Signal Processing: *Image Communication* 40 (2016) 97–110

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

From stereoscopic images to semi-regular meshes

Jean-Luc Peyrot *, Frédéric Payan, Marc Antonini

Laboratory I3S - University Nice - Sophia Antipolis and CNRS - UMR 7271, France

ARTICLE INFO

Article history:
Received 28 February 2015
Received in revised form
26 September 2015
Accepted 10 November 2015
Available online 27 November 2015

Keywords:
Semi-regular mesh
3D reconstruction
Stereoscopy
Acquisition
Poisson-disk sampling
GPU

ABSTRACT

The pipeline to get the semi-regular mesh of a specific physical object is long and fastidious: physical acquisition (creating a dense point cloud), cleaning/meshing (creating an irregular triangle mesh), and semi-regular remeshing. Moreover, these three stages are generally independent, and processed successively by different tools. To overcome this issue, we propose in this paper a new framework to design semi-regular meshes directly from stereoscopic images. Our semi-regular reconstruction technique first creates a base mesh by using a feature-preserving sampling on the stereoscopic images. Afterwards, this base mesh is passed to a coarse-to-fine meshing process to get the semi-regular mesh of the original surface. Experimental results prove the reliability and the accuracy of our approach in terms of shape fidelity, compactness, but also runtime, since many steps have been parallelized on the GPU.

1. Introduction

Motivated by the high fidelity and the realism of the numerical models, and supported by the increasing storage capacities, the acquisition devices provide now high resolution meshes, ensuring the preservation of the finest details. Consequently these data are massive, and cannot be easily managed by any workstation or mobile device with limited memory and bandwidth. The semi-regular meshes are a good way to overcome these issues, because of their scalability and their compactness. Indeed, the semi-regular meshes are based on a regular subdivision connectivity, well-suited to display or transmit a mesh at different levels of details. This subdivision connectivity also allows a compact representation since only the connectivity of the lowest level of details is needed to reconstruct the full connectivity. This semi-regular structure is also adapted to multiresolution analysis [10] and wavelet compression [14]. Despite their good properties, the semi-regular meshes are sometimes forsaken by users because they are not provided by current acquisition systems which only provide point clouds. So, if one wants to produce a semi-regular mesh of a specific physical object, the pipeline presented in Fig. 1 must be processed: physical acquisition (creating a dense point cloud), cleaning/meshing (removing redundant points and noise inherent to acquisition process, and creating an irregular triangle mesh), and then semi-regular remeshing [15]. This pipeline is long and fastidious, especially as these three stages are performed independently.

Our original idea is to make the design of semi-regular meshes easier, by simplifying the classical pipeline shown above. This paper, that is an extended version of Peyrot et al. [16], presents a *coarse-to-fine* approach that allows an acquisition system to provide semi-regular meshes as output, thus avoiding a remeshing process. We focused on stereoscopic systems, because stereoscopy is an increasing field of interest in surface reconstruction, due to its rapidity and accuracy.

Our method, depicted in Fig. 2, relies on an analysis of the stereoscopic images to get a base mesh that captures the salient features of the original object, followed by a coarse-to-fine meshing that generates the semi-regular output. The most innovative part of our algorithm is the

* Corresponding author.
E-mail address: peyrot@i3s.unice.fr (J.-L. Peyrot).

http://dx.doi.org/10.1016/j.image.2015.11.004
0923-5965/© 2015 Elsevier B.V. All rights reserved.

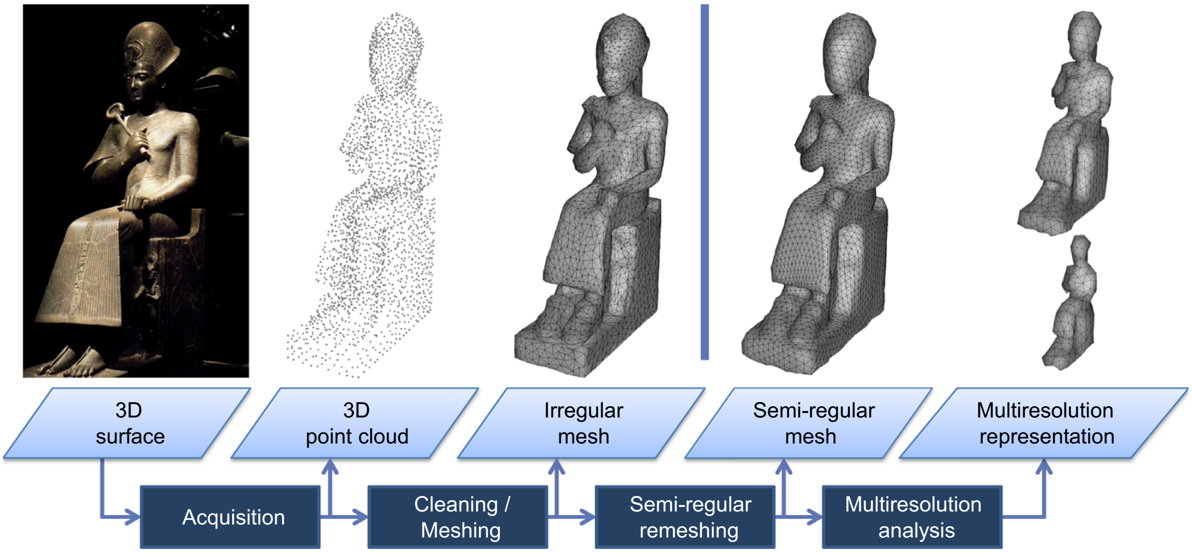


Fig. 1. The pipeline to get a semi-regular mesh from a physical object, and its application to multiresolution analysis.

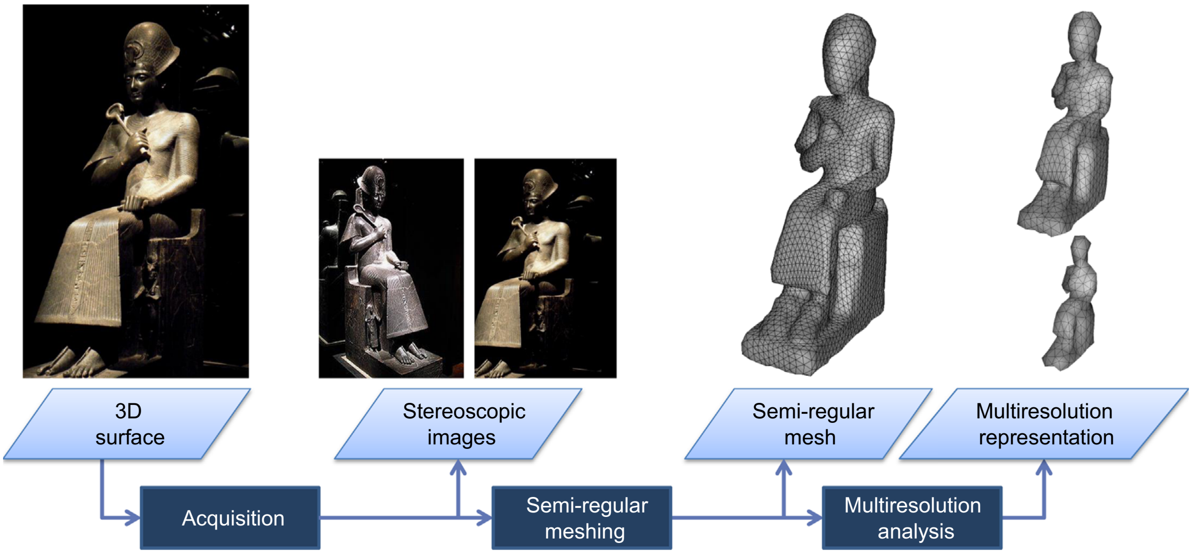


Fig. 2. Our 3D reconstruction technique that produces a semi-regular mesh directly from stereoscopic images.

use of the stereoscopic images as parameterization domain to create the semi-regular mesh.

The remaining of the paper is organized as follows. In Section 2, we remind the reader of the basics of semi-regular meshes and briefly review two prior methods of surface reconstruction based on stereoscopy and parameterization. Section 3 presents our semi-regular reconstruction method. Experimental results are presented in Section 4. Finally, Section 5 summarizes our contributions, and proposes future work.

2. Background

2.1. Semi-regular meshes

A semi-regular mesh M_{sr} is a structured mesh defined by L levels of resolutions (Fig. 3), where all the triangles at a specific level can be merged by fours down to a lower resolution mesh.

