Study of stereo matching algorithms used in Satellite Stereo Pipeline for the 3D reconstruction of digital elevation models.

BEDDIAF Amir, BERDEAUX Alexandre, BOULAHBAL Houssem, BOURGOIS Astrid and SADOU Loïc

Abstract—In this paper, we study the different stereo matching algorithms used in the software S2P and the possibility of expanding it to build a DEM as a 3D mesh. We offer a comparison of the stereo-matching algorithm following criterias such as robustness to artifacts. Then we explore the state of art of the methods reconstructing a 3D mesh from a point cloud generated by S2P.

Keywords—S2P, stereo-vision, stereo-matching, satellite image processing, mesh, point cloud, surface reconstruction

I. Introduction

With many satellite observation missions in orbit, there is a great abundance of material allowing accurate three dimensional reconstruction of earth's surface. From Lidar telemetry to optic stereoscopic imaging, reconstruction data can be obtained from different methods, resolutions and qualities. Such data may not be easily available to potential users but commercial distribution of image from optic satellite like Pleïades or Ikonos allows for more accessible use of stereo-vision methods for the construction of a digital surface model (DSM). To this end we use satellite stereo pipeline (S2P) [1] a software that allow the extraction of a DSM from satellites images and their corresponding rational polynomial coefficient (RPC) in separate xml files.

In the field of stereo-vision satellite images come with additional challenges that we need to overcome before applying standard stereo-matching algorithm. For instance stereo-rectification cannot be directly applied because the epipolar curve of both images are not conjugate meaning that the epipolar curve of a view can correspond to multiple epipolar curves in the other image [2]. This problem is handled by dividing the images in tiles and then makes a stereo-rectification on said tiles using affine camera approximation [2].

But a RPC is a simplified and approximate way of modeling the satellite camera obtained through estimation of the coefficients causing an offset between epipolar curve and corresponding pixel. [3]. As a consequence S2P must correct the images to ensure that the epipolar line associated with a view cross the corresponding pixels in the rectified image of the other one [3]. Once those problem solved, it uses a stereo-matching algorithm chosen by the user to extract a DSM of each tiles as a 3D point cloud [1].

In this paper we will seek to give direction on the way to use S2P in order to build a DSM in the form of a 3D surface mesh. Seeing as the software already handles the specific difficulties coming from the use of satellite images we will focus our approach on two major choice. First the choice of a stereo matching algorithm to build a 3D point cloud, to this end we will offer a review of stereo matching algorithms supplied by S2P. We will then offer a short survey of available reconstruction method to process the point cloud into a surface mesh.

II. Stereo Matching

A. General concepts

The 2001 survey of Scharstein and Szeliski on Two-Frame Stereo correspondence [4] offers a taxonomy to classify stereo-matching algorithms in two classes, local matching based on evaluation of a support region and global matching that consider finding disparities as a 2D optimization problem. It also identifies components that comprise a stereo-matching algorithm such as the matching cost function, the aggregation of cost and the computation of disparity value [4].

Matching cost function associates a pixel coordinate and a disparity, this create a space that we will try to separate on each pixel coordinate with a disparity of minimal cost. There is a great number of potential matching cost from ones using intensity differences, such as Sum of Squared Difference (SSD) [4], to matching specialized in handling specifics imaging problem like sampling like in [5].

Computing disparities with a local methods involve to sweep the secondary image with a patch centered on each pixel x' like :

$$x' = x + d \tag{1}$$

Where d is the disparity value for each pixel x' and x is the pixel in the reference image which we are looking for a correspondence. The disparity $d(x)$ is the value of d that provide the minimum value of a cost function for each pixel x' in the secondary image [6].

There are two difficulties during the disparity computation with a local algorithm:

- Matching ambiguities: The presence of a repetitive pattern in both images may cause two region in different place to yield similar results. This similarity can cause matching errors where a region of the reference is matched to a region with a similar pattern but at the wrong place [6].
- Depth discontinuities: This occurs when there is a big depth difference between an object and its background. In this case there is a high chance of seeing occlusion or dis-occlusion (Fig 1), meaning that the point is only

seen by one of the two views. This can challenge local method because they rely on computing correspondences between patches meaning that an occluded area will cause the algorithm to make mismatches [6]. Dis-occlusion can also cause a bias in favor of the foreground disparity at it boundary creating an effect of "foreground flattening" [6].

A reliable local algorithm have to deal with these two main difficulties for decent results.

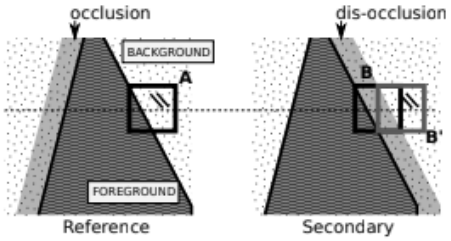

Fig. 1: Sketch showing occlusion and dis-occlusion situations
Source: [6]

On the other hand global method do not use a window of neighbors to match pixels but formulate the problem as an energy minimization problem. We then try to find a disparity function that minimize global energy presented by the sum of the costs of pixels in the image and a smoothness term that usually measure the disparity difference of neighbors in a function ρ [4]. This optimization being made on a two-dimensional space, it can prove itself to be NP-hard. Consequently a subclass of algorithm can be found, based on the use of dynamic programming along a one dimensional scan-line, however this change can result in streaking because of problems in enforcing inter-scan-line consistency [4]. It must be said that in a rectified image if a best match exist it is guaranteed to be inside the scan-line [2].

