

IMAGE-BASED TOPOGRAPHY RECONSTRUCTION BY MINIMIZING THE REPROJECTION-ERROR OVER A SURFACE

Shalev Mor, Sagi Filin

Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology, Haifa, Israel
(filin,shalevm)@technion.ac.il

Commission III, WG III/4

KEY WORDS: Surface reconstruction, Surface Evolution, Photo-consistency, Reprojection-error Minimization

ABSTRACT:

Image based three dimensional modelling of small scale scenes sees an increased demand in recent years due to algorithmic advances, and growing awareness of users to the value of scene documentation in three dimensions. Nonetheless, most of the reported image-based reconstruction methods are either point driven, relying on matching large set of automatically sampled points (e.g., SIFT, SURF), or integral approaches. The latter are commonly based on generating an entire map of matching. For many such stereo-correspondence methods, however, images which are acquired from a narrow baseline and similar orientations are usually needed, in order to ensure satisfying results. We present in this paper a reconstruction approach, assuming both visibility and Lambertian reflectance, which is intended to be less affected by image difference. The proposed method performs a surface evolution which is implemented by an iterative scheme, applying a gradient flow which minimizes a reprojection-error cost function. The algorithm is applied on scenes of complex topography and varied terrain. Our results show applicability of the method, along with its satisfactory performance for some difficult imagery configurations.

1. INTRODUCTION

Detailed three dimensional mapping of small scale scenes sees an increased demand in recent years. This trend can be attributed to increasing computing power, graphic hardware, and technological and algorithmic advances. It has also been thriving by growing awareness of users to the value of scene documentation in three dimensions, such as visualization, scene analysis, and more.

Among the various data acquisition techniques, the most effective ones for geographical scenes reconstruction are based on laser scanning technology and images. Although laser based point-clouds provide a dense set of data for scene description, limited availability of scanners and their prohibitive cost makes them not as widely applied as may be anticipated. In contrast, low cost cameras are far more available, and enable 3D modelling which is less costly, and in many cases generate data that are more detailed. Consequently, image-based 3D modelling is still relevant nowadays (Remondino and Menna, 2008). Image based methods for three-dimensional reconstruction, are often motivated by photo-consistency criteria. Most of the reported approaches for environmental and geographical scenes are point driven, and rely in many cases on the detection and matching or a large set of points (e.g., using the SIFT or SURF operators) (Remondino and El-Hakim, 2006; Barazzetti et al. 2010). Traditional methods for integral scene reconstruction are stereo-vision based and generate disparity-maps which match each image pixel to a counterpart in the other image (Figure 1a). Generation of such maps is often subjected to an optimization scheme of some energy function, e.g., graph-cuts, belief-propagation and so forth (Scharstein and Szeliski, 2002; Brown et al., 2003). Notwithstanding, implementation is usually carried out in image-space, and is subjected to its resolution and precision, whereas the final reconstruction model is given in three-dimensional space. Moreover, these methods seem ineffective for wide baselines or significant orientation differences. Therefore, they may turn less

applicable in complex natural environments which contain varied topography along with discontinuity and unsmooth features therein (Mor and Filin, 2009).

A somewhat different approach for 3D modelling is surface evolution (Faugeras and Keriven, 1998, Gargallo et al, 2007, Delaunoy and Prados, 2011) which often generates a model which minimizes the images reprojection-error objective function (also known as energy or cost function) (Seitz et al., 2006). This minimization is based on the assumption that the intensity image values of the same object-space point should share similar values; more realistically, have a minimal intensity difference between them. The process begins with an initial coarse approximated surface, and in each iteration the surface should change or "evolve" by some criteria until convergence to the optimal solution (by minimizing an objective function) is reached.

Following such approaches, this paper proposes a methodology for three-dimensional surface reconstruction of a textured scene, which is motivated by photo-consistency, assuming visibility and Lambertian reflectance. Our objective is to generate an explicit digital surface model (DSM) based on minimization of the integral reprojection-error generated by an image-set over the surface. An initial coarse surface is evolved iteratively using the gradient of a reprojection-error energy function (gradient-descent method). Contrary to traditional reconstruction methods, our approach is surface-based (Figure 1b) and advancement is computed in three-dimensional model-space. The advantages of the proposed approach over traditional ones lie in the fact that the surface reconstruction is generated in a three-dimensional model space which makes it less subjected to image resolution. Moreover, since no point- or pixel-direct matching is required, the use of image sets with wide baselines and significant difference in their orientations becomes possible and in a more effective manner. The paper presents a theoretical scheme, along with an implementation for a triangular mesh model. Modification to other explicit surface representations is, however, straightforward.

The organization of this paper is as follows: we first introduce the main concept of the surface reconstruction as casted by an energy minimization. Section 2.1 develops expressions for the energy function gradient which enables to evolve the surface. Section 2.2 elaborates on computational aspects which enable us to implement our method for digital images. Section 3 demonstrates the application of the method on an archaeological site that offers complex topography including varied terrain, discontinuities and unsmooth features therein. The results show applicability of the method, along with its satisfactory performance for difficult imagery configurations. The paper concludes with summary and discussion.