This merging process can be applied $(L-1)$ times to M_{sr} until obtaining a base mesh M_0 that represents the lowest resolution of M_{sr} (M_{sr} can be seen as M_{L-1}). A semi-regular

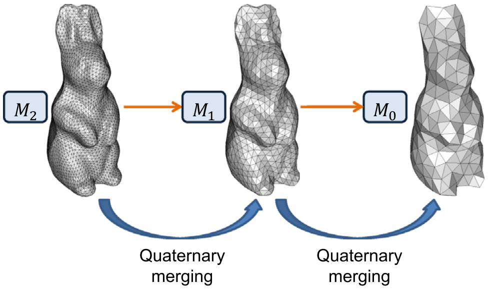


Fig. 3. Semi-regular mesh of the model Rabbit ($L=3$ levels of resolutions).

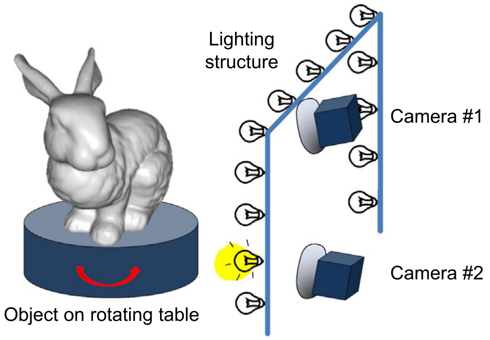


Fig. 4. Image acquisition system presented in Park et al. [12] (image of [12]).

mesh is sometimes called a subdivision mesh, because a subdivision scheme is applied on the mesh at resolution l to generate the semi-regular mesh at the finer level of resolution $(l+1)$.

2.2. Presentation of two prior surface reconstruction methods

We now present two prior reconstruction methods similar to our proposal, because they are based on multi-view images and use a parameterization. Interested reader will find a complete presentation of general reconstruction methods in Seitz et al. [22].

The method proposed in Park et al. [12] combines the advantages of geometric and photometric techniques, thanks to the surface parameterization. It consists in associating a *Multi-View Stereo* (*MVS*) reconstruction process that relies on a correspondence between pixels from different multi-view images, and a *Shape from Shading* method that utilizes the surface reflectance. The authors use two cameras, an array of lights and a rotation table on which the object is put (see Fig. 4). The rotation table allows to acquire several images of the object at different points of view, whereas only one light at a time is turned on to provide different lighting configurations.

First of all, the technique *Structure from Motion* [23] generates the 3D point cloud of the scanned object. The multi-view method *MVS* of Hernandez et al. [8] is then used to generate a depth-map, and the base mesh. The third step consists in creating a parameterization by *charts* [25] (see Fig. 5). Finally, from the parameterization and the normals estimated at each vertex of the base mesh, a refinement procedure is applied, leading to high-quality reconstructions.

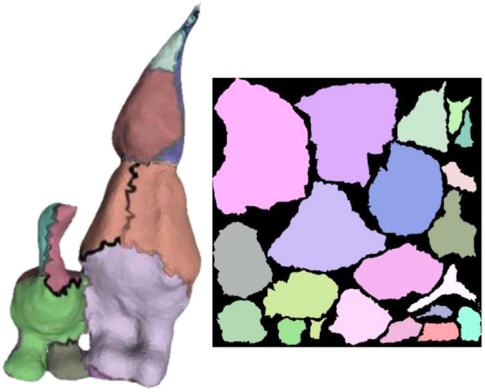


Fig. 5. Charts defined on the base mesh, and its associated parameterization (image of [12]).

Another relevant approach is proposed in Pietroni et al. [19]. The authors present a quadrangular remeshing technique based on a global and low distortion parameterization of different kinds of surfaces (polygonal meshes, point clouds, etc.). The principle, illustrated in Fig. 6, is to first generate a set of distance maps U^i of the input data. Then, each image U^i is parameterized into a 2D planar domain, while controlling the resulting distortion at the frontiers of the images in the final parameterization. Finally, a sampling in the parameterization domain creates a quadrangular semi-regular mesh.

Discussion: These two parameterization-based methods are reliable. However, we cannot refer to Park et al. [12] to get a semi-regular mesh directly from stereoscopic images, as a coarse 3D mesh must be built before creating the parameterization. The other method, [19], is closely related to our goal, it requires a cross-field technique that might be complex and uses triangles embedded in $\mathbb{R}^3$. *A contrario*, our method strives to minimize the use of the 3D connectivity by using the stereoscopic images as parameterization domain and a *coarse-to-fine* approach.

3. Presentation of our semi-regular reconstruction method

3.1. Overview

To highlight the interest of our approach, we first present the classical pipeline to get a semi-regular mesh of a physical object with a stereoscopic system (Fig. 7(a)).

1. *Stereo matching*: The goal is to find the *Pixels Of Interest* (*POI*) *region* in the two images that represents the physical object [21]. The POI region gathers the couples of pixels that correspond to a same point in the 3D space through both cameras (yellow parts on the left and right stereoscopic images). The POI region is only a subset of the stereoscopic images since it is impossible

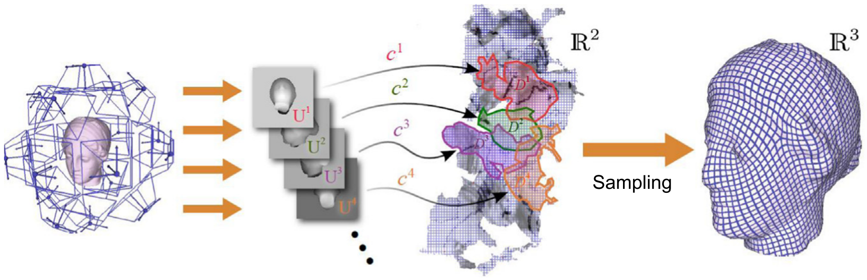
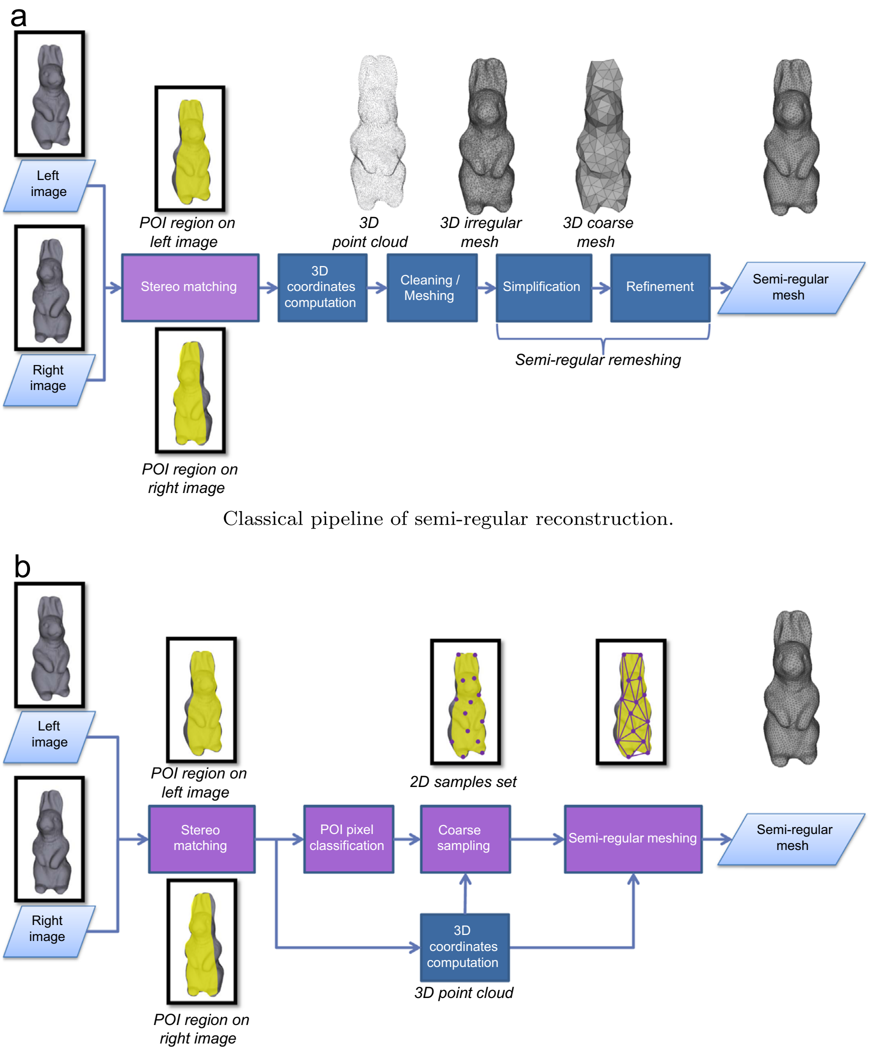


Fig. 6. Overview of Pietroni et al. [19]'s method (image of [19]).



Classical pipeline of semi-regular reconstruction.

Proposed semi-regular reconstruction method.

Fig. 7. How to get a semi-regular mesh from stereoscopic images? The classical pipeline (top) *vs* our direct *coarse-to-fine* reconstruction method (down). Purple and blue blocks indicate that the process is realized in 2D and 3D space, respectively.

to capture the same set of 3D points from two different points of views.