B. Stereo-matching algorithm used in S2P

1) Total Variation minimization with L^1 Norm (TV-L1): It is a method based on the minimization of a function which contains a data term using the L1 norm and a regularization term using the total variation of the flow. This formulation allows discontinuities in the flow field, while being more robust to noise than the classical approach by Horn and Schunck [7], [8].

This algorithm (Fig 2) takes 9 parameters. I_0 and I_1 are the images; u^0 is given by the enclosing multi scale procedure and it is zero at the coarsest level; τ is the time step with default value 0.15; λ is the data attachment weight with default value 0.15; θ is the tightness with default value 0.3; ε is the stopping criterion threshold with default value 0.01; N_{scales} and N_{warps} are respectively the number of scales and the number of warps with default value 5 [8].

This algorithm allows to detect the displacement of the objects in a scene like in (Fig 3). A small values of Îż return a smoother solution with underestimate optical flow for the moving objects. However, big values of Îż return a

Procedure TV-L^1_optical_flow($I_0, I_1, u^0, \tau, \lambda, \theta, \varepsilon, N_{maxiter}, N_{warps}$)

$\mathbf{p}_1 \leftarrow (0,0)$
$\mathbf{p}_2 \leftarrow (0,0)$
for $w \leftarrow 1$ **to** N_{warps} **do**
  Compute $I_1(\mathbf{x}+\mathbf{u}^0(\mathbf{x})), \nabla I_1(\mathbf{x}+\mathbf{u}^0(\mathbf{x}))$ using bicubic interpolation
  $n \leftarrow 0$
  while $n < N_{maxiter}$ **and** *stopping_criterion* $> \varepsilon$ **do**
    $\mathbf{v} \leftarrow TH(\mathbf{u}, \mathbf{u}^0)$
    $\mathbf{u} \leftarrow \mathbf{v} + \theta \mathrm{div}(\mathbf{p})$
    $\mathbf{p} \leftarrow \frac{\mathbf{p}+\tau/\theta \nabla \mathbf{u}}{1+\tau/\theta|\nabla \mathbf{u}|}$
    $n \leftarrow n+1$
  end
end

Fig. 2: TV-L1 algorithm by Janvier Sánchez and co
Source: [8]

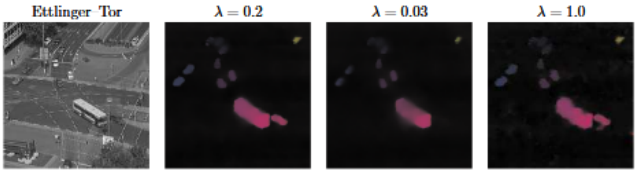

Fig. 3: Ettlinger-Tor example with different value of λ
Source: [8]

more important attachment term. This doubled with increased sensitivity to noise, resulting in unstable flow fields [8]. This algorithm can detect geometric shapes in a scene as well as their displacements. We do not have result for satellite images with this algorithm.

2) Multi-Scale Multi-Window (MSMW): This method try to handle the two previously discussed shortcomings of local method. To this end it introduce two features, adaptive windows to help with depth discontinuities and multiple scale to limit confusion from repetitive pattern [6].

Instead of sweeping the secondary image with a $n \times n$ patch which compute all its n^2 pixels, we use several patch shapes. The third image of (Fig 4) shows, for some points, the windows giving the best disparity, it demonstrates a real adaptation to the discontinuities borders [6]. Analyzing the context of pixels using only the pixel in the "same depth" shows very good results for depth discontinuities and also for slanted surface [6].

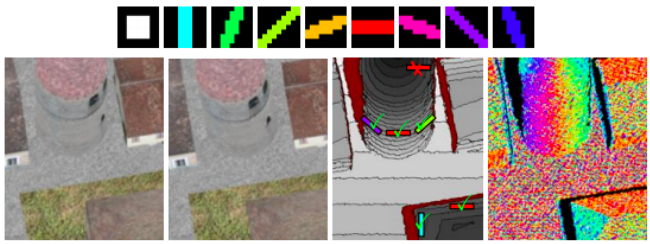

Fig. 4: Images showing the window shape used to compute the best disparity for some points
Source: [6]

In order to deal with confusion arising from repetitive pattern it applies a scale pyramid, each level is determined by

sub-sampling and blur (convolution with a Gaussian kernel) the previous level [6]. Thus, the final level shows only low frequency of the originals images giving coarse information and then eliminate noise. Executing stereo matching on a specific level restrict the search area at the previous level then it reduce the computation time and the probability of analyzing repetitive patterns [6].

