sc_v = \cos(\vec{n}, \vec{n}_{\text{cam}})$: the scalar product of its normal $\vec{n}$ and the vector director of the line of sight (represented by $\vec{n}_{\text{cam}}$ in Figure 9),
- PAN_v: the intensity, using bicubic interpolation, from the panchromatic image if the vertex is visible, NaN

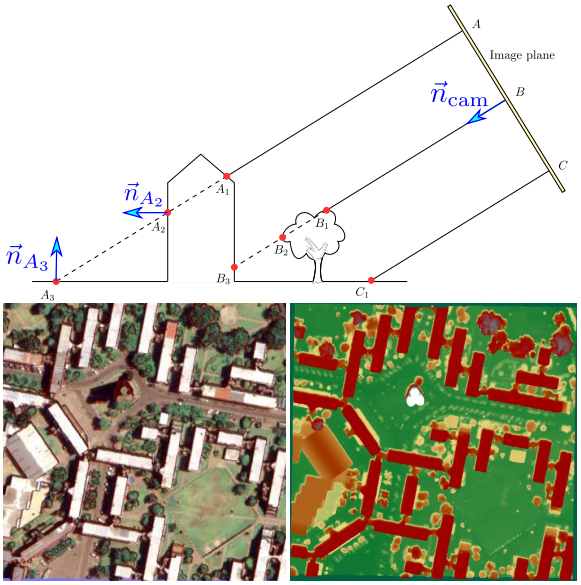


Figure 9: Top: sketch of the projection of the mesh onto the image plane. Bottom left: mesh textured from one image, seen from the camera viewpoint. Bottom right: elevation of each point seen by the camera on the image plane.

otherwise,

- MSI_v: the values from the multi-spectral image if the vertex is visible, using bicubic interpolation, NaN otherwise.

Thus, after the projection step we have a set of N textures defined on the same mesh. Each texture has "holes" of NaN at different parts, where the surface was not visible from the corresponding image. We can now already try merge all these textures (section 3.5), but for some merging criteria it is better to first detect the shadows.

3.4. Shadow detection

The images have all been taken around the same hour (2 pm +- 15min) but over a year. Thus while the attached shadows stay more or less the same, the length of the cast shadows varies a lot between summer and winter. We want to detect these shadows as accurately as possible.

Because the images provide us with the sun position (azimuth and elevation) it is natural to start by predicting the shadows using the mesh geometry. To this end, we consider the sun in the same way as we did the camera in the previous section. A "sun plane" is created by analogy with the image plane and we say that a vertex is in the shadow if it is not visible in this artificial plane. The method is exactly the same as the one described above in section 3.3. It is also possible to distinguish between the attached shadow and the cast shadow. Indeed a vertex is in the attached shadow if the triangles it belongs to face away from the sun. In this step we store

- S, a binary mask equal to zero in the shadow,
- Sa, a binary mask equal to zero in the attached shadow,

two vectors of length the number of vertices.

But the mesh is only a rough approximation of the real surface. Moreover the actual surface changes during the year: trees grow and lose their leaves in winter, cars move, etc. To get a more precise detection we use the fact that shadows in satellite images are very dark and we assume that they can be detected by a simple thresholding. The threshold t is obtained by L_1 minimisation:

$$t = \operatorname{argmin} \sum_v \mathbb{1}_{\text{PAN}_v > t} \mathbb{1}_{S_v = 0} + \mathbb{1}_{\text{PAN}_v < t} \mathbb{1}_{S_v = 1}. \quad (6)$$

The drift-field of an image I, a contrast invariant feature of the image defined as $\mathbf{d}_I = \frac{\nabla I}{I}$, encodes shadows only at their boundaries [28, 5]. Dilatation and contraction of the shadow masks provide the approximate boundaries of the shadows and the attached shadows. The terminator, the edge between the attached shadow and the lit areas of the image, is the intersection of the shadow boundary and the attached shadow boundary. These additional data make it possible to perform a shadow removal step as described in [5]. See Figure 10 for an illustration of these different steps.