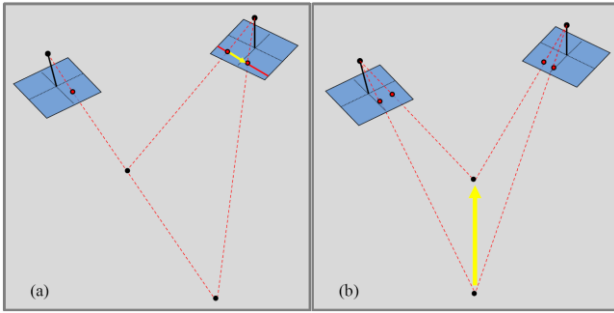


Figure 1. Illustration of 3D point determination by a one-dimensional search, a) an image-space search along the epipolar line using the radiometric information directly, b) the search is along the vertical locus, and the radiometric values are obtained by projection to image-space. The positioning in both cases is motivated by requiring minimal values for the intensity differences.

2. SURFACE RECONSTRUCTION

Our surface reconstruction approach is based on minimizing the reprojection-error in the three-dimensional object space. In order to do so we first describe the reprojection-error as a scalar field over the \mathbb{R}^3 Euclidean domain. For simplicity we focus in first on an image pair alone, but the expansion to multiview configuration is relatively straightforward by adding terms for all image pairs.

Consider two images of a specific scene: a reference image I and a target image I' . Every object-space point $\mathbf{x} = [x, y, z]^T$ can be projected into each one of the images, and retrieve the intensity value $\hat{I}(\mathbf{x})$ and $\hat{I}'(\mathbf{x})$. Notice that here $\hat{I}: \mathbb{R}^3 \rightarrow \mathbb{R}$ is a function over a point in the three-dimensional object-space (as opposed to $I(u, v)$ which is the intensity function for a two-dimensional image-space point). The relation between the two is $\hat{I}(\mathbf{x}) = I(P(\mathbf{x}))$, where $P: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the projection of the object-space coordinate onto the image. We can now define the reprojection-error scalar field as the square difference of intensity values as given in Eq. (1)

$$D(\mathbf{x}) = (\hat{I}(\mathbf{x}) - \hat{I}'(\mathbf{x}))^2 \quad (1)$$

This scalar field may be calculated for any point in the field-of-view of both images. Figure 2 provides a graphic illustration of such a field.

We would like to find a surface Γ which represents the scene optimally, i.e., a surface that would maintain photo-consistency, or minimize the reprojection-error. As our interest is in an integral solution for the entire surface, our objective would be minimizing the cost function given in Eq. (2):

$$E = \iint_{\Gamma} (\hat{I}(\mathbf{x}) - \hat{I}'(\mathbf{x}))^2 d\sigma \quad (2)$$

where $d\sigma$ is the differential surface area element. This is an integration of the reprojection-error field $D(\mathbf{x})$ over the surface Γ . Using continuous surface representation the optimization can be formulized as a variational problem. However, we can also use a discrete surface representation, such as Triangular Irregular Network (TIN) mesh or Digital Terrain Model (DTM) grid. In such a case, the surface would be represented by a set of piecewise functions (planes for TIN, bilinear or bi-cubic for grid etc.) where each of them depends on some finite number of parameters:

$$\min_{\mathbf{t}=[t_1, \dots, t_{n_p}]^T} \sum_{k=1}^{n_T} \iint_{\mathbf{T}_k} (\hat{I}(\mathbf{x}) - \hat{I}'(\mathbf{x}))^2 d\sigma \quad (3)$$

The choice here is of a triangle-mesh representation, and so, the integration is over a set of triangles $\{\mathbf{T}_k, k = 1, 2, \dots, n_T\}$, and the solution is of a finite vector of n_p z -ordinate $[t_1, t_2, \dots, t_{n_p}]$.

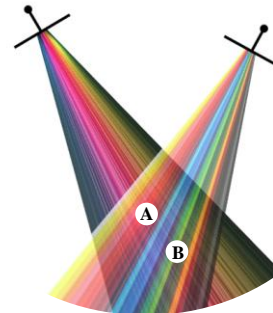


Figure 2. Illustration of the reprojection-error field. Region A is of low reprojection-error as its projection color in both images is red. On the other hand, the reprojection-error of region B is high due to the collision between red and green colors of its image projection.

Our choice of solving this optimization problem is by using the gradient-descent method. Using it we should calculate the gradient of the objective function given in Eq. (3), i.e., its vector of derivatives with respect to the z -ordinate $[t_1, \dots, t_{n_p}]$ of each of the vertices (Figure 3). The update for each iteration q should be $t_j^{(q+1)} = t_j^{(q)} - \alpha \frac{\partial E}{\partial t_j}$ for some scalar step of size α .

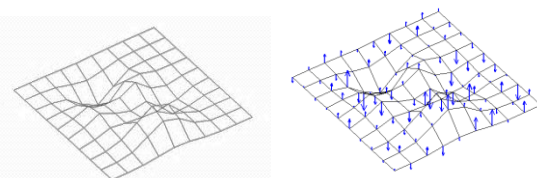


Figure 3. A surface (left) and the gradient of its vertices (right) visualized by the arrows.

2.1 Gradient of the Energy function

The objective energy function introduced in Eq. (3) is in fact an integral over the entire surface, and therefore depends on the parameters t_i . As these parameters change, the surface evolves. Notice that the domain of integration changes as well as the energy function value. This concept of a surface flow according to some direction, which is the gradient in our case, is

sometimes related as gradient flow (Delaunoy and Prados, 2011).

To illustrate the flow, consider a surface that depends on a single parameter, t , termed in the following as time, for traditional reasons. For simplification of the mathematical derivation, we restrict our surface to a single triangle. Our energy function is an integration of some scalar field over the surface.

