A Multi-view Dense Reconstruction for Rock Glacier Modelling

Article · September 2009

DOI: 10.1109/VSMM.2009.30

3 authors:

Javier de Matías
Universidad de Extremadura
7 PUBLICATIONS **17** CITATIONS
SEE PROFILE

José Moreno
Universidad de Extremadura
8 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Josechu J Guerrero
University of Zaragoza
139 PUBLICATIONS **1,312** CITATIONS
SEE PROFILE

A multi-view dense reconstruction for rock glacier modelling.

Javier de Matías Bejarano
Dpto. Ingeniería de Sistemas Informáticos y Telemáticos
University of Extremadura
Cáceres, Spain
jmatias@unex.es

José Moreno del Pozo
Dpto. Ingeniería de Sistemas Informáticos y Telemáticos
University of Extremadura
Cáceres, Spain
josemore@unex.es

José J. Guerrero Campo
Dpto. Informática e Ingeniería de Sistemas
Instituto de Investigación en Ingeniería de Aragón
Zaragoza, Spain
jguerrer@unizar.es

***Abstract*— Rock glacier is a rare case of study for 3D modelling. This type of models has particular properties which make it a special application for multi-view stereo reconstruction algorithms. In this work we analize different types of methods for dense reconstruction from wide base-line views. Besides that we select a method which was adapted and implemented for the specific case of rock glacier modelling. Principally we add two suitable innovations: implement a pyramidal image structure for get more efficient computation and use a feature-based method to obtain a initial terrain surface and improve the depth estimation. The proposed method has been tested in a real rock glacier example (Veleta in Sierra Nevada, Spain) with good results.**

Keywords: Photogrammetry, computer vision, dense reconstruction, rock glaciers.

I. INTRODUCTION

Basically a rock glacier consists of rocks covering a body of water ice. The ice creeps and carries the rocks and sediment along (active rock glaciers). Its study is very important in works related to geomorphology and geology. The geomorphologic structure observed in this study is the "Corral del Veleta" rock glacier (Sierra Nevada, Spain). This rock glacier is of high scientific interest because it is the southern most active rock glacier in Europe and therefore it has been analyzed every year since 2001 [18]. The research on the "Corral del Veleta" rock glacier is devoted to the study of its displacement and cartography through geodetic and photogrammetric techniques [15-17]. Traditional photogrammetry techniques to obtain a 3D model here are difficult to achieve and it took several trials since convergent and wide base-line images are taken from strategic points in the mountain. This is because the clasical restitution method is designed to work with image pairs in normal or quasi-normal configuration. Alternate techniques using convergent views require manual referencing or matching of corresponding points. Hence we need more automatic solutions to obtain dense reconstruction with enough accuracy.

In photogrammetry it start knowing the internal parameters of the cameras (internal orientation), the relation between the images (relative orientation) and the relation between the images and a general coordinate system (absolute orientation) [21]. And after it need to calculate a enough number of 3D points for build a DTM (Digital Terrain Model). It will use this DTM for obtain several types of cartographic products (2D maps, 3D videos, ortophotos, ...).

The aim of this work is to choose and adapt existing multiview stereo algorithms for rock glacier modelling. We search a computer vision approach for get a big amount of 3D points in an automatic way. Using common computer vision algorithms we can get the calibration and the exterior orientation of a set of cameras [19] in an automatic way using powerful features [12], but once we get this information we need a technique to generate a multi-view dense reconstruction.

There are a lot of techniques to obtain a dense reconstruction from images. Over the last years, a number of high-quality algorithms have been developed. The work by Seitz et al [1] categorizes existing methods according a six important properties: scene representation, photoconsistency measure, visibility model, shape prior, reconstruction algorithm, and initialization requirements.

We have a specific field of application and consequently we know what we want for the solution required. Therefore it is important to analyze multi-view algorithms respect to the properties established by Seitz. et al [1]. Following this work, we must consider six properties but, not all of them have the same importance. Usually scene representation and reconstruction algorithm are the principal properties because they often determine the other ones. For example, photo-consistency measure is influenced by scene representation and principally by reconstruction algorithm as we can see in the implemented solution. Furthermore there are two not important properties for our case of modelling: visibility model and shape prior. Visibility model is not so important in our application because we have very few occlusions in the 3D reconstruction

of a rock glacier surface. Besides that, shape prior is a difficult set of constraints in our application field since rock glaciers are too irregular. Consequently the properties with more influence in our election are *scene representation* and *reconstruction algorithm* and to a lesser degree initialization requirements. We want a simple solution (easy representations and easy implementation) with accurate results and a minimum of initialization requirements.

