



HAL
open science

From stereoscopic images to semi-regular meshes

Jean-Luc Peyrot, Frédéric Payan, Marc Antonini

► **To cite this version:**

Jean-Luc Peyrot, Frédéric Payan, Marc Antonini. From stereoscopic images to semi-regular meshes. *Signal Processing: Image Communication*, 2016, 40, pp.97-110. 10.1016/j.image.2015.11.004 . hal-01236999

HAL Id: hal-01236999

<https://hal.science/hal-01236999>

Submitted on 8 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From stereoscopic images to semi-regular meshes

Jean-Luc Peyrot, Frédéric Payan and Marc Antonini

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

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.

Keywords: Semi-regular mesh , 3D reconstruction, stereoscopy, acquisition, Poisson-disk sampling, GPU.

1. Introduction

2 Motivated by the high fidelity and the realism of the numerical models,
3 and supported by the increasing storage capacities, the acquisition devices pro-
4 vide now high resolution meshes, ensuring the preservation of the finest details.
5 Consequently these data are massive, and cannot be easily managed by any
6 workstation or mobile device with limited memory and bandwidth. The semi-
7 regular meshes are a good way to overcome these issues, because of their sca-
8 lability and their compactness. Indeed, the semi-regular meshes are based on

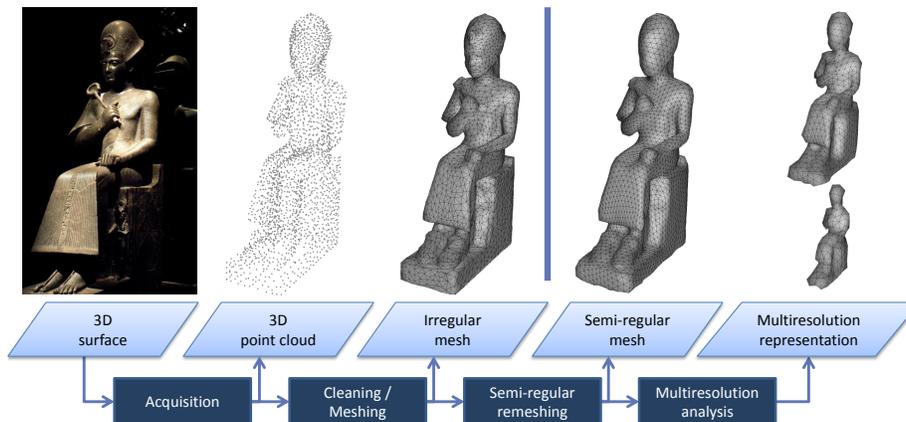


FIGURE 1: The pipeline to get a semi-regular mesh from a physical object, and its application to multiresolution analysis.

9 a regular subdivision connectivity, well-suited to display or transmit a mesh at
 10 different levels of details. This subdivision connectivity also allows a compact
 11 representation since only the connectivity of the lowest level of details is needed
 12 to reconstruct the full connectivity. This semi-regular structure is also adapted
 13 to multiresolution analysis (Lounsbery et al., 1997) and wavelet compression
 14 (Payan and Antonini, 2006). Despite their good properties, the semi-regular
 15 meshes are sometimes forsaken by users because they are not provided by cur-
 16 rent acquisition systems which only provide point clouds. So, if one wants
 17 to produce a semi-regular mesh of a specific physical object, the pipeline pre-
 18 sented in figure 1 must be processed : physical acquisition (creating a dense
 19 point cloud), cleaning/meshing (removing redundant points and noise inherent
 20 to acquisition process, and creating an irregular triangle mesh), and then semi-
 21 regular remeshing (Payan et al., 2015). This pipeline is long and fastidious,
 22 especially as these three stages are performed independently.

23 Our original idea is to make the design of semi-regular meshes easier, by
 24 simplifying the classical pipeline shown above. This paper, that is an extended
 25 version of (Peyrot et al., 2014), presents a *coarse-to-fine* approach that allows
 26 an acquisition system to provide semi-regular meshes as output, thus avoiding a

27 remeshing process. We focused on stereoscopic systems, because stereoscopy is
 28 an increasing field of interest in surface reconstruction, due to its rapidity and
 accuracy.