2.1.1 Time Derivative for a Single Triangle

Consider a triangle T in 3D-space, defined by its three vertices $\mathbf{x}_i = [x_i, y_i, z_i]^T$, $i=1,2,3$. Let $D: \mathbb{R}^3 \rightarrow \mathbb{R}$ be some scalar field, then its integration over T may be expressed as a surface integral E . In order to introduce surface evolution or flow, we consider the triangle surface as moving in time in such a way that its three vertices move along some time dependent curves (Figure 4a). Consequently, the domain of integration also depends on time, and thus can be written as:

$$E(t) = \iint_{T(t)} D(\mathbf{x}) d\sigma \quad (4)$$

Since the domain of integration in Eq. (4) is also time dependent, computing the derivative dE/dt is not a trivial task. Thus we should use the *Leibniz integral rule* for differentiation under the integral sign (Flanders, 1973). To overcome this difficulty we may transform the domain of integration into a 2D coordinate system for which it has no time dependency. This is done by changing the surface domain to the unit triangle in \mathbb{R}^2 which is defined by $\{(r,s) \in [0,1]^2 \mid r+s=1\}$. The integral may now be written as:

$$E = \int_0^1 \int_0^{1-s} D(\mathbf{x}(r,s)) \left\| \frac{\partial \mathbf{x}}{\partial r} \times \frac{\partial \mathbf{x}}{\partial s} \right\| dr ds \quad (5)$$

where any triangle points $\mathbf{x}(r,s) = \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1)r + (\mathbf{x}_3 - \mathbf{x}_1)s$, determined by r and s , is a convex combination of the three vertices. The vector $\mathbf{g} = \frac{\partial \mathbf{x}}{\partial r} \times \frac{\partial \mathbf{x}}{\partial s} = (\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)$ is also defined here in such a way that $d\sigma = \|\mathbf{g}\| dr ds$ enables us to compute a surface integral over the 2D domain space (r,s) as shown in Eq. (5). Since now the domain of integration is not time dependent, we may derive it and obtain the expression in Eq. (6).

$$\begin{aligned} \frac{dE}{dt} &= \frac{d}{dt} \int_0^1 \int_0^{1-s} D \|\mathbf{g}\| dr ds \\ &= \int_0^1 \int_0^{1-s} \left(\frac{dD}{dt} \|\mathbf{g}\| + D \frac{d\|\mathbf{g}\|}{dt} \right) dr ds \\ &= \iint_T (\nabla D \cdot \dot{\mathbf{x}}) d\sigma + \iint_T D \frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} d\sigma \end{aligned} \quad (6)$$

where we use the 3D gradient $\nabla D = [\frac{\partial D}{\partial x}, \frac{\partial D}{\partial y}, \frac{\partial D}{\partial z}]^T$ and the time derivative $\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial t} = [\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial z}{\partial t}]^T$, so that the transition $dD/dt = \nabla D \cdot \dot{\mathbf{x}}$ becomes a result of applying the chain rule. We have also used the notation $\dot{\mathbf{g}} = d\mathbf{g}/dt$. Moreover, since \mathbf{g} is computed only using the triangle vertices it does not depend on the integration parameters r and s , and therefore can be taken outside of the integral, which leads to the following expression for the derivative

$$\frac{dE}{dt} = \frac{d}{dt} \iint_T D(\mathbf{x}) d\sigma = \iint_T (\nabla D \cdot \dot{\mathbf{x}}) d\sigma + \frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} \iint_T D(\mathbf{x}) d\sigma \quad (7)$$

We can further show that the vector derivative $\dot{\mathbf{x}}$ and the expression $\frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2}$ can be expressed as a linear function of the vertices derivative $\dot{\mathbf{x}}_1$, $\dot{\mathbf{x}}_2$, and $\dot{\mathbf{x}}_3$, as given in Eqs. (8) and (9).

$$\dot{\mathbf{x}} = \frac{1}{g_0} [(\mathbf{x}_2 \times \mathbf{x}_3) \cdot \dot{\mathbf{x}}] \dot{\mathbf{x}}_1 + \frac{1}{g_0} [(\mathbf{x}_3 \times \mathbf{x}_1) \cdot \dot{\mathbf{x}}] \dot{\mathbf{x}}_2 + \frac{1}{g_0} [(\mathbf{x}_1 \times \mathbf{x}_2) \cdot \dot{\mathbf{x}}] \dot{\mathbf{x}}_3 \quad (8)$$

$$\frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} = \frac{1}{g} [(\mathbf{n} \times (\mathbf{x}_3 - \mathbf{x}_2)) \cdot \dot{\mathbf{x}}_1] + \frac{1}{g} [(\mathbf{n} \times (\mathbf{x}_1 - \mathbf{x}_3)) \cdot \dot{\mathbf{x}}_2] + \frac{1}{g} [(\mathbf{n} \times (\mathbf{x}_2 - \mathbf{x}_1)) \cdot \dot{\mathbf{x}}_3] \quad (9)$$

where the value g_0 is the triple product of the three vertices $g_0 = (\mathbf{x}_1 \times \mathbf{x}_2) \cdot \mathbf{x}_3$; $g = \|\mathbf{g}\|$ is the norm of vector \mathbf{g} and equals to twice the area of the triangle; and the vector $\mathbf{n} = \mathbf{g}/\|\mathbf{g}\|$ is the unit normal vector of the triangle.