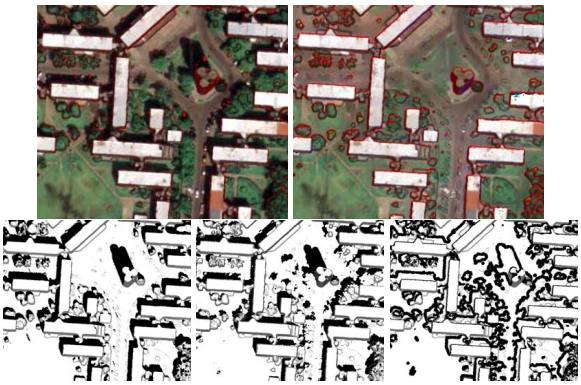


Figure 10: Top: image projected on the mesh before and after a shadow removal step [5]. Bottom: predicted shadow from geometry, shadow from thresholding and estimated shadow boundary.

3.5. Data fusion

The last and most important step is the data fusion. We have N textures defined over the vertices of the same mesh. We want to obtain a single texture over this mesh. This fusion is performed using weighted aggregators, operating independently at each vertex of the mesh. At each vertex there are N features $x_1, \ldots, x_N$ that we want to merge. These features may be either the RGB colours or, in which case $x_i \in \mathbb{R}^3$ or the colour gradient, in which case $x_i \in \mathbb{R}^6$.

For the aggregation, we use Fréchet means $F_{p,q}$ defined as

$$F_{p,q}(\omega; \mathbf{x}) := \arg\min_m \sum_i^N \omega_i^q \|x_i - m\|^p. \quad (7)$$

The Fréchet means include as particular cases the average ($p = 2, q = 0$), the geometric median ($p = 1, q = 0$), and a geometric "mode" ($p = \frac{1}{2}, q = 0$). By setting $q > 0$ we obtain weighted average, weighted median, and weighted modes. The value of the parameter q controls the importance given to the weights. For $q \to \infty$, only the feature with largest weight is kept.

This simplifies the scripting considerably: the fusion is always computed using all the images but a specific image can be discarded by setting its weight to zero. Notice that since the textures are incomplete (due to occlusions), some of the values x_i are NaN; this is equivalent to setting their weight to zero.

In what follows we assume that the weight w_i of a point is the combination of some criteria pertaining to the shadows and the positive part of the cosine of the angle between the normal at the surface on that point and the line of sight:

$$\omega_i = C(S_i, Sa_i) \max(0, \cos(\vec{n}, \text{sight}_i)) \quad (8)$$
$$= C(S_i, Sa_i) \max(0, Sc_i), \quad (9)$$

where Sc is the vector acquired at the data projection step. Thus the weight is 1 for exactly fronto-parallel surface patches, and decreases until 0 for perpendicular and invisible surface patches. Furthermore, the weight can be set to zero depending on the shadow trimaps and the fusion criterion used.

In the simplest case, we can aggregate the colours of each image to obtain a texture $y = f(x_1, \ldots, x_N)$. However, this has some limitations because the contrasts of the images are very different, as shown in the experiments. A more advanced idea is to use a PDE-based fusion [5, 28, 5]. Let D be a differential operator and f an aggregator function. You compute $D^{-1} \circ f \circ D$. Notice that D^{-1} involves solving a PDE (partial differential equation), and you typically have to give a Dirichlet boundary condition on some points (always), and optionally a few Neumann boundary conditions. Typical examples include $D =$ gradient and $D^{-1} =$ Poisson equation [23]

$$\Delta u = \text{div}(f(\nabla I_1, \ldots, \nabla I_N), \quad (10)$$

or $D =$ drift field and $D^{-1} =$ osmosis equation [28, 5]

$$\Delta u = \text{div}(f(\mathbf{d}_{I_1}, \ldots, \mathbf{d}_{I_N})u) \quad (11)$$

For $D = D^{-1} =$ identity we recover the previous cases based on colour values.

Using shadow trimaps T_i leads to even more fusion criteria. The T_i take values between 0 (completely inside the shadow), 1 (completely outside the shadow) and we can interpolate on the boundaries. There are different ways to use these trimaps. We can either:

1. Ignore the shadow trimaps
2. Multiply the weights of the aggregator by 0, at the points where the trimap does not equal to 1. Thus, the features that are inside the shadow are not used.
3. Use the trimap values as aggregator weights. If the trimap is smooth, this allows a smooth transition between the inside and the outside of the shadow.
4. Set the drift field to zero where the trimap does not equal 0 or 1. Thus, only drift fields that are completely inside or completely outside the shadow are used.
5. Solve the PDE on the region $T < 1$, with Dirichlet boundary condition determined by aggregating the images on the region T=1. For one image, it is equivalent to a shadow removal step [28, 5]. See Figure 10.