3) Semi-global Matching (SGM): As its name implies SGM propose a method taking from both local and global matching techniques, it also make use of dynamic programming in its disparity computation.

- Pixelwise Matching Cost Calculation : In order to determine matching cost, SGM uses a derivative of the Mutual Information criterion proposed by [9]. The general idea is to use the entropy between two images and mostly joint entropy, the more two images match the lower the cross entropy and the higher the mutual information [10]. This allows to explain stereo matching as finding the disparity that minimize cross-entropy [10]. SGM implements the HMI method (hierarchical MI computation), its implementation is detailed in Hirschmuller's works [10].
- Cost Aggregation : Like a global method SGM considers finding the disparity image as a energy minimization problem. However noise, outliers and other artifact can affect the pixel wise cost, in order to minimize the effect of such artifact a smoothness constraint is taken into account [10].
  This gives the following formulation of energy [10]:

$$E(D) = \sum_p \left(C(p, D_p) + \sum_{q \in Np} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \right) \quad (2)$$

  Where $C(p, D_p)$ is the matching cost of the affectation of the disparity D_p to the pixel$p$ and P_1, P_2 are constant penalties respectively for small (1 pixel) and large disparity with the pixel neighborhood N_p [10].

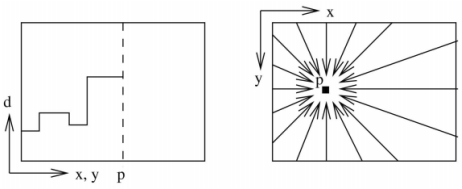

Fig. 5: Cost Aggregation in disparity space
Source: [10]

Optimization of this function in the 2D space is an NP-complete, but the problem is simplified as a 1D optimization by using dynamic programming on scan lines. This kind of solution can bring streaking artifact, to handle this matter SGM sums dynamic programming results from all directions around the pixel being matched [10]. Path are defined by the following equation with r as the direction, d the disparity, and p the pixel being matched [10]:

$$\begin{aligned} L_r(p,d) = & C(p,d) + \min(L_r(p-r,d), \\ & L_r(p-r,d-1) + P_1, \\ & L_r(p-r,d+1) + P_1, \\ & \min_i L_r(p-r,i) + P_2) - \min_k L_r(p-r,k). \end{aligned} \quad (3)$$

- Disparity Computation : Computing the disparity images D_b and D_m for the pair of base and match (represents the match image of I_b) images is based on the same method used in local stereo methods [10]. The disparity d is therefore the one corresponding to the minimal sum of dynamic programming results from all directions around the pixel in the match image [10]. Disparities images are filtered using median filter with a 3×3 dimension of for removal of artifacts present on the disparity images [10]. When D_b and D_m are computed the following consistency check can be made, where bp is a pixel in the base image and mq its match in the other [10]:

$$D_p = \begin{cases} D_{bp} \quad if \quad |D_{bp} - D_{mq}| <= 1 \\ invalid \quad otherwise \end{cases}$$

  This check allows to determine certain artifact like occlusion or false-match and enforces the uniqueness constraint on mapping pairs [10].
- Disparity refinement Resulting disparity images may always presents some artifacts or invalid disparities. SGM deal with errors such as discontinuities due to textures, noise, holes or outliers by filtering peaks, interpolating occlusions and mismatches which called Discontinuity Preserving Interpolation [10].

4) Semiglobal Block Matching (SGBM): SGBM is a modified implementation of SGM, as pointed out in its name it execute matching between blocks of images instead of pixel by pixel matches, it cuts the number of direction computed for cost aggregation to 5 [11] and changes the cost function to the metric detailed in [5].

5) More Global Matching (MGM): MGM algorithm is an approximation of global methods for stereo matching driven from SGM algorithm. It uses the same pixel-wise cost calculation and disparity computation but differ in its aggregation of costs in order to treat streaking artifacts present in SGM [12].

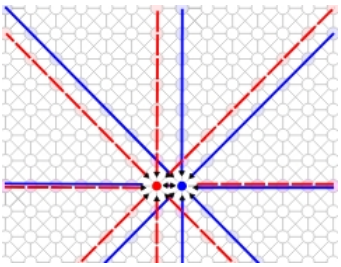

Fig. 6: Star-shaped graphs associated in SGM to two adjacent pixel
Source: [12]

Streaking artifact in SGM comes from a lack of relation between the optimization of neighboring pixels, causing a weak enforcement of smoothness constraint [12]. To solve this, information from the 2D problem is injected into the 1D path by taking into account paths computed by the previous scan-line. This change can be translated as searching from the path to the pixel above the one being computed relative to the direction followed by the DP step [12].

We indicate this with the direction $r^{\perp}$, this change the aggregation equation (3) into [12]:

$$\begin{aligned} L_r(p,d) =& C(p,d) + \sum_{x \in [r,r^{\perp}]} \frac{1}{2} \min(L_r(p-x,d), \\ & L_r(p-x,d-1) + P_1, \\ & L_r(p-x,d+1) + P_1, \\ & \min_i L_r(p-x,i) + P_2) - \min_k L_r(p-x,k) \end{aligned} \tag{4}$$

It introduces a dependency between scanline, preventing parallel processing of scan-lines in the way done by SGM. Parallelization in MGM is achieved diagonal-by-diagonal [12].