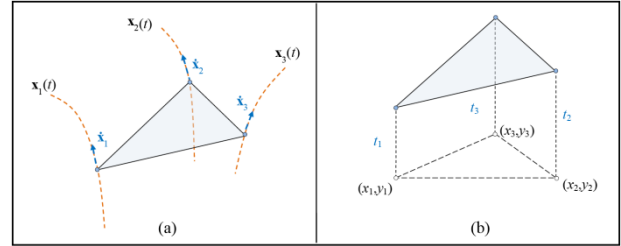


Figure 4. a) a general time dependent triangle surface model, b) an explicit triangle surface model.

2.1.2 The Derivative for an explicit TIN model

As we have chosen to use an explicit triangular mesh for a surface representation, the shape of the surface is determined by the z -ordinates of the triangles vertices (Figure 4b). For a single triangle T_k , the energy E_k which is integrated over it, depends only on a limited (three) set of parameters, t_j , which correspond to its three vertices. The gradient of our energy function, is therefore a vector containing the derivatives of the energy $E(t_1, \dots, t_{n_p})$ with respect to the values t_j of each vertex $j=1, 2, \dots, n_p$, where n_p is the number of vertices. Using Eq. (7) the j -th component of the gradient may be expressed by

$$\begin{aligned} \frac{\partial E}{\partial t_j} &= \sum_{k=1}^{n_t} e_{jk} \frac{\partial E_k}{\partial t_j} \\ &= \sum_{k=1}^{n_t} e_{jk} \left(\iint_{T_k} (\nabla D \cdot \dot{\mathbf{x}}) d\sigma + \frac{1}{\|\mathbf{g}\|^2} (\mathbf{g}^T \dot{\mathbf{g}}) \iint_{T_k} D(\mathbf{x}) d\sigma \right) \end{aligned} \quad (10)$$

where $e_{jk} = 1$ if the j -th vertex belonging to the k -th triangle T_k , and $e_{jk} = 0$ if not. Here, $\dot{\mathbf{x}}$ and $\dot{\mathbf{g}}$ are vector derivatives with respect to t_j .

For a single triangle T_k , and without any loss of generality, we name these three parameters t_1, t_2, t_3 , which correspond to the triangle vertices \mathbf{x}_i , $i=1,2,3$, respectively. The 3D derivative values of the vertices can either be $[0,0,0]^T$ or $[0,0,1]^T$ (each vertex is defined as $\mathbf{x}_i = [x_i, y_i, z_i]^T$ and derivatives are with respect to the z -ordinate of all vertices t_i , $i=1,2,3$). By substituting the derivatives into Eqs. (8) and (9) we get the following expressions:

$$\begin{aligned} \text{for } t = t_1 : \dot{\mathbf{x}} &= \frac{1}{g_0} [0, 0, (y_2 z_3 - y_3 z_2)x + (z_2 x_3 - z_3 x_2)y + (x_2 y_3 - x_3 y_2)z]^T \\ \text{for } t = t_2 : \dot{\mathbf{x}} &= \frac{1}{g_0} [0, 0, (y_3 z_1 - y_1 z_3)x + (z_3 x_1 - z_1 x_3)y + (x_3 y_1 - x_1 y_3)z]^T \\ \text{for } t = t_3 : \dot{\mathbf{x}} &= \frac{1}{g_0} [0, 0, (y_1 z_2 - y_2 z_1)x + (z_1 x_2 - z_2 x_1)y + (x_1 y_2 - x_2 y_1)z]^T \end{aligned} \quad (11)$$

and

$$\begin{aligned} \text{for } t = t_1 : \frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} &= \frac{1}{g} [n_1(y_3 - y_2) - n_2(x_3 - x_2)] \\ \text{for } t = t_2 : \frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} &= \frac{1}{g} [n_1(y_1 - y_3) - n_2(x_1 - x_3)] \\ \text{for } t = t_3 : \frac{\dot{\mathbf{g}} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} &= \frac{1}{g} [n_1(y_2 - y_1) - n_2(x_2 - x_1)]. \end{aligned} \quad (12)$$

Continuing with an expression for ∇D , the reprojection-error field is $D(\mathbf{x}) = (\hat{I}(\mathbf{x}) - \tilde{I}'(\mathbf{x}))^2$ as defined in Eq. (1), and therefore its 3D gradient is

$$\nabla D = \left[\frac{\partial D}{\partial x} \quad \frac{\partial D}{\partial y} \quad \frac{\partial D}{\partial z} \right]^T = 2(\hat{I} - \tilde{I}')(\nabla \hat{I} - \nabla \tilde{I}') \quad (13)$$

where \hat{I} and \tilde{I}' are the aforementioned 3D color fields generated by the reference and target image respectively. Deriving this color field function in 3D space is not as obvious, and so we use the 2D image function which generates it by the relation $\hat{I}(\mathbf{x}) = I(P(\mathbf{x}))$. The 3D gradient $\nabla \hat{I}$ may therefore be expressed as

$$\begin{aligned} \nabla \hat{I} &= \left[\frac{\partial \hat{I}}{\partial x} \quad \frac{\partial \hat{I}}{\partial y} \quad \frac{\partial \hat{I}}{\partial z} \right]^T \\ &= \left[\frac{\partial I}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial I}{\partial v} \frac{\partial v}{\partial x}, \quad \frac{\partial I}{\partial u} \frac{\partial u}{\partial y} + \frac{\partial I}{\partial v} \frac{\partial v}{\partial y}, \quad \frac{\partial I}{\partial u} \frac{\partial u}{\partial z} + \frac{\partial I}{\partial v} \frac{\partial v}{\partial z} \right]^T = I_u \nabla u + I_v \nabla v \end{aligned} \quad (14)$$