With respect to the scene representation we can find three posibilities: *Volumetric Methods* (Voxels [4], Levelsets [5], ...) which are very popular for their simplicity, uniformity, and ability to approximate any surface, but have complex processing. *Polygon Meshes* [6] which represents a surface as a set of connected planar facets, this representation is considered efficient to store and render, also have a complex processing. Finally *Depth Maps* [2,7] (one for each input view), the multi-depth-map representation avoids resampling the geometry on a 3D domain and hence simplify the scene representation.

According the reconstruction algorithm we can find four types. Methods that compute a *cost function on a 3D volume*, and then extracting a surface from this volume [4]. Those methods which iteratively *evolve a surface decreasing or minimizing a cost function* (Voxel, Levelset, Space carving [3,9]). Methods that compute a set of *depth maps*. These methods enforce consistency constraints between depth maps [7], or merge the set of depth maps into a 3D scene as a post process [2]. Finally algorithms that first extract and match a set of feature points and then *fit a surface to the reconstructed feature points* [8]. First and second types need complex implementations and expensive computation. The fouth type could not produce complete reconstruction since only discrete features are considered. The third type of reconstruction algorithms need simpler solutions and get enough accuracy.

Using these two properties (scene represenation and reconstruction algorithm) of the taxonomy proposed by Seitz [1] we have selected the work of Goesele [2] to implement a solution of dense reconstruction of rock glaciers. This is a suitable solution because depth maps are simpler than the other types of scene representation and the algorithm obtain enough accuracy using a easy implementation. Goesele propose a simple but robust multi-view stereo algorithm for compute depth maps using a window-based approach with good results in dense reconstructions.

The advantages of this approach are:

- The algorithm outputs are accurate shape estimates, according with the current best performing methods.
- Confidence is provided for each reconstructed point (it will be used for example for merging different depth maps into a mesh model).
- The performance is easy to understand, analyze and predict, due to algorithm's simplicity.

We can also mention drawbacks like:

- Holes can occur in areas with insufficient texture.
- Each surface point must be seen in at least three views to be reconstructed.

Therefore the chosen solution have insignificant inconvenients for rock glacier applications and very good advantages for an easy implementation and an easy scene representation. In the following section, we explain the method and the improvements we added to the original algorithm.

II. METHOD DESCRIPTION

The principal input parameters required by the algorithm to work are the set of images (3 or more) and the projection matrices (internal parameters and exterior orientation) of each camera. Also we need to know some data about the scene which we want reconstruct (measure of the scene). For example we will set a rough approximation of the scene depth where the correct depth will be searched.

The algorithm implemented in our solution consists of two steps:

- Reconstructing a depth map centered in each input view.
- Merging the different depth maps into a global mesh model.

A. Depth map generation.

Principal algorithm input is a set of views $V = \{V_1,...,V_n\}$. For each reference view $R \in V$, we select a set of k neighboring views $C = \{C_1,...,C_k\} \in V - R$ against which we correlate *R* using a robust window matching.

For each pixel *p* in *R*, we cover along its back-projected ray inside the depth interval established respect to the scene being reconstructed. We calculate the back-projection (as we can see in Fig.1) of pixel *p* in several values of *d* between the minimum and maximum depth with and increment of Δ_{depth}. For each depth value we compute the normalized cross correlation $NCC(R, C_j, d)$ using a $m \times m$ window centered on *p* and the corresponding windows centered on the projections in each of the views C_j ($j \in [1..k]$) with subpixel accuracy. When two views show a similar area of a textured object, we will obtain a high *NCC* score for value of *d*. If, otherwise, there is for example an occlusion (few important for rock glaciers examples), specular highlight, or other componding factor, the *NCC* value obtained will be low for all depths. We will rely on a depth value only if the window in the reference view correlates well with the corresponding window in several views. We define that a depth value *d* is valid if $NCC(R, C_j, d)$ is larger than a threshold *thresh* for at least two views in *C*. The set of views with *NCC* larger than *thresh* for a given depth *d* is denoted as $C_v(d)$.