C. Comparison and conclusion

In order to compare the different methods previously introduced, we display point clouds of the same tile returned by different methods with the software MatLab (Fig 7). We will test MGM, MSMW, TV-L1 and a multi-scale version of MGM, we argue that MGM, SGM and SGBM sharing the same base principle we can regroup them under MGM.

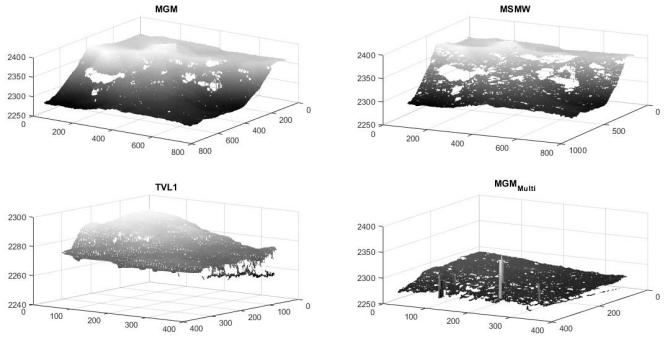

Fig. 7: DSM obtained by each algorithms shown with Matlab From left to right, top to bottom : MGM, MSMW, TV-L1, a multi window version of MGM

Our test images is the one supplied with s2p, it is a satellite stereo view of a rocky mountain side near the Piton de la Fournaise at the Reunion in France. In the result of (Fig 7) MGM proves itself to give the cloud point of highest quality. There is less missing data, that can be seen with blank space on the surface and represent failed match, and represent the desired shape more faithfully than TV-L1.

This test tend to favor MGM over others algorithm, however we can summarize a few more argument for the algorithm. For a start, in the case of satellite imaging local method are inherently at a handicap when compared with global method because of the matching ambiguities problem exposed earlier. As man architectural object like a facade are noted as prone to cause such issue [6] it is likely that a local algorithm yield peculiar poor result in an urban area where depending such features are common. Furthermore things like intersection mostly in grid based city, homogeneous buildings or terrain (relatively flat and homogeneous in radiometry) make all occasion for an algorithm. More associated with urban area is the presence of many depth discontinuities (buildings and street) that give occasion for occlusion or dis-occlusion problem, this can also happen with steep terrain. If in this case MSMW yielded decent result it must be reminded that the test sample did not offer many of the features that weaken local algorithm. Finally in the case of TV-L1, the technique is meant to detect form and motion rather than build disparity map to create a geometric model of the object which can explain the poor performance observed and make it ill-suited for our purpose.

III. 3D RECONSTRUCTION

A. Introduction

With S2P our stereo-matching is already transformed into a digital surface model as a 3D point cloud. We will now look into using this cloud point in order to reconstruct the surface as a surface mesh.

Our point cloud give us information about the geometry of the surface, we can however distinguish three main sources of error caused by previous step.

- There can be faults in the images acquisition.
- There can be failure in the matching
- There can be errors in the computation of point positions

This mean that a point cloud obtained through S2P is likely to contain artifacts.

Using the list of artifacts identified in [13], we bring focus to three artifact that we consider especially relevant in our context.

- Sampling error: The distribution of point sampling is not uniform. The points generated by stereo matching algorithms can produce a non-uniform sampling patterns that may be due to the orientation of the scanner as well as to the geometry of the observed shape [13].
- Noise: Points that are randomly distributed near the surface distribution of the noise. It is a function of sensor noise, depth quantification and distance or orientation of the surface from the scanner [13]. It must also be brought to attention that S2P do not supply its point cloud normals, meaning that to use a reconstruction scheme depending on them a way to estimate the normals must be chosen. With this step, it means that the noise can indirectly and strongly impact a reconstruction by skewing the estimated normals.
- Outliers : Points that are away from the original surface are classified as outliers. Outliers are due to structural artifacts in the acquisition process. In some cases, outliers are distributed randomly in the volume, where their density is lower than the density of the points sampling the surface [13].
- Missing Data : In our case missing data is particularly likely from match failure in the stereo-matching process. Many factors can lead to missing data such as limited

sensor range, high light absorption and occlusions in the scanning process where large parts of the shape are not sampled [13].