2. *3D coordinates computation*: The coordinates of the 3D points are computed for all the pixels belonging to the POI region [7]. These two first steps are done by the acquisition system.
3. *Cleaning/meshing*: The 3D point cloud must be cleaned, and then triangulated, leading to a dense irregular mesh. This is the second independent process.
4. *Simplification*: The semi-regular remeshing can now be done (third independent process): the irregular mesh is first simplified to obtain a coarse mesh corresponding to

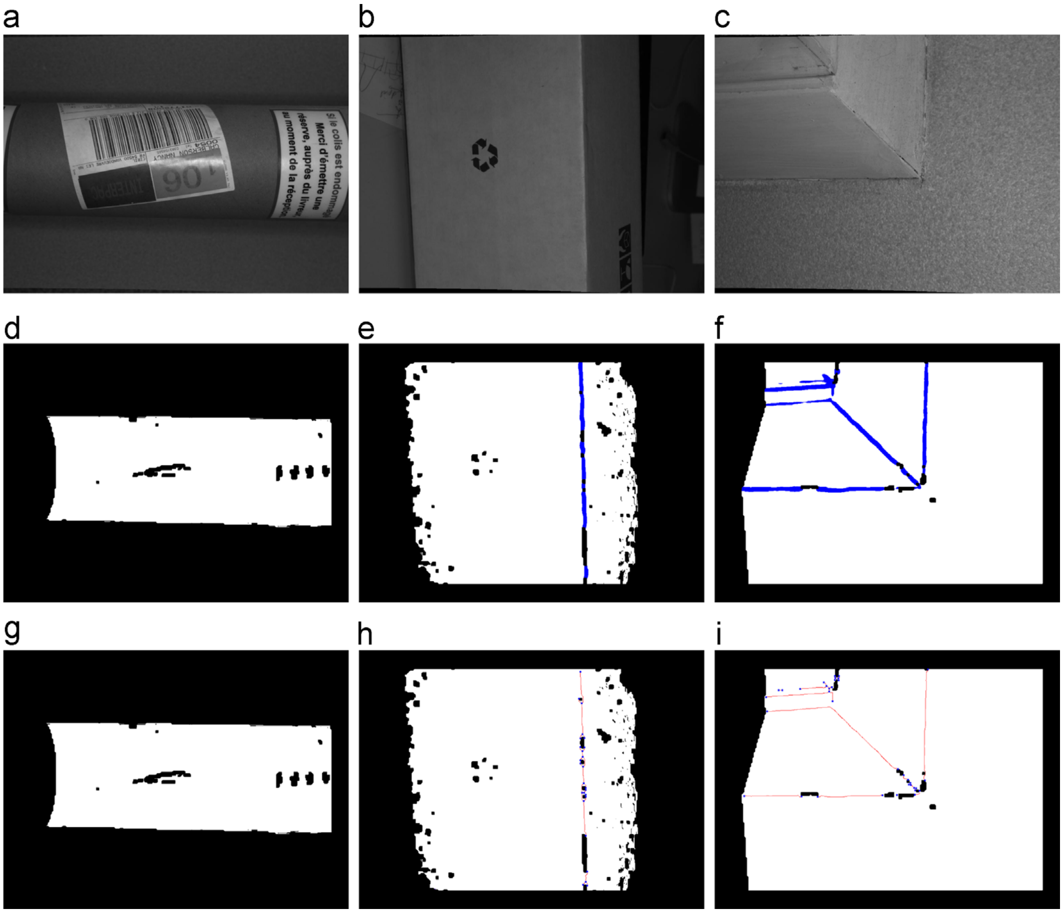


Fig. 8. Detection of the feature lines *via* our classification technique. First row: left stereoscopic images obtained with our scanner (models Pipe, Box and Wall). Second row: detection of high curvature areas (in blue). Third row: resulting classification after thinning: white, red and blue pixels represent respectively the *smooth regions*, the *sharp features* and the *corners*.

the base mesh of final semi-regular mesh. During this stage, a parameterization of the irregular mesh vertices is generally computed onto this base mesh.

5. *Refinement*: The base mesh is subdivided several times (1:4 subdivision) to create the different resolutions of the final semi-regular mesh. Generally, the aforementioned parameterization optimizes the positioning of the new vertices added by subdivision.

The originality of our semi-regular reconstruction method, illustrated in Fig. 7(b), is that it mainly works onto the 2D domain defined by the stereoscopic images, and thus can be included in the acquisition system:

1. *Stereo matching*: This stage is identical to the one in the classical approach.
2. *POI pixel classification*: The goal is to detect the feature lines in the POI region. The creation of the base mesh will be guided by these feature lines to ensure that the geometrical features are preserved on the final semi-regular output. Moreover, such assertion greatly improves the reconstruction quality.
3. *Coarse sampling*: A coarse sampling, constrained by the feature lines, is done to retrieve a set of 2D samples that will be later the vertices of the base mesh. This stage is based on 2D Poisson-disk sampling, to ensure a good distribution of the samples over the POI region.
4. *Semi-regular meshing*: The set of 2D samples is first triangulated to obtain a 2D base mesh. Then, this 2D base mesh is subdivided several times, to get a 2D semi-regular mesh of the POI region. Finally, the 3D semi-regular mesh is obtained by computing the 3D coordinates associated to the 2D samples.

3.2. POI pixel classification

To detect the feature lines in the POI region, we first classify the pixels according to their curvature values,[1] as described below.

[1] In the current version, the curvature values are calculated with the technique of Park et al. [13] from the 3D normals associated to the corresponding 3D point cloud. *In fine*, to limit the use of 3D information, this technique will be replaced in our algorithm by the recent technique of Daval et al. [3] that computes the 3D normals directly from stereoscopic images.

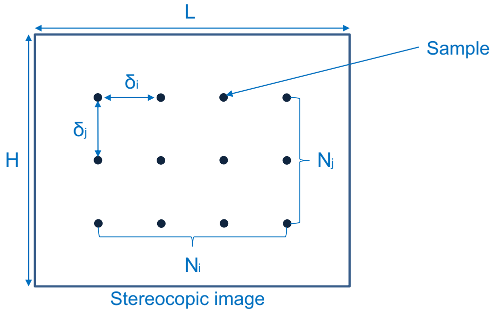


Fig. 9. Example of uniform sampling performed on one stereoscopic image.

A tensor $T_{p(u,v)}$ is calculated at each pixel $p(u,v)$ in the POI region using

$$T_{p(u,v)} = \sum_{u'=u-n}^{u'=u+n} \sum_{v'=v-n}^{v'=v+n} \overrightarrow{N'} \cdot \overrightarrow{N'}^{t}, \tag{1}$$

where $\overrightarrow{N'}$ is the 3D normal associated to the neighbor pixel $p'(u',v')$, and n depends on the size of the considered neighbor region of p.

The three eigenvalues of $T_{p(u,v)}$ are then computed with the Jacobi operator, and a thresholding operation performs the segmentation of high curvature area. In order to reduce the runtime and benefit the independence of the operation at each pixel p, this classification is GPU-parallelized.

However, this classification is not precise enough to be exploited as it is. A parallelized thinning technique [24] is thus applied to the 'high curvature' pixels to finely detect the sharp edges. Thinning a set of neighbor pixels consists in generating a skeleton (*i.e.* a set of median lines) that presents the same topology as the related shape.

Once the thinning is done, we classify the pixels in the POI region according to three classes:

- *corners* containing the pixels where the median lines intersect in the image;
- *sharp features* containing the remaining skeleton pixels;
- *smooth regions* containing the other pixels.

Fig. 8 depicts several results of classification obtained with this method.

This classification of POI pixels will help the subsequent sampling to preserve geometrical features and thus to provide a consistent base mesh in terms of global shape and geometrical characteristics, as explained below.

3.3. Coarse sampling

This stage is inspired by the feature-preserving Poisson-disk sampling for surfaces of Peyrot et al. [18], which is based on a dart throwing.

This approach can be efficiently adapted to our setting: the sampling domain Ω becomes the POI region of the stereoscopic images (instead of a surface mesh in [18]), and the feature lines detected by the previous stage guide the distribution of 2D samples.

However, as the output 2D samples of this stage will define the vertices of the base mesh, they must be consistently distributed *over the surface* of the object (and not especially *over its stereoscopic images*). Therefore the sampling is done onto the POI region, to benefit from its implicit 2D connectivity, but the distances between samples are computed in the 3D space with Dijkstra's algorithm [4].

The principle of the dart throwing on a 2D image is the following: (i) one pixel in the 2D domain Ω is chosen randomly, (ii) a disk is computed around it, according to a radius R that depends on the target number N of samples and a density function, (iii) this pixel is considered as a valid sample if the disk does not intersect the disks relative to the samples already accepted (ensuring a minimal distance between the samples).