It is clear that taking the colour of the most frontal view for each vertex leads to clear discontinuities. These discontinuities also appear when taking a weighted median of the vertices colours (not shown). With the weighted average on

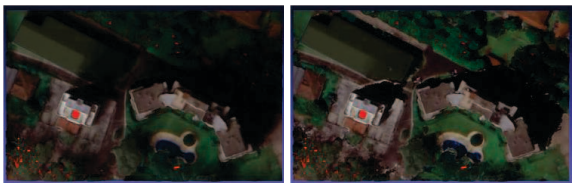


Figure 11: Weighted average of the colour information with and without the shadowed areas. Note how much information is recovered on the right when using only lit images.

Figure 12: Left: Colour-based shadow-less fusion. Right: Osmosis-based shadow-less fusion. For both results we took $f = F_{1,3}$. Note how the colour-based fusion results in visible seams in the image that disappear in the PDE-based fusion output. The white tower and the side of the hotel in the shadow in all the satellite images are much clearer using the PDE-based fusion.

Figure 13: Top rows: seven input images taken from one webcam and one from another webcam. Middle row: fusion by average (left) and median (right) of the colour values. Bottom row: median of the color values without the extraneous image (left) and median (right) of the drift-fields.

Figure 14: Left: mesh textured after a shadow-less osmosis-based fusion performed with $f = F_{2,3}$. Right: same result after adding back shadows. Note how it makes the tower and the trees easier to distinguish.

the vertices colours, with or without shadows, these seams are less noticeable but still present.

Performing a PDE-based fusion gives much smoother results. It is particularly noticeable for the weighted average of the drift-fields without features in a shadowed region. For these PDE-based fusions, we solved the osmosis equation on the region $T = 1$ and used as Dirichlet boundary conditions the weighted average of the vertices colours without shadows. In the result obtained with the fusion of the canonical drift-fields without shadowed features (weighted average or weighted median), the cast shadows have almost disappeared. The best results are obtained by computing the weighted median of the drift-fields without shadows. The median has the additional advantage of being robust to the presence of a few badly registered images.

Because of the imperfect geolocation of the images, even after the registration step, the weighted averages lead to some blurring, which is mitigated when computing a robust fusion using the Fréchet p-means such as the weighted median or by increasing q, the power attributed to the weights in the fusion. In the experiments we illustrate several interesting combinations of these choices. To better illustrate the effect of different criteria we first show it on a 2D case in Figure 13.

3.6. Adding back shadows

While it is satisfying to obtain a shadow-less final texture, in reality shadows are a vital part of our understanding of the topography of an area. For this reason we add back a shadow to our final result. For simplicity we arbitrarily choose the predicted shadow of one of the images. Note that this doesn't negate the importance of performing a shadow-less fusion. Indeed not removing the shadows in the fusion step would lead to a result with aberrant shadows in several directions.

4. Conclusion

Our algorithm proved to be well-suited for the case of multi-date satellite images. The PDE-based fusion is robust to incompatible dynamic ranges and varying shadow positions, especially when adding a geometric criterion to detect the shadows of the known objects. The output 3D models are such that all facades are correctly and seamlessly textured, with uniform colours and coherent shadows.

The vertex-wise representation is easier to work with for computing differential equations, and this is the choice retained. However large meshes will be huge and sluggish to display. Maybe in the future we can have the best of both worlds. Since refining and simplifying a mesh are standard and well-known operations, we can use the atlases for storing a final, efficient mesh, and still keep the vertex-wise representation for all the intermediate processing. These operations are standard in the CGAL library.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *2009 IEEE 12th international conference on computer vision*, pages 72–79. IEEE, 2009.

[2] M. Bosch, Z. Kurtz, S. Hagstrom, and M. Brown. A multiple view stereo benchmark for satellite imagery. In *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–9. IEEE, 2016.