Surface reconstruction is the procedure of automatically generating 3D surfaces from a discrete sampling of real object. The general approach used can be divided in two classes discrete or continuous [14]

- Discrete: uses the could points directly to regenerate the surface
- Continuous : takes either a surface fitting or a function fitting approach

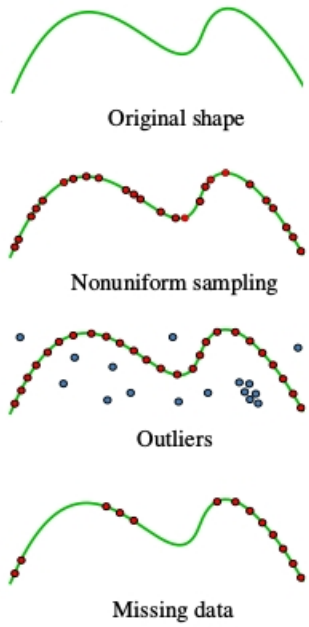

Fig. 8: Different forms of point cloud artifacts
Source: [13]

B. Greedy projection method

This method was introduced by: Zoltan Csaba Marton and Radu Bogdan Rusu and Michael Beetz. As this method is real-time, memory efficient and adaptable to various densities.thus, this method is implemented in "PCL" [15] and used widely in robotics. The input is an unorganized 3D point clouds which represents the observed scene. The constraints of water-tightness and completeness are not considered [16].

this method can be summarized into four steps [16]:

- Select k neighborhood points for each point p
- Compute the tangential plane
- Project the selected points to the tangent plane
- The triangulation is done by connecting the k points to p

1) Neighborhood selection: The estimation of the neighborhood is problematic as the densities may varies even in a single disparity map. This algorithm adapt the selecting using a sphere with radius $r = \mu_0 d$ where d is the distance the nearest neighborhood and μ_0 is a user defined parameter used to speed up/down the computations in steps 2,3 and 4 [16].

2) Tangential plane computation: Due to the noise present in the the disparity estimation. The tangential plane is not computed using an estimation of the normal. The approach used is a weighted least squares plane [16].
let C be the weighted co-variance matrix :

$$C = \sum_{i=1}^{k} \epsilon_i (p_i - \bar{p_i})^T . (p_i - \bar{p_i})$$

The eigenvector of the smallest eigenvalue correspond to the normal vector. $\epsilon_i = exp(\frac{-d_i^2}{\mu^2})$ Îij is the mean distance from the query point p to all its neighbors p_i and d is the distance from the query point p a neighbor p_i [16].

3) Projecting and triangulation: We project each point and its neighborhood to the plane calculated. The triangulation is done by connecting the vertices.

C. Power Crust 3D reconstruction

Power crust was introduced by : Nina Amenta, Sunghee Choi and Ravi Krishna Kolluri. It uses medial axis transform to approximate the object then an inverse transform is used to produce the surface representation [17].

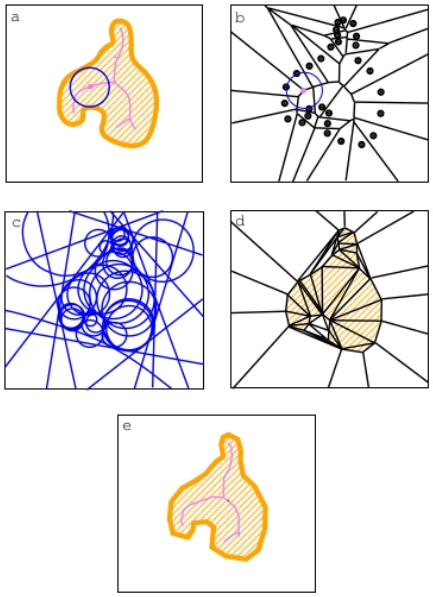

Fig. 9: a: 2D representation of the shape being reconstructed. b: Voronoi diagram with a pole. c: Power diagram. d: delaunay triangulation inside the shape. e: the reconstructed shape .
Source : [17]

1) Geometric definitions:

- The ball in a power diagram context: a ball in the power diagram is represented by its center c and weight ρ^2 where ρ represent its radius [17].
- Medial Ball: A medial ball is the maximal empty ball that's not contained in no other empty balls [17].
- Medial axis transform : it is the set of all center of medial balls [17].
- Power distance $d_{pow}(B_1, B_2) = d^2(c1, c2) - \rho_1^2 - \rho_2^2$ [17].
- Power diagram : Subdivision of the closed space that represent the scene into cells each cell consists of x closest point in d_pow to some vertex $v \in Voronoi$ [17].

```
Label_Poles() {
    For all poles p,
        initialize in(p) = out(p) = 0.
        insert p in the queue.
    For each pole p adjacent to points of Z,
        out(p) = 1.
        Update_Priority(p)
    while (queue is not empty) {
        Remove the top element p of the priority queue
        If in(p) > out(p), label(p) = in, tmp(p) = in(p)
        Otherwise, label(p) = out, tmp(p) = out(p)
        For each sample s of which p is the pole,
            let q be the other pole of s,
            opp(label(p))(q) = max(tmp(p) * w_pq, opp(label(p))(q))
    /* opp(in) = out, opp(out) = in, w_pq = -cos(∠psq)*/
            Update_Priority(q)
        For each deeply intersecting neighboring poles q,
            (label(p))(q) = max(tmp(p) * w_pq, (label(p))(q))
    /* w_pq = -cos(α), α is angle between balls p and q*/
            Update_Priority(q)
    }
}

Update_Priority(pole p) {
    If in(p) > 0 and out(p) > 0, pri(p) = |in(p) - out(p)| - 1.
    Otherwise, pri(p) = max(in(p), out(p)).
}
```

Fig. 10: Labelling algorithm.
Source: [17]

2) Algorithm pipeline: this algorithm can be summarized as follows [17]:

- For each point p = (x,y,z) Compute the Voronoi diagram the software *Hull* is used in the original paper.
- Compute the poles of each point p
- Compute the power diagram
- Label each pole as either inside or outside.
- Output the power diagram faces separating the cells of inside and outside pole
- Output the triangulation faces connecting inside poles as the power shape.