where here u and v are the 2D image coordinates and $\nabla u = \left[\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z} \right]^T$. Therefore, to complete the expression for the gradient we also need to find the derivatives of the 2D image-space coordinates $\mathbf{u} = [u, v]^T$ with respect to the object-space 3D coordinates $\mathbf{x} = [x, y, z]^T$. We assume a pinhole model and therefore use the co-linearity principle, which can be written using the projection matrix $\mathbf{P}_{3 \times 4}$:

$$\begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} \sim \mathbf{P} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \quad \text{or} \quad \begin{aligned} u &= P_1(x, y, z) = \frac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} \\ v &= P_2(x, y, z) = \frac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} \end{aligned} \quad (15)$$

The partial derivative of (u, v) with respect to (x, y, z) may be expressed by the Jacobian matrix

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \frac{\partial (u, v)}{\partial (x, y, z)} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{p_{11} - p_{31}u}{w} & \frac{p_{12} - p_{32}u}{w} & \frac{p_{13} - p_{33}u}{w} \\ \frac{p_{21} - p_{31}v}{w} & \frac{p_{22} - p_{32}v}{w} & \frac{p_{23} - p_{33}v}{w} \end{bmatrix} \quad (16)$$

where $w = p_{31}x + p_{32}y + p_{33}z + p_{34}$. This provides us with the required expressions, since the first row of the matrix in Eq. (16) is ∇u^T and the second row is ∇v^T .

We can now write explicit expressions for the derivatives of E . Prior to that we note that the vectors we obtained for $\dot{\mathbf{x}}$ in Eq. (11) have zero value for their first two components, namely, $\dot{\mathbf{x}} = [0, 0, \partial z / \partial t_i]^T$ for $i=1,2,3$. This is an expected result, since the parameters t_j affect the surface only on the z -direction. Consequently, in order to compute the expression $\nabla D \cdot \dot{\mathbf{x}}$, we may consider only the derivative $D_z = \partial D / \partial z$ and ignore the derivatives with respect to x and y . Therefore, combining Eqs. (13), (14), and (16), we have

$$\begin{aligned} D_z &= \frac{\partial D}{\partial z} = 2 \left[(\hat{I} - \tilde{I}') \left(I_u \frac{p_{13} - p_{33}u}{w} + I_v \frac{p_{23} - p_{33}v}{w} \right) \right. \\ &\quad \left. + (\tilde{I}' - \hat{I}) \left(I'_u \frac{p'_{13} - p'_{33}u'}{w'} + I'_v \frac{p'_{23} - p'_{33}v'}{w'} \right) \right] \end{aligned} \quad (17)$$

we use the tag sign ($'$) to relate to the target image. As a result and using Eqs. (7), (11), and (12), we obtain the necessary derivatives of E .

$$\begin{aligned} \frac{\partial E_k}{\partial t_1} &= \frac{1}{g_0} \iint_{T_i} [(y_2 z_3 - y_3 z_2) \mathbf{x} + (z_2 x_3 - z_3 x_2) \mathbf{y} + (x_2 y_3 - x_3 y_2) \mathbf{z}] D_z d\sigma \\ &\quad + \frac{1}{g} [(y_3 - y_2) n_1 + (x_2 - x_3) n_2] \iint_{T_i} D d\sigma \\ \frac{\partial E_k}{\partial t_2} &= \frac{1}{g_0} \iint_{T_i} [(y_3 z_1 - y_1 z_3) \mathbf{x} + (z_3 x_1 - z_1 x_3) \mathbf{y} + (x_3 y_1 - x_1 y_3) \mathbf{z}] D_z d\sigma \\ &\quad + \frac{1}{g} [(y_1 - y_3) n_1 + (x_3 - x_1) n_2] \iint_{T_i} D d\sigma \\ \frac{\partial E_k}{\partial t_3} &= \frac{1}{g_0} \iint_{T_i} [(y_1 z_2 - y_2 z_1) \mathbf{x} + (z_1 x_2 - z_2 x_1) \mathbf{y} + (x_1 y_2 - x_2 y_1) \mathbf{z}] D_z d\sigma \\ &\quad + \frac{1}{g} [(y_2 - y_1) n_1 + (x_1 - x_2) n_2] \iint_{T_i} D d\sigma \end{aligned} \quad (18)$$

2.2 Computation

Computing the integral in Eq. (3) in 3D object space is a challenging task. Although the 3D function $\hat{I}(\mathbf{x})$ is theoretically well defined, it is difficult to implement it practically. Moreover, our basic units for color data are image pixels, and therefore it is reasonable to compute this integral in the 2D image-space (u, v) . We may therefore write Eq. (3) in the following form

$$E = \iint_{P(T)} (I(\mathbf{u}) - I'(\mathbf{u}'))^2 \left\| \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v} \right\| du dv \quad (19)$$

where $\mathbf{u} = [u, v]^T$ is the 2D reference image coordinates. The function $P(\mathbf{x})$ in the domain of integration is the projection function from 3D object-space into 2D image-space: $\mathbf{u} = P(\mathbf{x})$.