One key idea of the proposed sampling technique is the computation of the radius R onto the surface in the 3D space, while handling the POI region of the stereoscopic images. Given the requested number of samples N, we first calculate the horizontal δ_i and vertical δ_j deviations between samples when a uniform sampling pattern is realized on the stereoscopic image. It generates a grid of samples of dimension $\mathbb{N}_{\delta_i} \times \mathbb{N}_{\delta_j}$, as depicted in Fig. 9, where N_i and N_j represent the number of samples per row and per column, respectively ($N_i \times N_j = N$).

To take care of the fact that the sampling domain Ω is restricted to the pixels in the POI region, the distances δ_i and δ_j between samples along each dimension are shrunk by a factor $L \times H / Card\{POI\}$, with $Card\{POI\}$ being the number of pixels in Ω. A uniform sampling can be realized using

$$R = \tfrac{1}{3} \cdot \max(\delta_i, \delta_j) \cdot S_r, \tag{2}$$

where S_r is the spatial resolution of the scanner (0.3 mm in our case). With some objects, it can be convenient to realize an adaptive sampling, to better preserve the geometrical features for instance. In that case, the radius will depend on the surface curvature according to the following equation [18]:

$$R = \tfrac{1}{3} \cdot \max(\delta_i, \delta_j) \cdot S_r \cdot \left(1 + e^{C \cdot \lambda_2} + e^{C \cdot \lambda_3}\right). \tag{3}$$

Empirically, we put $C = -8.0$ for the pixels of the class *sharp features*, and $C = -6.0$ for the pixels of the class *smooth regions*. λ_2 and λ_3 are the eigenvalues of the tensor $T_{p(u,v)}$ computed in Section 3.2. In this formulation, the *corners* pixels keep the minimum radius given by Eq. (2). To determine the disks associated to the samples in function of the radius R, we recall that we use Dijkstra's algorithm to compute geodesic distances between 3D points, while using the connectivity of the 2D sampling domain Ω. Therefore, a disk does not depend on the Euclidean distance between two given 2D samples, but on the sum of the lengths of the 3D segments defined by the shortest path in the POI region, as shown in Fig. 10. As output of this stage, we get a set of 2D samples that ensures a good distribution of the vertices of the 3D base mesh all over the scanned surface.

3.4. Semi-regular meshing

We now present how to generate a semi-regular mesh directly from the set of 2D samples defined previously. It is a three-stage process: creation of the 2D base mesh from

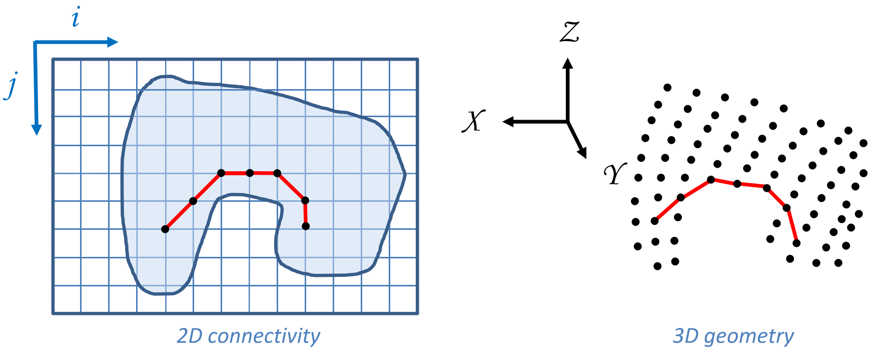


Fig. 10. Computation of a geodesic distance between two points of the surface in $\mathbb{R}^3$ (right image), driven by the shortest path between the associated pixels in the POI region (light blue region, left image).

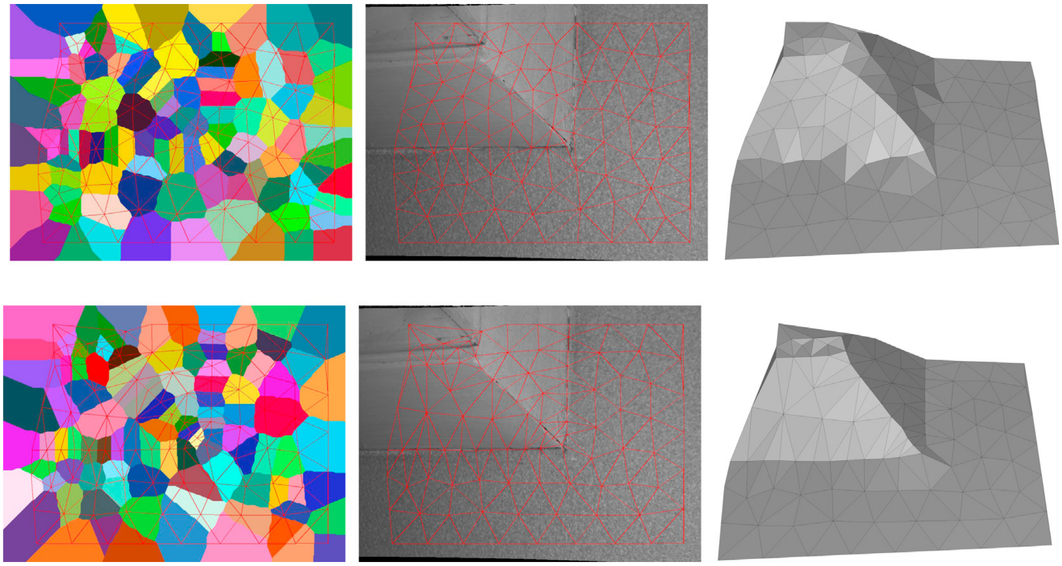


Fig. 11. Base mesh generated by our Voronoi relaxation without (first row) or with (second row) the constraint on the feature lines. Left: final Voronoi diagram and triangulation. Middle: the same triangulation on the left stereoscopic image. Right: the resulting 3D base mesh.

the set of 2D samples, refinement by iterative subdivisions to get a 2D semi-regular mesh, and fitting in 3D space.

Creation of the base mesh: The base mesh is obtained *via* a constrained Voronoi relaxation [9] of the samples in the stereoscopic image domain, followed by the triangulation of the relaxed samples (via the dual of the Voronoi diagram [20]). In our context, the Voronoi relaxation consists in first computing a Voronoi diagram of the pixels in function of the set of samples, and then displacing each sample to the centroid of its cell. This process is iteratively repeated until convergence. The relaxation greatly improves the mesh quality, when comparing with the triangulation that we could obtain directly from the initial voronoi diagram.

In this work, the Voronoi diagrams are generated with Munshi et al. [11] that is a GPU implementation of popular Dijkstra's algorithm [4]. We had to adapt this algorithm to process stereoscopic images. Moreover, to preserve the feature lines on the created base mesh, we added a constraint during the relaxation: the new samples must belong to the same class than the initial samples (*corners*, *sharp features* or *smooth regions*). In other words, if after relaxation a sample is moved to a pixel which does not belong to the same class, then the sample is displaced to the closest pixel of the same class. This technique is straightforward, but produces nice triangulations, while preserving geometrical features of the scanned object, as shown in Fig. 11. This figure shows also the poor triangulation obtained if the constraint is not included.

Refinement: A 2D semi-regular mesh is first obtained by applying several midpoint subdivisions [1] to the base mesh of the left stereoscopic image (see Fig. 12). Then, the surface fitting will embed the semi-regular mesh in the 3D space.

During the subdivision, some new vertices might be either outside the POI region, or in holes (areas without 3D correspondences). In the first case, they are displaced to their closest POI pixel, as shown in Fig. 12(c) and (d). To avoid an exhaustive research over the POI region, we use a parallelized k-Nearest Neighbors algorithm (with $k=1$).

In the second case, if we use the same technique, the resulting triangles will be badly shaped (see Fig. 13(a)), which globally decreases the mesh quality. To reduce such

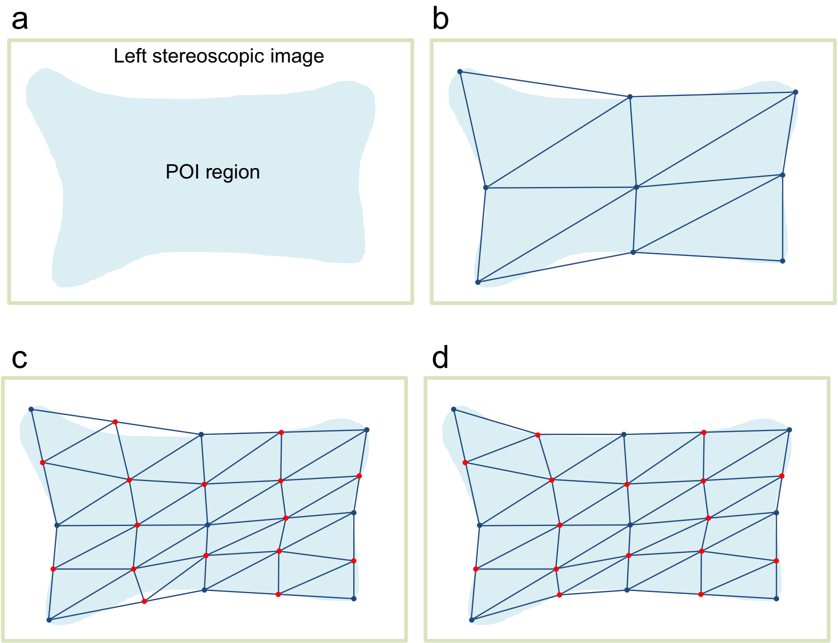


Fig. 12. Generation of the 2D semi-regular mesh. (a) Left stereoscopic image and its POI region. (b) 2D base mesh. (c) Subdivision. (d) Displacement of the new vertices (red ones).