For a valid depth *d* we compute a correlation value *corr(d)* as the mean of the *NCC* values of all views in $C_v(d)$:

$$corr(d) = \frac{\sum_{C_j \in C_v(d)} NCC(R, C_j, d)}{\|C_v(d)\|}$$

$\|C_v(d)\|$ evaluates the number of elements in $C_v(d)$. For each pixel p in R, the depth is chosen to be the value of d that maximizes *corr(d)*, or none if no valid d is found. The algorithm also compute a confidence value *conf(d)* for each recovered depth value as follows:

$$conf(d) = \frac{\sum_{C_j \in C_v(d)} NCC(R, C_j, d) - thresh}{\|C\|(1 - thresh)}$$

The confidence function increases with the number of valid views and is used to inform the merging step. In figures 2 and 3 the algorithms of the depth-map and correlation computations are shown.

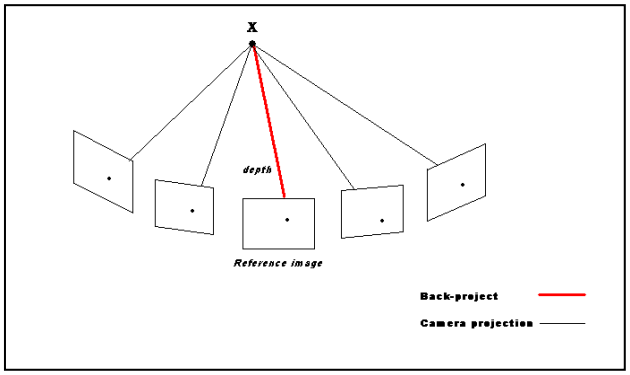

Figure 1. Back-projection and reprojection scheme.

Another important aspect in our solution is the depth definition. Usually depth is measured as the distance in the direction perpendicular to the image plane [21]. In our solution we consider the depth as the distance along the line joining the camera center and the 3D point which we want to reconstruct. In this way it is easier to define the prior interval where our algorithm searchs the final depth. In Fig. 2 and Fig. 3 we can see a pseudo-code of the method.

```
Pseudo-code of depth-map program
Objective: Calculate a depth map
Function:
for i from 1 to WITDH_IMAGE, and j from 1 to HEIGHT_IMAGE do
                q_2d = [i, j];
                for d from DEPTH_MIN to DEPTH_MAX do
                        Q_3d = back_project(q_2d, d);
                        corr = calculate_corr(q_2d, Q_3d, d);
                        we select the depth d with the best corr
                endfor
endfor
```

Figure 2. Pseudo-code calculate_depth_map

```
Pseudo-code of calculate_corr
Objective: calculate corr for a pixel.
Function:
k: number of neighboring views
NCC_total = 0
NCC_number = 0
for j from 1 to k do
        project Q_3d in j view;
        calculate NCC;
        if NCC > thresh then
                NCC_total = (NCC_total + NCC)/2;
                NCC_number = NCC_number +1;
        endif
endfor
```

Figure 3. Pseudo-code calculate_corr

B. *First depth estimation using previous terrain surface*

One of the main problem that we found using this dense reconstruction method is the estimation of an interval of initial depth for searching the final depth. Usually two prior maximum and minimum values are used for each image point [2]. If we have not a good initial depth estimation, the use of big intervals can be problematic because we need a big amount of computation time and the results could be no good (we can find high NCC values in other depth different to the real one). Hence it is neccesary a way to get a initial depth estimation with reasonable accuracy.

In our application we propose to use a previous terrain surface of the rock glacier area which we want to reconstruct. The intersection of camera rays with this terrain surface give us a good initial estimation of depth (Fig. 4).

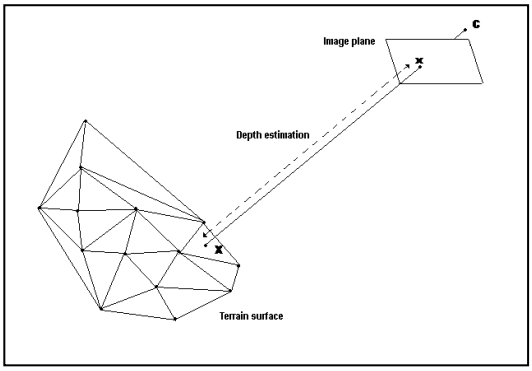

Figure 4. Estimation using triangulated surface.