Eq. (19) would be entirely computable in the reference image-space (u, v) had it not included the term $I'(\mathbf{u}')$ which is the actual target image-space. Therefore, in order for the integrand to be a function of single image-space domain (u, v) , we have to reproject the target image coordinates (u', v') onto the reference image. Since the two projection functions are defined by $\mathbf{u} = P(\mathbf{x})$ and $\mathbf{u}' = P'(\mathbf{x})$, we may determine that $\mathbf{u}' = P'(P^{-1}(\mathbf{u}))$. This composite function may seem indefinite, since we cannot trivially transform image coordinates from one to the other (by inverting P). However reprojection may be computed using the surface data. Taking the surface into consideration we can intersect a reference image ray generated by an image point, leading to a 3D object-space point, which can be then projected to the target image. A geometric illustration of such reprojection is given in Figure 5.

We now define the reprojected image function $\tilde{I}'(\mathbf{u}) = I'(\mathbf{u}') = I'(P'(P^{-1}(\mathbf{u})))$, which may be constructed by the original target image I' , the two perspective projection matrices of both images, and the surface. Notice that the reprojected image is a synthetic one created by first inversely-projecting the target image onto the surface, and then projecting it from the surface to the reference image-space. Using the above notations we may now write Eqs. (3) or (19) as:

$$E = \iint_{P(T)} (I(\mathbf{u}) - \tilde{I}'(\mathbf{u}))^2 J du dv \quad (20)$$

where

$$J = \left\| \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v} \right\| = \det(\mathbf{P}_3)^{-1} \frac{w^2}{\mathbf{n}^T (\mathbf{P}_3^{-1}) [u \ v \ 1]^T}, \quad \mathbf{P}_3 = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (21)$$

is the coefficient of the surface area element in the image-space, which is developed in the Appendix. As Eq. (21) shows, J depends on the image-space point $\mathbf{u} = [u, v]^T$, its inversely

projected 3D object-space coordinates $\mathbf{x} = [x, y, z]^T$, the image perspective projection matrix \mathbf{P} and the surface normal vector \mathbf{n} at this point. The value w is $p_{31}x + p_{32}y + p_{33}z + p_{34}$.

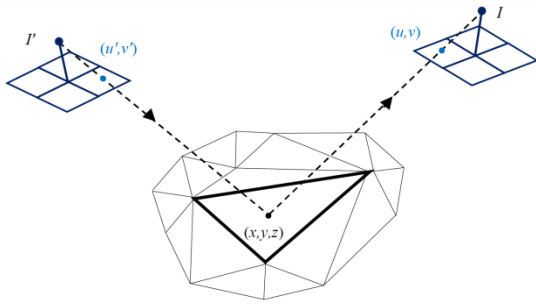


Figure 5. Illustration of an image-pair reprojection.

Applying the reprojection approach we can now write the following expression for integration of D_z . Using Eqs. (17) and (20) we may write

$$\iint_T D_z d\sigma = 2 \iint_{P(T)} (I - \tilde{I}) \left(I_u \frac{p_{13} - p_{23}u}{w} + I_v \frac{p_{23} - p_{33}v}{w} \right) J du dv + 2 \iint_{P'(T)} (I' - \tilde{I}') \left(I'_u \frac{p'_{13} - p'_{23}u'}{w'} + I'_v \frac{p'_{23} - p'_{33}v'}{w'} \right) J' du' dv' \quad (22)$$

which transforms the domain of integration from the 3D surface in object-space into both the reference and target image-spaces. Here \tilde{I}' is the inverse reprojected image from the reference to the target one, and J' is the coefficient for the target image. This enables us to compute the gradient of our energy function, and therefore perform a surface evolution in that direction. We detail here the algorithm for evolving an initial coarse surface based on minimization of the reprojection-error accumulated of an image pair of the scene.

```

#1  input:
#2  two images I and I'
#3  set of triangles T_i, i = 1, 2, ..., n_T
#4  initial set of vertices x_i, y_i, z_i = t_i, i = 1, 2, ..., n_p
#5  output:
#6  final values for the z-ordinates t_i, i = 1, 2, ..., n_p
#7  routine:
#8  assign E = 0 /* energy */
#9  define the gradient vector ∇E of n_p elements and assign them to 0
#10 for each triangle T_k, k = 1, 2, ..., n_T
#11     compute the values g_1, g_2, g_3, g_0 for the triangle
#12     compute the triangle normal n_1, n_2, n_3
#13 project all the triangle vertices coordinate on the image I
#14 for each pixel (u, v) in the image I
#15     find the triangle T_k containing the pixel (u, v)
#16     assign t_1, t_2, t_3 to be the indices of the vertices of T_k
#17     compute the surface point x, y, z (by intersection)
#18     compute J using Eq. (21)
#19     compute u_z and v_z using Eq. (16)
#20     compute the numerical derivatives I_u and I_v of image I at (u, v)
#21     compute u', v' by projecting x, y, z on image I'
#22     assign D = 1/2 [ I(u, v) - I'(u', v') ]^2
#23     assign D_z = [ I(u, v) - I'(u', v') ] [ I_u u_z + I_v v_z ]
#24     assign E_0 = DJ, E_1 = xD_z J, E_2 = yD_z J, E_3 = zD_z J
#25     assign E = E + E_0
#26     compute additions to the three gradient components ∂E/∂t_1, ∂E/∂t_2, ∂E/∂t_3
#27     using Eq.(18)
#28 repeat the actions in rows #13-#26 while swapping I and I'
#29 for each vertex i = 1 to n_p
#30     assign t_i = t_i + α ∂E/∂t_i /* for some step-size α */
    
```

The computation of the surface point (x, y, z) in line #17 of the algorithm can be done by intersection of the ray line, generated by the image pixel and the camera exposure point, and the triangle plane. Moreover, the image derivative I_u and I_v in line #20 has to be computed numerically by using, for example, a gradient kernel. This can be done as a preliminary stage since the images do not change between iterations.