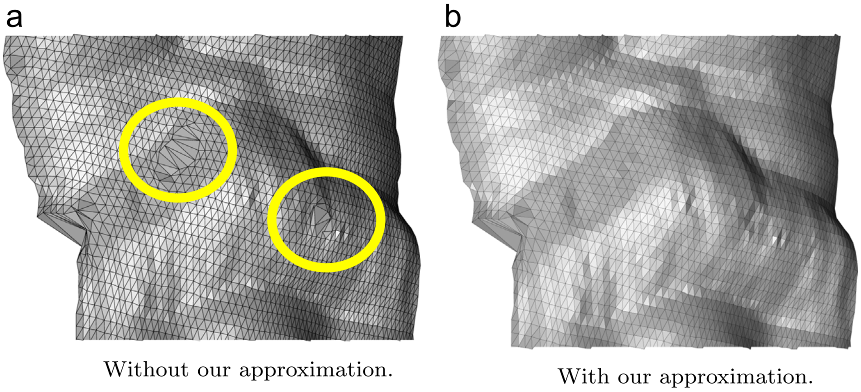


Fig. 13. Technique proposed to fill the holes during the refinement. (a) Without our approximation. (b) With our approximation.

artifacts, we choose to keep the vertices “fallen in a hole” in the 2D domain, and so in the 3D space. Fig. 13(b) shows that, with this technique, the triangles filling the holes are better shaped. Note that instead of using this simple scheme, one could use an interpolating scheme such as *Butterfly* [5].

4. Experimental results

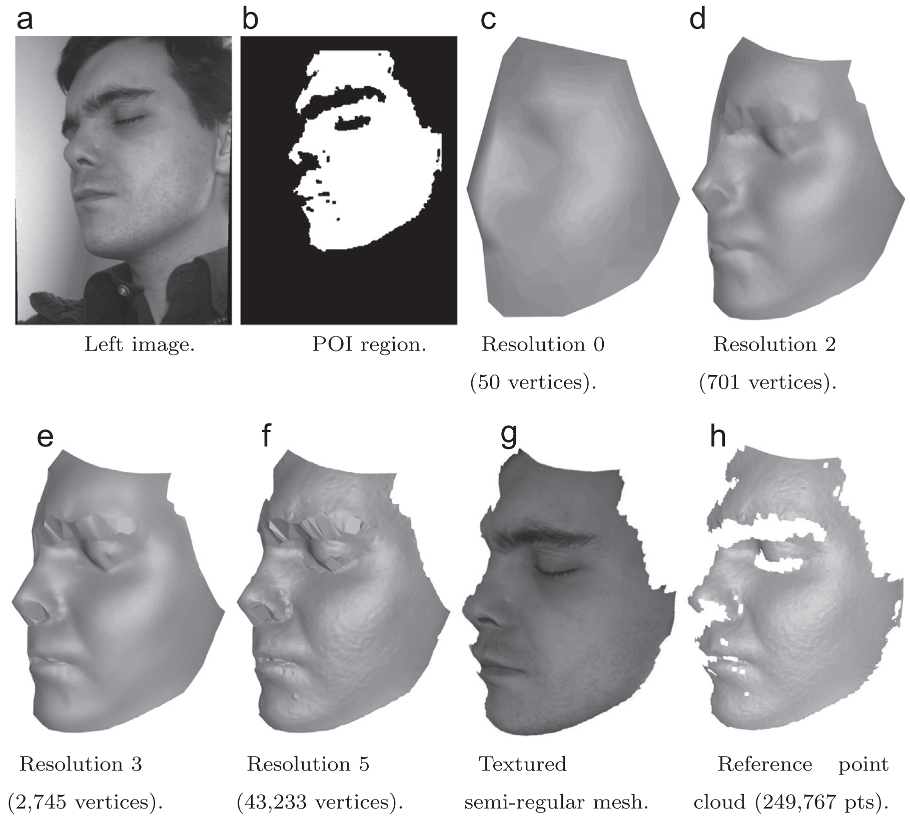
4.1. Visual results

All our results are generated with a single pair of stereoscopic images obtained with a hand-held scanning system. Fig. 14 gives an overview of our method on the model Face.

From the stereoscopic images (a), the POI region (b) is defined, and the base mesh (resolution 0) is created (c). Then, our coarse-to-fine approach generates several resolutions (d,e,f). At resolution 5, our semi-regular reconstruction, with only 43k vertices, is already a good approximation of the original cloud of 250k points given by the stereoscopic system. This is promising in terms of both compactness and compression. Fig. 14(g) also shows the textured semi-regular mesh, produced in a very simple way, with one stereoscopic image. No additional texturing technique is necessary as the connectivity of the semi-regular mesh is generated directly on the image domain. This is another great advantage of our approach.

4.2. Uniform vs adaptive sampling

We now study the efficiency of our feature-preserving technique, and the difference in terms of triangle quality, between the meshes produced with the uniform/adaptive



Left image.

POI region.

Resolution 0 (50 vertices).

Resolution 2 (701 vertices).

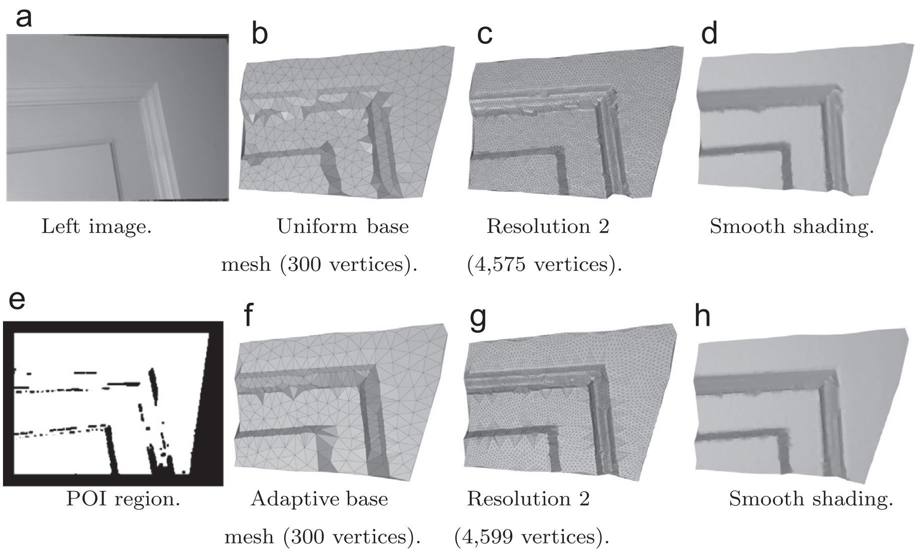
Resolution 3 (2,745 vertices).

Resolution 5 (43,233 vertices).

Textured semi-regular mesh.

Reference point cloud (249,767 pts).

Fig. 14. Semi-regular reconstruction of the model Face.



Left image.

Uniform base mesh (300 vertices).

Resolution 2 (4,575 vertices).

Smooth shading.

POI region.

Adaptive base mesh (300 vertices).

Resolution 2 (4,599 vertices).