3) Selecting poles: After computing the Voronoi diagram for each sample point s we select the farthest Voronoi vertex p_1 which is the first pole of s. We compute the furthest Voronoi vertex p_2 such as $s\vec{p}_1.s\vec{p}_2 = 0$ [17].

4) Labelling algorithm: After computing the poles we calculate the Power diagram of all poles. We uses the following algorithm (Fig.10) to label the poles as either inside or outside.

We Output the regular triangulation faces connecting inside poles as the power shape [17].

D. Poisson 3D Surface Reconstruction Method

Poisson 3D surface reconstruction, an approach introduced by Michael Kazdhan, Matthew Bolitho and Hugues Hoppe, widely used in mesh processing [18].

This method is implemented in various surface reconstruction libraries like "PCL" [19] and "CGAL" [20] and considered as one of the best algorithms due to its efficiency, where it gives a watertight smooth surface with good details representation.

The method based on computing a 3D indicator function called χ which discriminates the interior of an object from its exterior. The challenge is how to accurately compute the indicator function from a set of points, with inward faced normals $\vec{N}$ assumed to lie on or near the surface of the object,and finally approximate the isosurface. The Poisson surface reconstruction method requires an oriented set of point cloud, uniformly or non-uniformly distributed, and it's robust facing noisy data [18].

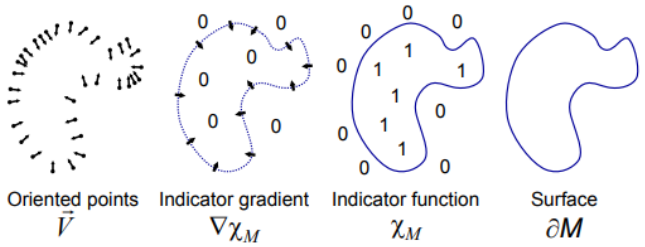

Fig. 11: Poisson method
Source: [18]

This method can be resumed in three steps [18]:

- Approximation of the indicator function's gradient where the set of input points provides enough information to approximate the integral relation between the gradient of the smoothed indicator function and the surface normal field to a discrete summation .
- Derive the indicator function from the gradient.
- Extract the surface by using methods like the marching cube algorithm (octree data structure).

1) Comparison of Poisson method to previous works: Interpolation based methods like power-crust,robust cocone shown their limitation with noisy point clouds where they generate noisy surfaces, so for the quality of reconstructed surfaces with noisy data Poisson can be compared only to VRIP and FFT based approach which considered as non-interpolate methods [18].

2) Limitations: Poisson method showed its limitation for surfaces with big gaps between two regions, where it connect those regions, generates errors in reconstructed surface, unlike the VRIP method that proved its robustness facing this problem, tests and results are shown in K.H works [18].

E. Smooth Reconstruction Method (SRM)

SRM uses three step to process a point cloud obtained through stereo-matching into a surface mesh. It pre-processes the point cloud to eliminate duplicates and outliers [21]. It merges point too close to neighbors for the first case and uses a cone angle criterion to see if the point was acquired by too few viewpoints or ones that are too close from each others . It then smoothes the cloud using a method described in [22] [21].

It uses the images used for the stereo matching to extract information on the contours of the scenes. The point are projected into the image plane of their views and a Delaunay Triangulation is applied to them [21]. The edges are fitted to

the image contours with a weighting algorithm [23]. The set of segment obtained is used in a 2D constrained Delaunay Triangulation which will is projected back into the 3D space to obtain a triangle soup [21].

The triangle soup is then refined by applying visibility constraint between the points and the viewpoint to filter triangle [21]. Triangle whose vertices are only grazed by the different viewpoints are deleted [21]. A photo-consistency check is applied to big triangles by projecting them into the views image these region are then scored using the average of their Normalized Cross Correlation [21]. Once processed the triangle soup can finally be turned into a mesh using a Delaunay refinement method [21].

F. A TV Prior for High-Quality Scalable Multi-View Stereo Reconstruction

This method was introduced by :Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein and Helmut Mayer. This method is used to reconstruct 3D models from multiple high resolution images. This method assumes that 3D reconstruction can be improved by modeling the uncertainties of disparity maps [24].

1) Algorithm pipeline: this algorithm can be summarized as follows [24]:

- Computation of disparity maps using the SGM method [10].
- For each disparity we compute the estimation quality.
- Propagation of discrete 1D probability functions on the lines of sight into 3D space.
- Optimization of points on the surface based on the probability function.
- Filtering by visibility checks considering probabilistic information.
- Local triangulation [25] of the optimized point cloud.