In this way we have a good initial depth value where start searching the correct depth. The reconstruction method give us a way to complete the initial terrain surface and to get a dense 3D reconstruction. In experiment section we explain how get a terrain surface to get the initial depth estimation.

C. *Pyramidal structure.*

Another main problem is found working with big images (i.e. from camera sizes bigger than 12 Megapixels) since the original method [2] calculates a depth value in each pixel of the reference image. Therefore we need to calculate a depth value for each pixel and this increase a lot the computation cost and hence increase the required time for get a reconstruction too. In that sense we establish a pyramidal structure for the image processing. Pyramidal structure is a typical improvement in image processing [22], and is a suitable way to adapt an algorithm to our computation requirements.

In our case the goal is to work with images in different size levels (2 or 3). The highest level (smallest image) will be the image which we scan pixel by pixel. When we find the best correlation value (greater than the threshold) we test the correlation process with the corresponding pixel in the other

levels (i.e. lowest level in Fig. 5). Therefore each level has a specific purpose in the algorithm:

- The highest level (smallest image) determine the amount of pixels where we try to calculate a depth value and consequently the amount of posible 3D points for our reconstruction.
- The lowest level/s (largest images) enforce the depth that we choose in the lowest level.

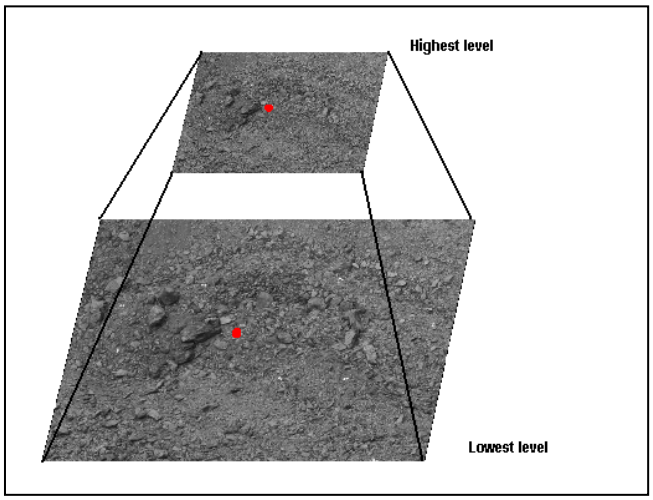

Figure 5. Pyramid structure

Using this simple technique we can decide the density of the reconstruction and reduce the run time or adjust it to particular requirements.

D. Depth map fusion.

The previous step produces a set of incomplete depth maps with confidence values. In the following step, we merge them into a single surface mesh representation. To do that merging, the freely available implementation of the volumetric method of Curless and Levoy [10,11] is used. This approach was originally developed for merging laser range scans. It converts each depth map into a weighted signed distance volume, takes a sum of these volumes, and extracts a surface at the zero level set.

This merging approach has a number of nice properties that make it particularly appropiate for our algorithm, in particular robustness in the presence of outliers and representation of directional uncertainty. The merging algorithm starts by reconstructing a triangle mesh for each view and downweighting points near depth discontinuities and points seen at grazing angles. These meshes are then scan-converted using per-vertex weights into a volume merging. Outliers consisting of one or two samples are filtered out automatically, because they cannot form triangles in the first phase of the algorithm. Larger handfuls of outliers will be reconstructed as small disconnected surfaces; these surfaces will have low weight, since all the points are near depth discontinuities and are probably not substantiated by other views. They can be eliminated in a postprocessing step by removing low confidence geometry or by extracting the largest connected component. In addition, the approach has been shown to be least squares optimal under certain conditions, particulary assuming uncertainty distributed along sensor lines of sight which by constructions applies to the depth maps from previous step.

III. EXPERIMENTS

We test the algorithm with 9 photographs (Fig. 6) of the Veleta Rock Glacier (Sierra Nevada, Spain) taken from different view points with covergent angles along the Veleta peak like we can see in Fig. 8. The baselines between cameras are in an interval of 10 m and 50 m. We use a digital camera (Canon EOS 5D with a 35 mm lens) and a total station (Topcon GTS-502E) to locate the 9 control points that are being supervised in the rock glacier area (Fig. 7).