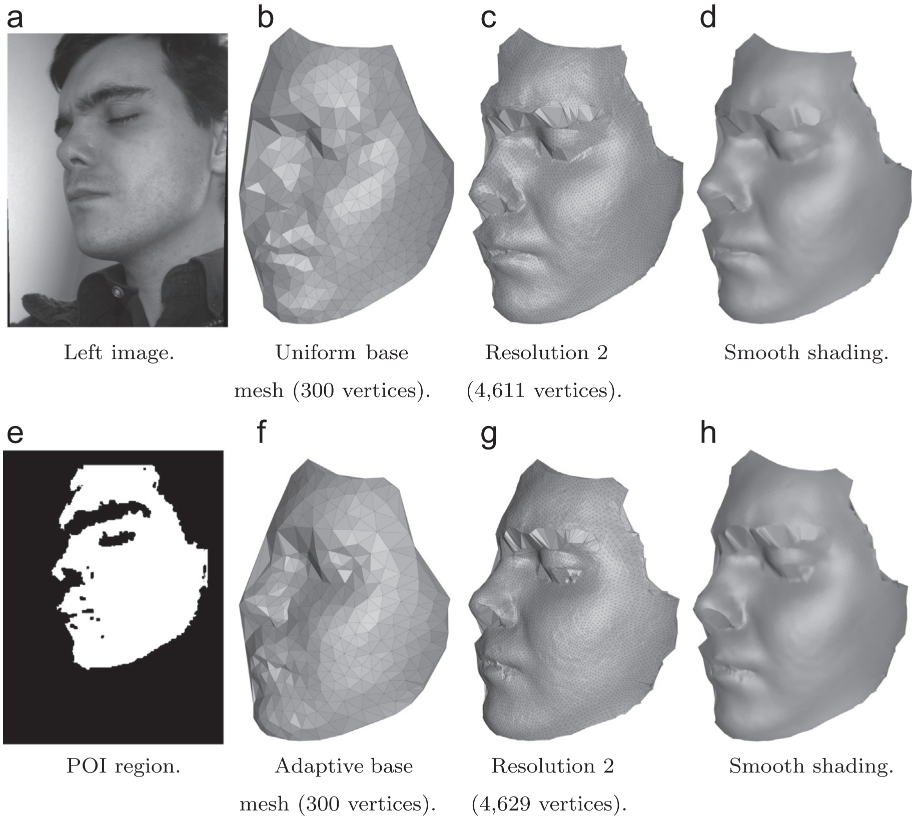
Smooth shading.

Fig. 15. Preservation of the geometrical features of Door, with the uniform or the adaptive sampling.

samplings during the creation of the base mesh (Section 3.3).

Fig. 15 shows a reconstruction of a surface having sharp features called Door.

Fig. 15 (b)–(d) present the results with the uniform sampling, whereas Fig. 15(f)–(h) present the results with the adaptive sampling. We observe on the smooth shadings that the features are globally well preserved whatever the sampling. Some artifacts along them are visible, but they are due to the holes in the POI region that generate notches along features when the base mesh is created (these artifacts would be removed by improving the stereo matching in the scanning system).



Left image. Uniform base mesh (300 vertices). Resolution 2 (4,611 vertices). Smooth shading.

POI region. Adaptive base mesh (300 vertices). Resolution 2 (4,629 vertices). Smooth shading.

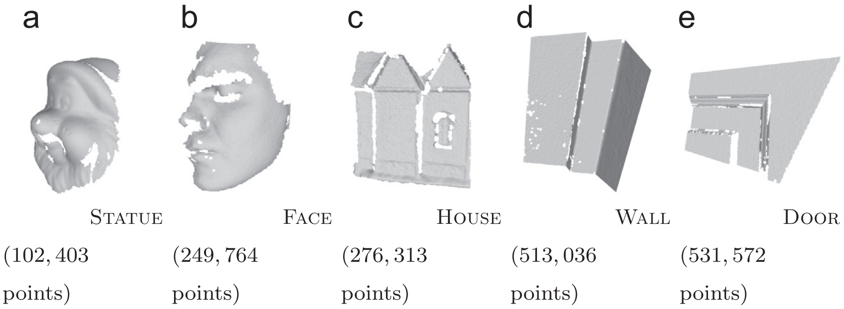
Fig. 16. Difference of sampling quality obtained on FACE in function of uniform/adaptive sampling.

However, we observe in Fig. 15(b) and (f) that the adaptive sampling tends to better preserve the features from the lowest resolution. This result was expected, as the adaptive approach takes into account the curvature during the computation of the disks, leading to a dense sampling pattern along the geometrical features. The counterpart is that the sampling is globally less uniform, and the quality of the triangles is lower: the average minimum angle is 42.5° and 37.5° respectively for the uniform and the adaptive sampling.

Fig. 16 gives an additional result on FACE: we see that the uniform sampling tends to provide a more isotropic mesh, and that the edges of the base mesh are less visible at high resolutions, which is advocated in case of smooth surfaces. On the database of five objects shown in Fig. 17, the uniform sampling increases of 11% the average min angles.

4.3. Runtime

We now evaluate the runtime of our semi-regular meshing on five surfaces shown in Fig. 17. The results have been obtained with an Intel Core i3 CPU 2.30 GHz processor, associated to a 4 GB RAM.



STATUE (102, 403 points) FACE (249, 764 points) HOUSE (276, 313 points) WALL (513, 036 points) DOOR (531, 572 points)

Fig. 17. Database used to compute the runtimes of Fig. 19.

Fig. 18 shows the runtime in function of the resolution, when the base mesh has 50 vertices, and eight resolutions. The bottom part of this figure is a zoom of the graphic,

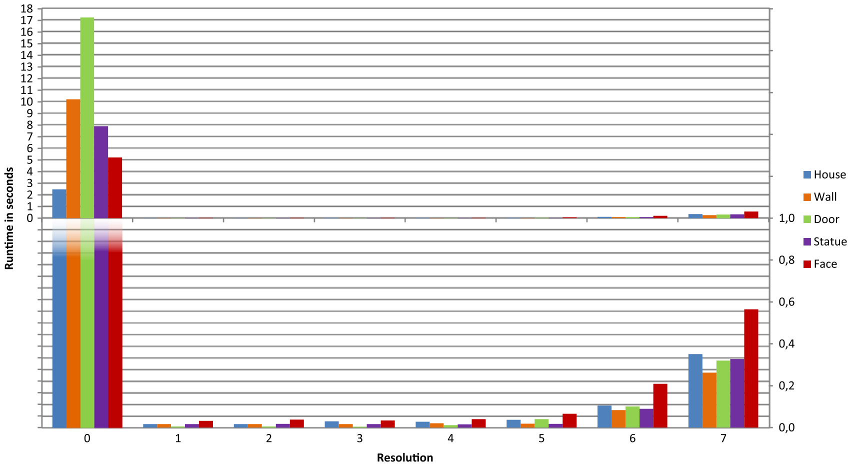


Fig. 18. Runtime in seconds of our method, per resolution. The bottom part is a zoom of the top one where the *Y*-axis spans from 0 to 1.

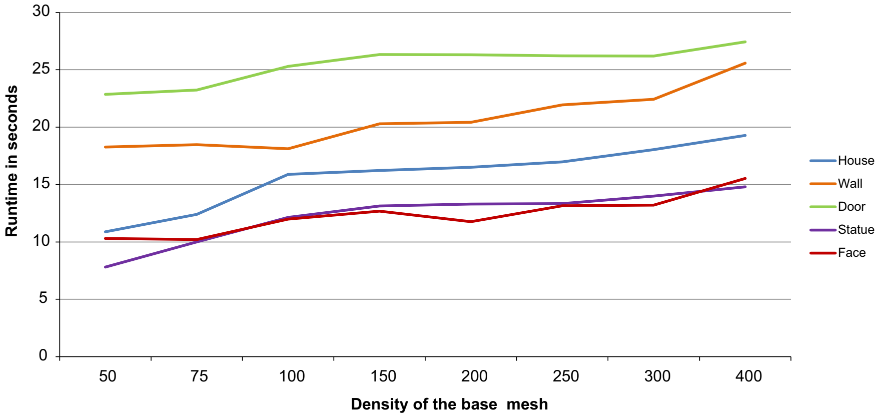


Fig. 19. Total runtime (in seconds) of our method in function of the number of vertices of the base mesh.

where the *Y*-axis spans from 0 to 1. Note that the preprocessing runtime that consists in detecting the POI region and the POI classification is not included.