2) Disparity Quality Modeling: This section discuss the methods used to model the uncertainties of the disparities obtained by SGM [10]. The quality of the estimation depends on three factors: stereo error model, uncertainties in stereo matching and learning error distributions [24].

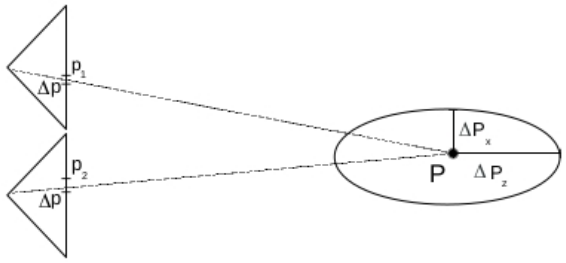

Fig. 12: Pair of cameras with baseline t and focal f the disparity uncertainty Δp leads to an ellipsoidal uncertainty
Source: [24]

3) Stereo Error Model: The error of the x and y direction of the camera coordinates *fig.12* can be modeled as [24]:

$$\Delta P_x = \Delta p \frac{P_z}{ft}\sqrt{(t-P_x)^2+P_x^2}$$
$$\Delta P_y = \Delta p \frac{P_z}{ft}\sqrt{2P_y^2+\frac{t^2}{2}}\ \Delta P_z = \Delta p \frac{P_z}{ft}\sqrt{2}$$

TV class	1	2	3	4	5	6	7
μ_{EM}	0.98	0.48	0.11	0.04	0.03	0.03	0
σ_{EM}	4.44	3.11	1.65	1.07	0.67	0.50	0.40
TV class	8	9	10	11	12	13	14
μ_{EM}	-0.03	-0.03	-0.03	-0.03	-0.03	-0.02	-0.02
σ_{EM}	0.33	0.34	0.34	0.30	0.28	0.26	0.24
TV class	15	16	17	18	19	20	
μ_{EM}	-0.02	-0.01	0	0.01	0.01	-0.01	
σ_{EM}	0.22	0.22	0.21	0.20	0.19	0.18	

Fig. 13: Estimated error for the 20 classes
Source: [24]

4) Classification of Error Levels Based on TV Features: Disparity maps show oscillations with unknown frequency. To model the local oscillation a feature classes based on total variation is introduced. The classes represent the disparity quality in a stable way. Thus for a pixel y, d is expressed over a 2D neighborhood $\mathcal{N}_y$ as [24]:

$$TV(y) = \sum_{(i,j)\in\mathcal{N}_y}\sqrt{(d_{i+1,j}-d_{i,j})^2+(d_{i,j+1}-d_{i,j})^2}$$

And $TV(y)$ represent the degree of local oscillations. To define the TV class, an increasing radius square window with some threshold τ is used as follows [24]:

$$argmax_n(\sum_{m=1}^{n} TV_{i,j\in x_m})$$

x_m represent a square ring shaped with radius m and $|x_m| = 8$. The first step is to compute The TV with n=1 if it exceeds the threshold the class is 1, else we keep calculating TV with increasing n ($n \leq 20$) until it exceeds the threshold and the class is n=k [24].

5) Learning Error Distributions: They assume that the error for each case follow a combination of Gaussian $\mathcal{N}(\mu_n, \sigma_n)$ Where n is the TV class [24]. The Gaussian is estimated for all classes $0 < n \leq 20$ by an Expectation Maximization method. The priors used for this calculation are derived using the 2014 Middlebury stereo datasets with accurate ground truth [26]. The results are shown in (Fig 13)

6) Volumetric Modeling: In summery the expected noise is modeled as [24]:

$$p(d_i^x) = \mathcal{N}(P_z, (\sigma_{EM}\frac{P_z}{ft}\sqrt{2})) \text{ with } P_z = \frac{ft}{d_i+\mu_{EM}}$$

7) Choice of voxel size: As in our case an octree corresponds to the voxels. The algorithm chooses the voxel individually. The voxel size v_s is chosen as follows [24]:

$$\sigma < av_s < 2\sigma$$

8) Propagation into Probabilistic Space: The method proposed by Curless and Levoy [27] is used. However instead of using a linear signed function we propagate using Gaussian cumulative distribution function. Let $p(v_i^0)$ and $p(v_i^1)$ be the probability that a voxel is in front or behind the surface, in summary [24]:

$$p(v_i^1) = \mathcal{N}_{cdf}(d_i) \text{ and } p(v_i^0) = 1 - p(v_i^1)$$

The logarithmic space transformation yields [24]:

$$l_i = \log\frac{p(v_i^1)}{p(v_i^0)} = \sum_j log\frac{p(v_{ij}^1)}{1-p(v_{ij}^1)}$$

The propagation is calculated as follows [24]:

$$l_{i+1} = l_i + log\frac{p(v_i^1)}{1-p(v_i^1)},\ l_0 = 0$$

9) Optimization of Point Positions on the Surface: With I the search interval, the voxel that maximizes front and behind probability of a voxel and its neighbors are [24]:

$$i^* = argmax_{i \in I}(p(v_i^0)p(v_{i+1}^1)$$

d^* is obtained using [24]:

$$\frac{1}{4}\sum_{j=i^*-1}^{i^*+1} d_j|0.5 - p_j|\ ,\ p_j = \frac{e^{lj}}{1+e^{lj}}$$

10) Filtering Voxels using Visibility Checks: Consistency is checked by filtering points that have small front probabilty and Triangulation is done using Streaming surface reconstruction from real time 3D measurements. [25]

IV. Conclusion

In order to move from a pair of satellite image to a 3D reconstruction of earth's surface using S2P, we identified two main choice to be made.