3. RESULTS

The proposed surface reconstruction method has been applied for scenes within an archaeological excavation site. It is demonstrated on a documentation project in Oren cave in the Carmel mountain ridge, as part of on-going research studying the origins and use of Manmade-bedrock-holes (MBH) in those sites. An excavation campaign performed in June 2011 serves as the case study here. Images were acquired during the excavation with a Nikon D70 camera, with 48mm Nikkon lens.

We focus here on the reconstruction of a 3D surface model of MBH embedded within rough topographic settings. The input images were taken from a distance of approximately 1.5m from the MBH, such that the baseline is ~80cm, with a large orientation difference (Figure 6). The computation is generated by a few manually sampled points for producing the initial coarse surface (Figure 6). The initial surface can be seen in Figure 7a.



Figure 6. Images used for the reconstruction, along with sampled points for generating the initial coarse surface.

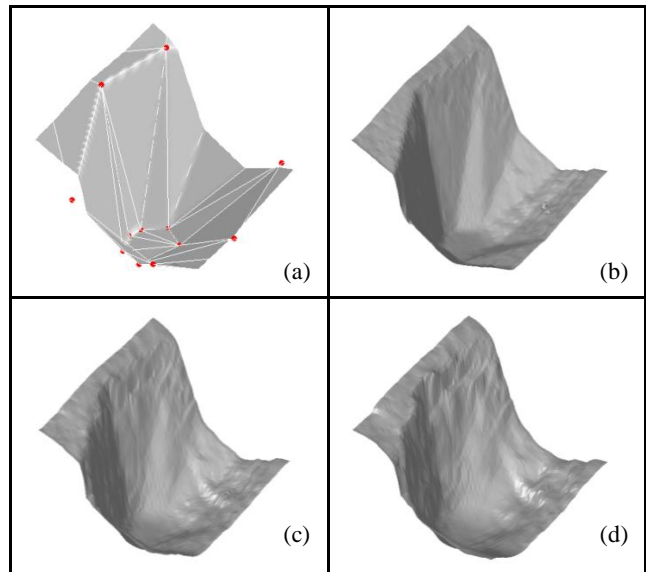


Figure 7. Stages of the surface evolution process, with the initial coarse surface in (a).

As noted the surface evolution is implemented by an iterative scheme, applying gradient flow on the z-ordinates of the vertices. Selected stages of the evolution are presented in Figure 7, and the final resulted surface is presented Figure 8a.

Results show a complete and successful reconstruction of the surface enabling also to depict fine details that reflect its formation. Application of the proposed model to other MBHs has proven successful as well.

The reconstruction has been compared to existing methods as well. Use of the same images for reconstruction using the graph-cuts method has failed in the reconstruction completely (Mor and Filin, 2009). It seems that these methods which are based on disparity map in image-space tend to give inadequate results for poorly textured scenes, and seem to fail when the imaging configuration is of wide-baselines and is characterized by significant orientation difference. Furthermore, applying the SIFT algorithm on the images has yielded a sparse set of points, most of them are on the outer surface of the MBH and are insufficient to reconstruct the object successfully (Figure 8b). Here again, texture, and the wide baseline and orientations have a significant effect on the outcome.

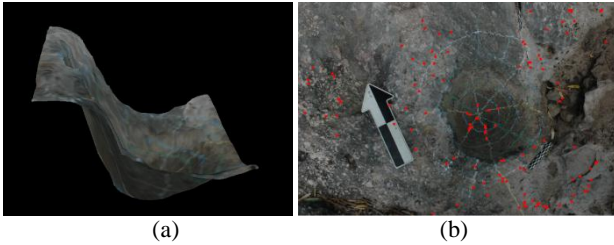


Figure 8. The reconstructed surface with texture in (a), whereas for comparison, a set of match points generated by the SIFT algorithm is presented in (b).

4. CONCLUSION AND DISCUSSION

This paper has presented a method for 3D reconstruction by surface evolution based on an image-pair, by employing a gradient flow for minimization of the reprojection-error. Additionally, an implementation scheme on a TIN mesh has been presented, and applied for complex topographic scenes. The results have shown applicability of the method, along with its satisfactory performance for some difficult imaging configurations. Future work may deal with expanding the method for several images, and evolving surfaces of non-explicit representation.

REFERENCES

Barazzetti, L., Remondino, F., Scaioni M., 2010. Extraction of accurate tie points for automated pose estimation of close-range blocks. *ISPRS The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Saint-Mandé, France, Vol. XXXVIII, Part 3A, pp. 151-156.

Brown, M. Z., Burschka, D., Hager, G. D., 2003. Advances in computational stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **25**(8), pp. 993-1008.

Delaunoy, A., Prados, E., 2011. Gradient flows for optimizing triangular mesh-based surfaces: applications to 3D reconstruction problems dealing with visibility. *Int. Journal of Computer Vision*, **95**(2), pp. 100-123.

Faugeras, O., Keriven, R., 1998. Variational-principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Trans. on Image Processing*, **7**(3), pp. 336-344.