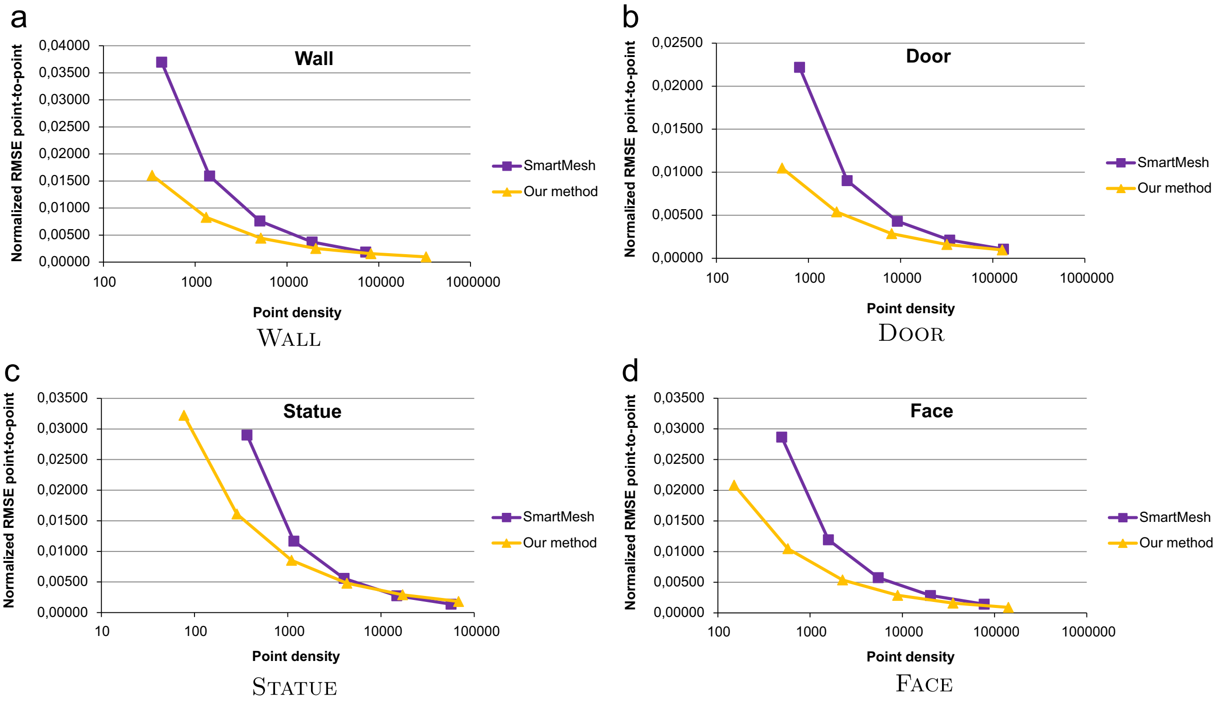
As expected, the most "greedy" resolution is the first one, during which the base mesh is generated (including the Poisson-disk sampling, the constrained 2D Voronoi relaxation, and the 2D Delaunay triangulation). For instance, the runtime to generate the 8 resolutions is around 17 s for the Door model, among which 90% is dedicated only to build the base mesh. Indeed, the Poisson-disk and the relaxation phases are not parallelized and time-demanding. The obtaining of the other resolutions is much faster, partly because these steps have been parallelized on GPU. As a proof, the generation of the seventh resolution defined by 705,281 vertices (528,768 vertices are added) requires less than 0.6 s in the worst case. Nevertheless, this remains very fast.

Now, Fig. 19 shows the total runtime (including the aforementioned preprocessing runtime) to generate 8 resolutions, in function of the base mesh density. For each surface, the curve is obtained by averaging the runtime of five tests. Indeed, our algorithm is not deterministic (because of the dart throwing), and the relaxation time depends on the initial sampling. So, we can obtain slightly differences at each resolution for a same surface.

We observe the linearity of the total runtime with respect to the number of vertices of the base mesh. The differences between the models is due to the original point cloud density (ranging from 102,403 points for Statue to 531,572 points for Door). The curvature of the scanned surface also influences the runtime. For instance, Wall contains around 18*k* points less than Door, but it contains much more pixels classified as *sharp features*. Finally, the mean sampling runtime is slower: 5.70 s for Wall, while it amounts to 10.90 s for Door.

4.4. Comparison with the classical pipeline

We now compare our direct semi-regular meshing with the classical acquisition pipeline to get a semi-regular mesh (*point cloud generation* → *triangulation* → *semi-regular remeshing*). We use a Voronoi-based triangulation technique to generate the original irregular reference mesh M_{ori} from the point cloud provided by our stereoscopic system. This irregular mesh is then remeshed semi-



WALL

DOOR

STATUE

FACE

Fig. 20. Comparison of the geometry sampling obtained with our method and with *SmartMesh* [2], depending on the vertex density of the semi-regular meshes.

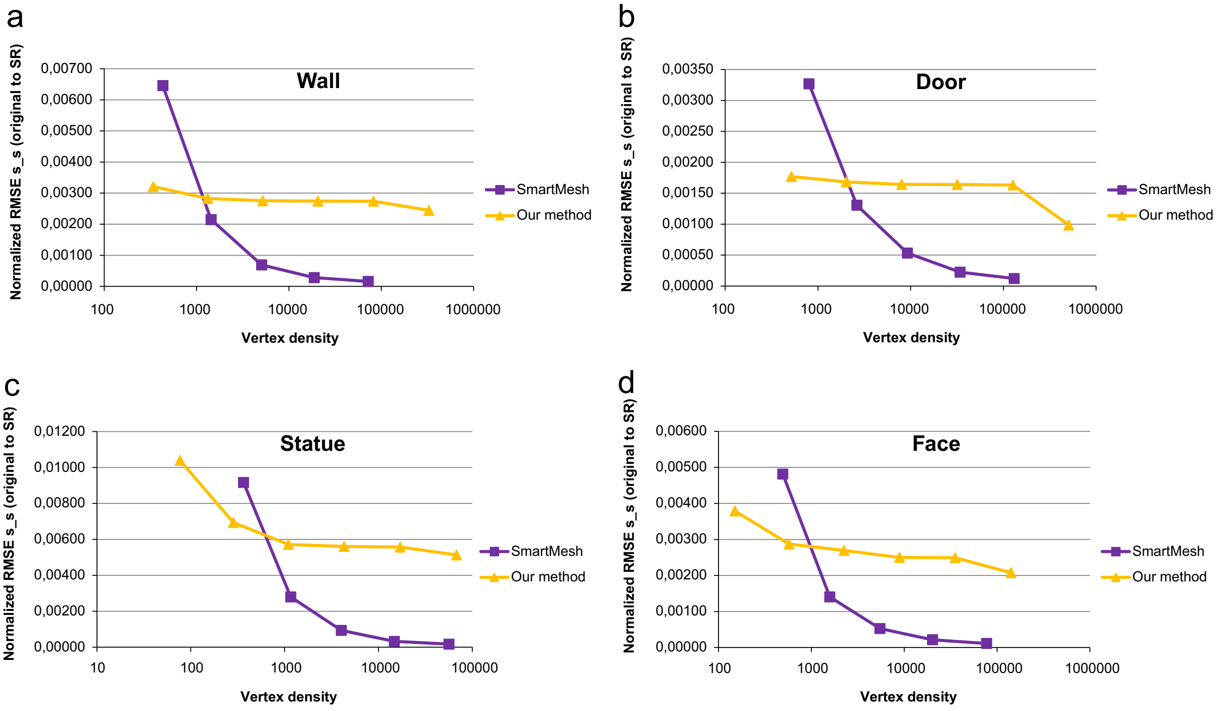


Fig. 21. Asymmetric RMS distance $RMS(M_{ori} \rightarrow M_{sr})$ obtained with our method and with *SmartMesh* [2] in function of the resolution.

regularly either with the SDK *SmartMesh* developed by the company [2], or with *Trireme* [6]. To our knowledge, they are the only semi-regular remeshing techniques available on the web. Unfortunately, we found out that *Trireme* is not a suitable tool to remesh our data. Indeed, we could not produce any semi-regular meshes without severe degeneracies and outlier triangles. In the contrary, *SmartMesh* always provides manifold semi-regular meshes, in

Table 1
Runtime comparison in seconds between our method and *SmartMesh* [2], to generate 5 resolutions.

Models	Wall	Door	Statue	Face
Base mesh density	431	800	365	492
SmartMesh	≥ 3 min	≥ 7 min	≥ 2 min	≥ 6 min
Our method	19.1 s	27.7 s	10.1 s	10.3 s

particular because it does not use any parameterization, unlike *Trireme*. Consequently, we only compare the reconstruction errors and runtimes relative to our semi-regular meshing and to *SmartMesh*.

We first compute the symmetric root mean square distance between the set of vertices of our semi-regular meshes, and the set of vertices of M_{ori} (the original point cloud provided by the acquisition system). It allows us to assess the fidelity of our sampling to the reference point cloud. The same distance is calculated with the set of vertices of the semi-regular meshes produced by *SmartMesh*. Fig. 20 shows the evolutions of these distances depending on the resolutions: the X-axis indicates the associated number of points. We observe that our method presents lower distances than *SmartMesh*. It was expected as our method is approximating, contrary to *SmartMesh* that is interpolating (it optimizes the positions of the vertices such as its semi-regular mesh is close to the reference mesh). Our method has the advantage to determine the majority of vertices among the original point cloud, as the vertices are selected *via* the pixels of the POI region in the image domain. The only vertices that do not exist in the original point cloud are associated to pixels selected outside the POI region during subdivision. Thus, our method is more accurate when considering only the geometry of the initial surface.