In this survey, we explored a short selection of stereo-matching method supplied with S2P, covering the general functioning of a stereo-matching. We have displayed the specific working of algorithm present inside S2P, before exposing some element to help us choose a method.

With the necessary tools to obtain a 3D point cloud. We have explored the meshing pipeline. At first by setting light on the way the extracted point cloud could have artifacts that would influence the quality of the reconstruction. Then we proposed a short selection of reconstruction method available to transform a point cloud into a surface mesh.

References

[1] 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*, ZÃijrich, Switzerland, Sep. 2014. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01109526

[2] C. d. Franchis, E. Meinhardt-Llopis, J. Michel, J. Morel, and G. Facciolo, "On stereo-rectification of pushbroom images," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5447–5451.

[3] ——, "Automatic sensor orientation refinement of PlÃl'iades stereo images," in *2014 IEEE Geoscience and Remote Sensing Symposium*, Jul. 2014, pp. 1639–1642.

[4] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47, pp. 5–12, 2001.

[5] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, Apr. 1998.

[6] A. Buades and G. Facciolo, "Reliable Multiscale and Multiwindow Stereo Matching," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 888–915, Jan. 2015. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/140984269

[7] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, Aug. 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370281900242

[8] J. S. PÃl'rez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, vol. 3, pp. 137–150, Jul. 2013. [Online]. Available: http://www.ipol.im/pub/art/2013/26/

[9] Junhwan Kim, Kolmogorov, and Zabih, "Visual correspondence using energy minimization and mutual information," in *Proceedings Ninth IEEE International Conference on Computer Vision*. Nice, France: IEEE, 2003, pp. 1033–1040 vol.2. [Online]. Available: http://ieeexplore.ieee.org/document/1238463/

[10] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[11] "OpenCV: cv::stereo::StereoBinarySGBM Class Reference." [Online]. Available: https://docs.opencv.org/3.4.0/d1/d9f/classcv_1_1stereo_1_1StereoBinarySGBM.html#details

[12] G. Facciolo, C. d. Franchis, and E. Meinhardt, "MGM: A Significantly More Global Matching for Stereovision," in *BMVC*, 2015.

[13] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva, "State of the Art in Surface Reconstruction from Point Clouds," *EUROGRAPHICS star reports*, vol. 1, no. 1, pp. 161–185, Apr. 2014. [Online]. Available: https://hal.inria.fr/hal-01017700/document

[14] M. G. Bolitho, *The Reconstruction of Large Three-dimensional Meshes*. Johns Hopkins University, 2010.

[15] "Documentation - Point Cloud Library (PCL)." [Online]. Available: http://pointclouds.org/documentation/tutorials/greedy_projection.php

[16] Z. C. Marton, R. B. Rusu, and M. Beetz, "On fast surface reconstruction methods for large and noisy point clouds," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3218–3223.

[17] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust, unions of balls, and the medial axis transform," *Computational Geometry*, vol. 19, no. 2-3, pp. 127–153, 2001.

[18] H. Hoppe, "Poisson Surface Reconstruction and Its Applications," in *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*, ser. SPM '08. New York, NY, USA: ACM, 2008, pp. 10–10. [Online]. Available: http://doi.acm.org/10.1145/1364901.1364904

[19] "Point Cloud Library (PCL): pcl::Poisson< PointNT > Class Template Reference." [Online]. Available: http://docs.pointclouds.org/trunk/classpcl_1_1_poisson.html

[20] "CGAL 4.13 - Poisson Surface Reconstruction: User Manual." [Online]. Available: https://doc.cgal.org/latest/Poisson_surface_reconstruction_3/index.html

[21] N. Salman, "From 3d point clouds to feature preserving meshes," phdthesis, UniversitÃl' Nice Sophia Antipolis, Dec. 2010. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00536984/document

[22] F. Cazals and M. Pouget, "Estimating Differential Quantities Using Polynomial Fitting of Osculating Jets," in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 177–187. [Online]. Available: http://dl.acm.org/citation.cfm?id=882370.882394

[23] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, Mar. 1970. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022283670900574

[24] A. Kuhn, H. Hirschmüller, D. Scharstein, and H. Mayer, "A tv prior for high-quality scalable multi-view stereo reconstruction," *International Journal of Computer Vision*, vol. 124, no. 1, pp. 2–17, 2017.

[25] T. Bodenmueller, "Streaming surface reconstruction from real time 3d measurements," Ph.D. dissertation, Technische Universität München, 2009.

[26] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 31–42.

[27] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.