Figure 6. 9 images of Veleta rock glacier

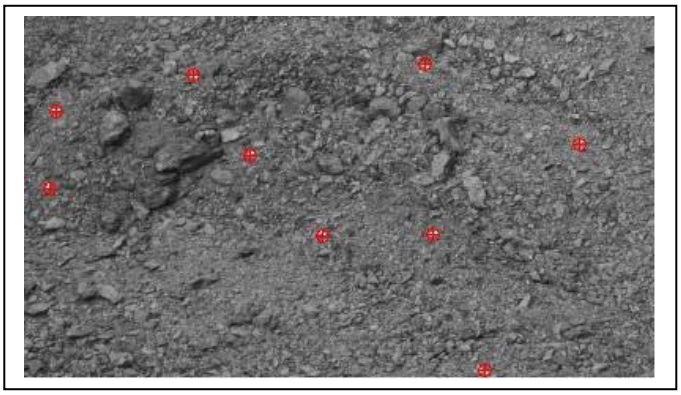

Figure 7. Control points spread around rock glacier

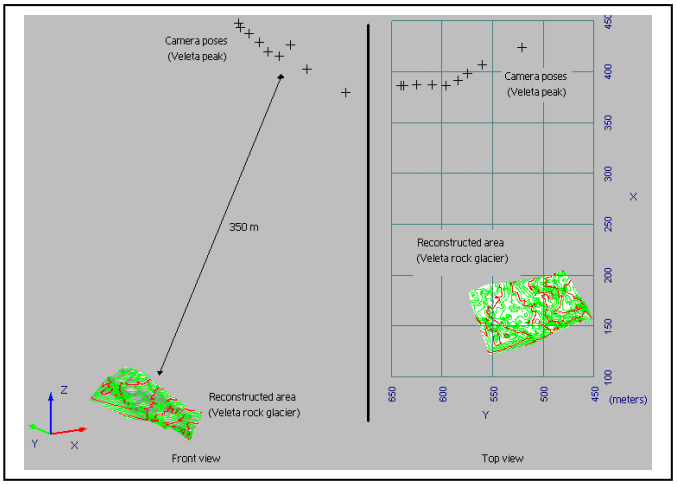

Figure 8. Camera situation respect the reconstructed area

A. *Calibrated camera system.*

We work with a calibrated camera (focal length, principal point, radial distorion) and we only need to calculate the exterior orientation (rotation and translation of cameras). Using bundle adjustment method [20] and the 9 control points (Fig. 7) we calculate exterior orientation. In table 1 the reprojection error (in the control points E1, E2, ..., E9) obtained computing the cameras (P1, P2, ..., P9) are shown.

TABLE I. CAMERA ADJUSTMENT REPROJECTION ERROR (PIXELS)

	Average (in pixels) respect points {E1, …, E9}
P1	0.687
P2	0.434
P3	0.535
P4	0.382
P5	0.529
P6	0.403
P7	0.449
P8	0.425
P9	0.631
Average all cameras (in pixels)	0.497

B. *Feature-based method (Automatic initial terrain surface)*

Althought the calculation of the initial terrain surface it is not the aim of this work it can be interesting know how we get an acceptable surface if we want use it to calculate initial depth estimation.

In this work we implement a feature-based method using SIFT descriptors [12]. Using SIFT we obtain a set of features which will be the basis to get 3D points. SIFT is used in computer vision applications and even in photogrammetry [13]. It allows us find intereting points in 186 cm a image which we can match with points in other image [14]. This method can work with changes in rotation, scale or point of view with very good performance.

Once we obtain these matching points we have to triangulate them (using our known camera system) [19] to get a set of 3D point (≈8000 points). The last step is to convert this 3D point cloud in a surface using a Delaunay triangulation. The result is a surface which we can use for initial depht estimation as we explain in Method description section.