[3] M. Boussaha, B. Vallet, and P. Rives. Large scale textured mesh reconstruction from mobile mapping images and lidar scans. In *International Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences ISPRS*, 2018.

[4] L. Cipriani, F. Fantini, and S. Bertacchi. 3d models mapping optimization through an integrated parametrization approach: cases studies from ravenna. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 45, 2014.

[5] M. d'Autume, J.-M. Morel, and E. Meinhardt-Llopis. A flexible solution to the osmosis equation for seamless cloning and shadow removal. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2147–2151. IEEE, 2018.

[6] C. de Franchis. *Earth Observation and Stereo Vision*. PhD thesis, ENS Cachan, 2015.

[7] C. de Franchis, G. Facciolo, and E. Meinhardt-Llopis. s2p ipol demo, 2014.

[8] C. De Franchis, E. Meinhardt-Llopis, J. Michel, J.-M. Morel, and G. Facciolo. An automatic and modular stereo pipeline for pushbroom images. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014.

[9] J. Demantké, B. Vallet, and N. Paparoditis. Facade reconstruction with generalized 2.5 d grids. *Int. Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, pages 67–72, 2013.

[10] G. Dial and J. Grodecki. Rpc replacement camera models. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(part XXX), 2005.

[11] G. Facciolo, C. De Franchis, and E. Meinhardt-Llopis. Automatic 3d reconstruction from multi-date satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 57–66, 2017.

[12] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, J. HUGHES, and E. ANGEL. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.

[13] B. Goldlücke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *International journal of computer vision*, 106(2):172–191, 2014.

[14] K. Gong and D. Fritsch. Point cloud and digitalsurface model generation from high resolution multiple view stereo satellite imagery. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42(2), 2018.

[15] R. Grompone, E. Meinhardt-Llopis, J.-M. Morel, M. Rais, and M. Rodriguez. Aerial/satellite image registration final report to TOSA, Thalès. 2018.

[16] C. Guérin, R. Binet, and M. Pierrot-Deseilligny. Dsm generation from stereoscopic imagery for damage mapping, application on the tohoku tsunami. In *International Geosciences and Remote Sensing Symposium*, 2013.

[17] C. Guerin, R. Binet, and M. Pierrot-Deseilligny. Automatic detection of elevation changes by differential dsm analysis: Application to urban areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(10):4020–4037, 2014.

[18] M. Guislain, J. Digne, R. Chaine, D. Kudelski, and P. Lefebvre-Albaret. Detecting and correcting shadows in urban point clouds and image collections. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 537–545. IEEE, 2016.

[19] M. Guislain, J. Digne, R. Chaine, and G. Monnier. Fine scale image registration in large-scale urban lidar point sets. *Computer Vision and Image Understanding*, 157:90–102, 2017.

[20] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] Y. Liu, X. Zheng, G. Ai, Y. Zhang, and Y. Zuo. Generating a high-precision true digital orthophoto map based on uav images. *ISPRS International Journal of Geo-Information*, 7(9):333, 2018.

[22] S. Loriot, J. Tournois, and I. O. Yaz. Polygon mesh processing. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.14 edition, 2019.

[23] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM TOG*. ACM, 2003.

[24] H. sun Kim, Y. ji Ban, and C. joon Park. A seamless texture color adjustment method for large-scale terrain reconstruction. *SIGGRAPH*, 2018.

[25] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello. Hierarchical structure-and-motion recovery from uncalibrated images. *Computer Vision and Image Understanding*, 140:127–143, 2015.

[26] V. Tsiminaki, J.-S. Franco, and E. Boyer. High resolution 3d shape texture from multiple videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1502–1509, 2014.

[27] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision*, pages 836–850. Springer, 2014.

[28] J. Weickert, K. Hagenburg, M. Breuß, and O. Vogel. Linear Osmosis Models for Visual Computing. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 26–39. Springer, Berlin, Heidelberg, Berlin, Heidelberg, Aug. 2013.

[29] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. Reynolds. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

[30] B. Wu, L. Xie, H. Hu, Q. Zhu, and E. Yau. Integration of aerial oblique imagery and terrestrial imagery for optimized 3d modeling in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 139:119–132, 2018.