We now assess the fidelity of our semi-regular meshes with respect to the reference mesh M_{ori}. To achieve this goal, we compute the symmetric Root Mean Square (RMS) distance between M_{ori} and our semi-regular meshes M_{sr} (normalized by the diagonal length of the bounding box), which is widespread used in the state-of-the-art [15]. However, in our context, this measure is not suited. Indeed, as explained in Section 3.4, our method fills the holes, in order to make the texturing easier and to enhance the mesh quality. As a consequence, when measuring the symmetric RMS distances between our semi-regular meshes and the reference irregular mesh M_{ori}, the distances between the triangles filling the holes and the original surface are inevitably high. It severely corrupts the comparison with *SmartMesh*, as this latter has been initially developed to preserve the potential borders of a surface and consequently the holes. So, to fairly make comparisons, we compute the asymmetric RMS distance $RMS(M_{ori} \rightarrow M_{sr})$, which excludes the filled holes: see Fig. 21 (the X-axis still indicates the number of points of each resolution). Globally, we observe that our method is better than *SmartMesh* in the first resolutions. It can be explained by the fact that our method tends to preserve the geometrical features in the base mesh, and that it is approximating [15]. On the other hand, *SmartMesh* becomes better on the highest resolutions, because it minimizes the geometric distortion directly onto the original surface, without any parameterization, which avoids the relative distortion. Furthermore, our method is penalized by the fact that it works in the image domain, but also by our feature preservation that positions more vertices on them.

On the other hand, our algorithm is direct and thus significantly faster than *SmartMesh*. The runtime comparison is summarized in Table 1. *SmartMesh* indeed takes several minutes to produce the semi-regular meshes: from 2 to 7 min in function of the data, excluding the triangulation time, whereas our method needs always less than one minute. Finally, it shows that the proposed pipeline is a promising alternative to the classical one.

5. Conclusion and perspectives

In this paper we proposed an alternative to the fastidious pipeline to get semi-regular meshes from physical objects. The idea is to generate semi-regular meshes directly from the stereoscopic images acquired with a hand-held stereo acquisition system. The key idea of our work is that the stereoscopic images can be considered as a parameterization of the acquired surface. Therefore, our reconstruction method processes the data as much as possible into the image domain, before embedding the surface in the 3D space.

The first contribution is an original sampling that creates a base mesh of the scanned surface. We show that the Poisson-disk sampling developed by Peyrot et al. [17] can be extended to a stereoscopic system, while retrieving the 3D information necessary to preserve features all along the process. This allows us to take into account the surface geometry, although the sampling is realized on the stereoscopic images. The second contribution concerns our *coarse-to-fine* approach that allows us to get a semi-regular mesh preserving the geometrical features as output of our acquisition system, by working mainly in the image domain.

Our pipeline could be easily included into any stereoscopic acquisition system. It also has the advantage to create semi-regular output that can be directly textured with the stereoscopic image, and is also much more faster and convenient than the classical pipeline.

However, a lot of improvements remains possible. For instance, the runtime of our algorithm can be improved as some parts are implemented on CPU. It would be interesting to investigate parallel algorithms for all the stages, to allow quasi real-time reconstructions. We could also investigate new means to improve the shape fidelity, in order to be competitive with the semi-regular remeshing techniques. Another promising improvement would be to manage several views. Indeed, our current algorithm handles only one view, and thus only a part of the scanned object can be reconstructed. It would be relevant to study, for instance, *mosaicing* techniques, widespread in photogrammetry, to generate a large POI region representing the whole parameterized object, and thus to output a complete semi-regular representation of a physical object.

Acknowledgments

This work was supported by a grant from *Région Provence Alpes Côte d'Azur* (11BDE004ACSR) in France.

References

[1] Qi Chen, Hartmut Prautzsch, General midpoint subdivision. CoRR, abs/1208.3794, 2012. URL ⟨http://arxiv.org/abs/1208.3794⟩.

[2] Cintoo3D. Sdk smartmesh. ⟨http://www.cintoo3D.com⟩.

[3] Vincent Daval, Frederic Truchetet, Olivier Aubreton, Primitives extraction based on structured-light images, in: QCAV'13, Fukuoka, Japan, June 2013, p. 45. URL ⟨http://hal.archives-ouvertes.fr/hal-00861169⟩.

[4] Edsger W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.

[5] R. Egli, J.-P. Dussault, (Eds.), Technique butterfly généralisée, Dijon, France, Actes de Compression et Représentation des Signaux Audiovisuels, 2001, pp. 133–136.

[6] I. Guskov, Manifold-based approach to semi-regular remeshing, Graph. Models 69 (1) (2007) 1–18, http://dx.doi.org/10.1016/j.gmod.2006.05.001. ISSN 1524-0703.

[7] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second edition, . Cambridge University Press, 2004 ISBN: 0521540518.

[8] E.C. Hernandez, G. Vogiatzis, R. Cipolla., Multiview photometric stereo, IEEE Trans. Pattern Anal. Mach. Intell. 30 (March (3)) (2008) 548–554, http://dx.doi.org/10.1109/TPAMI.2007.70820. ISSN 0162-8828.

[9] Stuart P. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theory 28 (1982) 129–137.

[10] Michael Lounsbery, Tony D. DeRose, Joe Warren, Multiresolution analysis for surfaces of arbitrary topological type, ACM Trans. Graph. 16 (January (1)) (1997) 34–73, http://dx.doi.org/10.1145/237748.237750. ISSN 0730-0301.

[11] A. Munshi, B.-R. Gaster, T.-G. Mattson, J. Fung, D. Ginsburg, OpenCL Programming Guide, Prentice Hall, 2011.

[12] Jaesik Park, Sudipta N. Sinha, Yasuyuki Matsushita, Yu-Wing Tai, In So Kweon, Multiview photometric stereo using planar mesh parameterization, in: IEEE International Conference on Computer Vision, 2013, pp. 1161–1168. ISSN 1550-5499. http://dx.doi.ieeecomputersociety.org/10.1109/ICCV.2013.148.

[13] Min Ki Park, Seung Joo Lee, Kwan H. Lee, Multi-scale tensor voting for feature extraction from unstructured point clouds, Graph. Models 74 (4) (2012) 197–208.

[14] F. Payan, M. Antonini, Mean square error approximation for wavelet-based semiregular mesh compression, Trans. Vis. Comput. Graph. 12 (2006).

[15] F. Payan, C. Roudet, B. Sauvage, Semi-regular triangle remeshing: a comprehensive study, Comput. Graph. Forum 34 (February) (2015) 86–102, http://dx.doi.org/10.1111/cgf.12461.

[16] J.-L. Peyrot, F. Payan, N. Ruchaud, M. Antonini, Stereo reconstruction of semiregular meshes, and multiresolution analysis for automatic detection of dents on surfaces, in: Proceedings of IEEE International Conference in Image Processing (ICIP), Paris, France, October 2014.

[17] Jean-Luc Peyrot, Frédéric Payan, Marc Antonini, Feature-preserving Direct Blue Noise Sampling for Surface Meshes, in: Eurographics (Short Papers), 2013, pp. 9–12. http://dx.doi.org/10.2312/conf/EG2013/short/009-912 URL ⟨http://diglib.eg.org/EG/DL/conf/EG2013/short/009-912.pdf⟩.

[18] Jean-Luc Peyrot, Frédéric Payan, Marc Antonini, Direct blue noise resampling of meshes of arbitrary topology, Vis. Comput. (September) (2014). http://dx.doi.org/10.1007/s00371-014-1019-1.

[19] Nico Pietroni, Marco Tarini, Olga Sorkine, Denis Zorin, Global parametrization of range image sets, ACM Trans. Graph. 30 (December (6)) (2011) 149:1–149:10, http://dx.doi.org/10.1145/2070781.2024183. ISSN 0730-0301.

[20] Guodong Rong, Tiow-Seng Tan, Thanh-Tung Cao, Stephanus, Computing two-dimensional Delaunay triangulation using graphics hardware, in: Eric Haines, Morgan McGuire (Eds.), SI3D, ACM, Redwood city, 2008, pp. 89–97. ISBN 978-1-59593-983-8. URL ⟨http://dblp.uni-trier.de/db/conf/si3d/si3d2008.htmlRongTCS08⟩.

[21] Daniel Scharstein, Richard Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Int. J. Comput. Vis. 47 (April (13)) (2002) 7–42, http://dx.doi.org/10.1023/A:1014573219977. ISSN 0920-5691.

[22] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, Richard Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, CVPR '06, Washington, DC, USA, 2006, pp. 519–528. IEEE Computer Society. ISBN 0-7695-2597-0. http://dx.doi.org/10.1109/CVPR.2006.19.

[23] Noah Snavely, Steven M. Seitz, Richard Szeliski, Photo tourism: exploring photo collections in 3d, ACM Trans. Graph. 25 (July 3) (2006) 835–846, http://dx.doi.org/10.1145/1141911.1141964. ISSN 0730-0301.

[24] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (March (3)) (1984) 236–239, http://dx.doi.org/10.1145/357994.358023. ISSN 0001-0782.

[25] Kun Zhou, John Synder, Baining Guo, Heung-Yeung Shum, Isocharts: stretch-driven mesh parameterization using spectral analysis, Eurographics, July 2004. URL ⟨http://research.microsoft.com/apps/pubs/default.aspx?id=69060⟩.