C. *Input parameters value*

Once we know the orientation of the cameras (interior and exterior orientation) and before use the dense reconstruction algorithm we have to choose several extra input parameters:

- Δ_{depth} : depth increment. It is important because it determines the resolution in depth searching. We have worked with values between 0.1 and 1 meter.
- *thresh*: It is the threshold that we use to determine when a NCC value is valid or not. Values between 0.6 and 0.8 are used.
- *k*: number of neighbors. We use 3 or 4 neighbor images, as it is adviced in Goesele [2] work.
- Initial depth estimation. A depth interval for searching the final depth is required by the method. As explained above, we use a first depth estimation using a previous terrain surface. In our case we set a range of 20 meters around the initial depth computed. Without the automatic initial terrain surface improvement (section III.B) we should use a depth interval between 220 meters and 360 meters to obtain comparable results but with ≈ 7 times more computational cost (Veleta rock glacier example).
- 2 levels of pyramidal structure. The first level is double size respect second level.

D. *Results with rock glaciers images*

After the use of algorithm (with Veleta rock glacier images) we have obtained 9 depth maps (one per photograph) with ≈ 200000 points per map. Normally it is difficult calculate quantitatively the error of 3D model because we have not a comparable "real" model. In our example we test the algorithm with a set of additional terrain points (TP1, TP2, ..., TP10) measured with Total Station. Their 3D location is obtained with an accuracy about ± 3-4 cm. It is important to use points that were not used in any step of the method (points of Fig.7 were used in exterior orientation of cameras). In these terrain points we calculate the reprojection error obtained by the dense reconstruction algorithm. In Table II we can see the reprojection error obtained during the calculation of a concrete deph map. In this case we show the reprojection error obtained in the reference image (Img1) and in the 3 neighbors images (Img2, Img3, Img4). As Table II shows, we get little errors (with a maximum of about 1 pixel).

TABLE II. TERRAIN POINTS REPROJECTION ERROR (PIXELS)

	Img1	**Img2**	**Img3**	**Img4**	**Average (pixels)**
TP1	0.05	0.06	0.28	0.58	
TP2	0.05	0.03	0.10	0.22	
TP3	0.06	0.14	0.63	1.20	
TP4	0.01	0.01	0.03	0.07	
TP5	0.05	0.14	0.59	1.14	
TP6	0.08	0.26	1.08	1.20	
TP7	0.07	0.18	0.77	1.17	
TP8	0.08	0.09	0.42	0.85	
TP9	0.10	0.05	0.24	0.50	
TP10	0.02	0.01	0.05	0.10	
Average	0.057	0.097	0.419	0.703	0.319

Beside the results shown in Table II, we have obtained products using as basis the dense reconstruction. In Fig. 9 we show an example of a 2D map of the reconstructed area. Also it is possible generate a video to show the reconstruction. In the url http://robolab.unex.es you can download a video sample of the Veleta rock glacier reconstruction.

IV. CONCLUSIONS

In this work we present a solution for 3D reconstruction of rock glacier surface from images. This is a specific case of 3D modelling with particular properties which motivated us to get a particular 3D dense reconstruction method. It is a good application for multi-view stereo reconstruction algorithms [1], hence we searched the suitable method in this set of algorithms, trying to get a method with easy implementation and using simple scene representations. We propose a method based in the work of Goesele [2] which is suitable for getting a dense 3D rock glacier model as can be seen in experiments section. We use this method adding two important improvements: a feature based approach to obtain a initial depth estimation in an automatic way (adjusting the depth processing), and a better performance using a image pyramidal structure (reducing computation time). Using the implemented algorithm we can obtain a rock glacier reconstructions with a big amount of points with enough accuracy, merging the information of several images. In experiments we can find an example of use in Veleta rock glacier (Sierra Nevada, Spain) with good results using 9 convergent and separated images.