Flanders, H., 1973. Differentiation under the integral sign. *The American Mathematical Monthly*, **80**(6), pp. 615-627.

Gargallo, P., Prados, E., Sturm, P., 2007. Minimizing the reprojection error in surface reconstruction from images. *In*

Proc. Of the Int. Conference on Computer Vision 2007, Rio de Janeiro, Brazil, (on CD-ROM)

Mor, S., Filin, S., 2009. Analysis of imaging configuration effect on surface reconstruction of complex sites. *In Proc. of FIG working week 2009*, Eilat, Israel, (on CD-ROM).

Remondino, F., El-Hakim, S., 2006. Image-based 3D modelling: a review. *Photogrammetric Record*, **21**(115), pp. 269-291.

Remondino F., Menna, F., 2008. Image-based surface measurement for close-range heritage documentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Beijing, China, Vol. XXXVII, Part B5, pp. 199-206.

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, **47**(1-3), pp. 7-42.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. *In Proc. of CVPR 2006*, New-York, USA (on CD-ROM).

APPENDIX

The appendix presents an expression for J , the coefficient transforming the area element from the surface to the image-space. This is the expression given in Eq. (21). The relation between the image-space coordinates (u,v) and the object-space 3D coordinates (x,y,z) can be stated by the following three equations:

$$\begin{aligned} p_{11}x + p_{12}y + p_{13}z + p_{14} &= (p_{31}x + p_{32}y + p_{33}z + p_{34})u \\ p_{21}x + p_{22}y + p_{23}z + p_{24} &= (p_{31}x + p_{32}y + p_{33}z + p_{34})v \\ g_1x + g_2y + g_3z &= g_0 \end{aligned}$$

where the first two equation are variation of the *co-linear law* of Eq. (15), and the third one is the triangle plane equation in the 3D object-space. Deriving these three equations with respect to u and v gives

$$\begin{aligned} p_{11}\frac{\partial x}{\partial u} + p_{12}\frac{\partial y}{\partial u} + p_{13}\frac{\partial z}{\partial u} &= (p_{31}\frac{\partial x}{\partial u} + p_{32}\frac{\partial y}{\partial u} + p_{33}\frac{\partial z}{\partial u})u + (p_{31}x + p_{32}y + p_{33}z + p_{34}) \\ p_{21}\frac{\partial x}{\partial u} + p_{22}\frac{\partial y}{\partial u} + p_{23}\frac{\partial z}{\partial u} &= (p_{31}\frac{\partial x}{\partial u} + p_{32}\frac{\partial y}{\partial u} + p_{33}\frac{\partial z}{\partial u})v + 0 \\ g_1\frac{\partial x}{\partial u} + g_2\frac{\partial y}{\partial u} + g_3\frac{\partial z}{\partial u} &= 0 \\ p_{11}\frac{\partial x}{\partial v} + p_{12}\frac{\partial y}{\partial v} + p_{13}\frac{\partial z}{\partial v} &= (p_{31}\frac{\partial x}{\partial v} + p_{32}\frac{\partial y}{\partial v} + p_{33}\frac{\partial z}{\partial v})u + 0 \\ p_{21}\frac{\partial x}{\partial v} + p_{22}\frac{\partial y}{\partial v} + p_{23}\frac{\partial z}{\partial v} &= (p_{31}\frac{\partial x}{\partial v} + p_{32}\frac{\partial y}{\partial v} + p_{33}\frac{\partial z}{\partial v})v + (p_{31}x + p_{32}y + p_{33}z + p_{34}) \\ g_1\frac{\partial x}{\partial v} + g_2\frac{\partial y}{\partial v} + g_3\frac{\partial z}{\partial v} &= 0. \end{aligned}$$

These two sets of equations may be written as

$$\mathbf{Q}\left[\frac{\partial x}{\partial u}\right] = [w \ 0 \ 0]^T \quad \text{and} \quad \mathbf{Q}\left[\frac{\partial x}{\partial v}\right] = [0 \ w \ 0]^T$$

where $\frac{\partial x}{\partial u} = \left[\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right]^T$, $w = p_{31}x + p_{32}y + p_{33}z + p_{34}$, and

$$\mathbf{Q} = \begin{bmatrix} p_{11} - p_{31}u & p_{12} - p_{32}u & p_{13} - p_{33}u \\ p_{21} - p_{31}v & p_{22} - p_{32}v & p_{23} - p_{33}v \\ g_1 & g_2 & g_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \\ \mathbf{g}^T \mathbf{P}_3^{-1} \end{bmatrix} \mathbf{P}_3.$$

Using the identity $(\mathbf{A}\mathbf{b}) \times (\mathbf{A}\mathbf{c}) = \det(\mathbf{A}) \mathbf{A}^{-T} (\mathbf{b} \times \mathbf{c})$

$$\frac{\partial x}{\partial u} \times \frac{\partial x}{\partial v} = \det(\mathbf{Q}^{-1}) \mathbf{Q}^T \left(\begin{bmatrix} w \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ w \\ 0 \end{bmatrix} \right) = \frac{1}{\det(\mathbf{Q})} \mathbf{Q}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w^2 = \left(\frac{w^2}{\det(\mathbf{Q})} \right) \mathbf{g}$$

and since the determinant of \mathbf{Q} is $\mathbf{g}^T (\det(\mathbf{P}_3) \mathbf{P}_3^{-1}) [u \ v \ 1]^T$, we get the expression of Eq. (21).