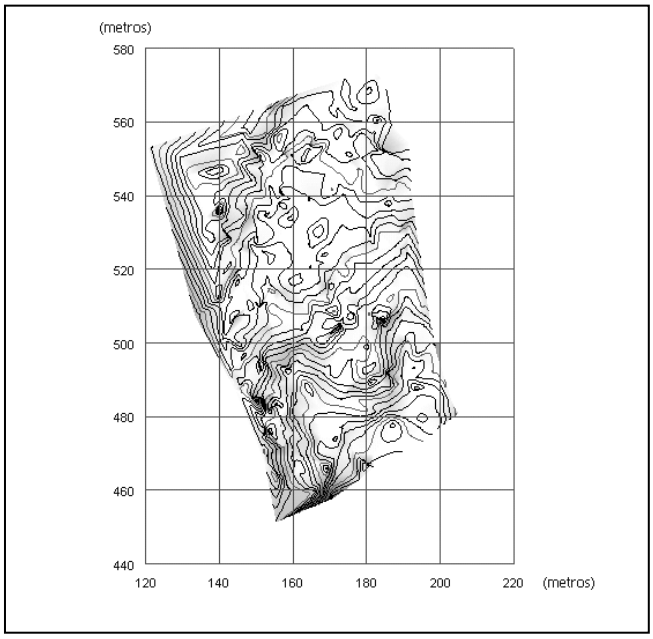

Figure 9. 2D map of rock glacier

ACKNOWLEDGMENT

REFERENCES

[1] S. Seitz et al. Multi-view stereo evaluation web page. http://vision.middlebury.edu/mview/.

[2] M. Goesele, B. Curless, and S. Seitz. Multi-view stereo revisited. In CVPR, 2006.

[3] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In ICCV, pp. 591- 596, 2003.

[4] G. Vogiatzis, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In CVPR, pp. 391-398, 2005.

[5] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In CVPR, vol. II, pp. 822-827, 2005.

[6] T. Yu, N. Xu, and N. Ahuja. Shape and view independent reflectance map from multiple views. In ECCV, pp. 602-616, 2004.

[7] P. Gargallo and P. Sturm. Bayesian 3D modeling from images using multiple depth maps. In CVPR, vol. II, pp. 885-891, 2005.

[8] C. J. Taylor. Surface reconstruction from feature based stereo. In ICCV, pp. 184-190, 2003.

[9] Y. Furukawa and J. Ponce. High-fidelity image-based modeling. Technical Report 2006-02, UIUC, 2006.

[10] Vrippack: Volumetric range image processing package.Available at http://grail.cs.washington.edu/

[11] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In SIGGRAPH, pp 303-312, 1996.

[12] D. Lowe. Distinctive Image Features from Scale-Invariant Features. International Journal of Computer Vision, Vol. 60(2), pp 91-110, 2004.

[13] F. Remodino. Detectors and descriptors for photogrammetric applications. Photogrammetry and Computer Vision, 2006.

[14] J. De Matías, J. Moreno, J.J. Guerrero. Automatización de trabajos fotogramétricos mediante técnicas de Visión Artificial. Congreso Internacional de Matemáticas en la Arquitectura y la Ingeniería, 2007.

[15] J.J. de Sanjosé, A.D.J. Atkinson, F.S. Franch, A. Gómez, J. De Matías. Geodetic and photogrammetric techniques for the cartographic production and the observation of the geomorphic dynamic of the Corral del Veleta active rock glacier (Sierra Nevada, España). International Symposium on High Mountain Remote Sensing Cartography, 2006.

[16] J.J. de Sanjosé, A.D.J. Atkinson, N.A. De Pablo, J.J. Zamorano, D. Palacios, J. De Matías. Cartographic production (1997-2003) of the Popocatépetl (México) and the Ventorrillo glacier for the analysis of hydro volcanic risks. International Symposium on High Mountain Remote Sensing Cartography, 2006.

[17] G. López, J.J. Guerrero, J.J. de Sanjosé, Automatic photogrammetry for the production of geomorphologic cartography. 6th International Conference on Geomorphology, Zaragoza, 2006.

[18] A. Gómez, D. Palacios, M. Ramos, L. Schulte, Location of Permafrost in Marginal Regions: Corral del Veleta, Sierra Nevada, Spain. Permafrost and Periglacial Processes, 12, pp 93-110, 2001.

[19] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision, 2nd Edition. Cambridge University Press.

[20] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. Bundle adjustment – A modern synthesis. Proc. Int. Workshop on Vision Algorithms, pp 298-372, 2000.

[21] J.C. McGlone. Manual of photogrammetry 5th edition. American Society for Photogrammetry and Remote Sensing, 2004.

[22] P. Bachiller, P. Bustos and L.J. Manso. Attentional Selection for Action in Mobile Robots. Advances in Robotics, Automation and Control, ISBN 978-953-7619-16-9.