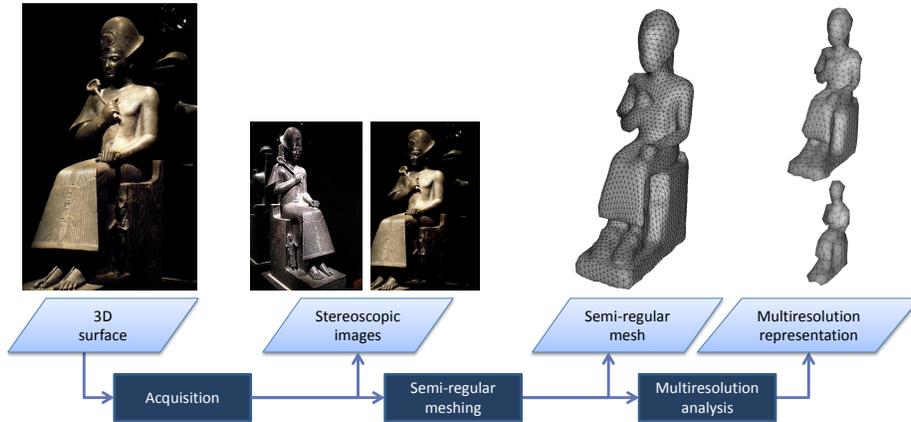


FIGURE 2: Our 3D reconstruction technique that produces a semi-regular mesh directly from stereoscopic images.

29

30 Our method, depicted in figure 2, relies on an analysis of the stereoscopic
 31 images to get a base mesh that captures the salient features of the original ob-
 32 ject, followed by a coarse-to-fine meshing that generates the semi-regular output.
 33 The most innovative part of our algorithm is the use of the stereoscopic images
 34 as parameterization domain to create the semi-regular mesh.

35

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

42

43 **2. Background**

44 *2.1. Semi-regular meshes*

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

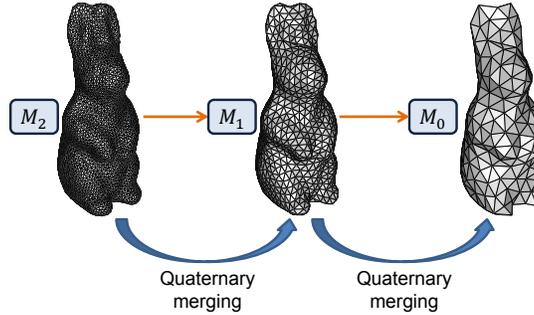


FIGURE 3: Semi-regular mesh of the model RABBIT ($L = 3$ levels of resolutions).

47

48 This merging process can be applied $(L - 1)$ times to M_{sr} until obtaining a
49 base mesh M_0 that represents the lowest resolution of M_{sr} (M_{sr} can be seen as
50 M_{L-1}). A semi-regular mesh is sometimes called a subdivision mesh, because
51 a subdivision scheme is applied on the mesh at resolution l to generate the
52 semi-regular mesh at the finer level of resolution $(l + 1)$.

53 *2.2. Presentation of two prior surface reconstruction methods*

54 We now present two prior reconstruction methods similar to our proposal,
55 because they are based on multi-view images and use a parameterization. Inter-
56 ested reader will find a complete presentation of general reconstruction methods
57 in (Seitz et al., 2006).

58 The method proposed in (Park et al., a) combines the advantages of geo-
59 metric and photometric techniques, thanks to the surface parameterization. It
60 consists in associating a *Multi-View Stereo (MVS)* reconstruction process that
61 relies on a correspondence between pixels from different multi-view images, and
62 a *Shape from Shading* method that utilizes the surface reflectance. The authors

63 use two cameras, an array of lights and a rotation table on which the object
 64 is put (see figure 4). The rotation table allows to acquire several images of the
 65 object at different points of view, whereas only one light at a time is turned on
 to provide different lighting configurations.

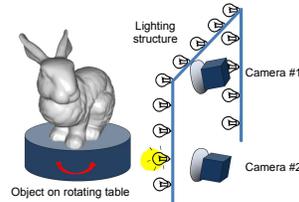


FIGURE 4: Image acquisition system presented in (Park et al., a) (image of (Park et al., a)).

66

67 First of all, the technique *Structure from Motion* (Snavely et al., 2006) ge-
 68 nerates the 3D point cloud of the scanned object. The multi-view method *MVS*
 69 of (Hernandez et al., 2008) is then used to generate a depth-map, and the base
 70 mesh. The third step consists in creating a parameterization by *charts* (Zhou
 71 et al., 2004) (see figure 5). Finally, from the parameterization and the normals
 72 estimated at each vertex of the base mesh, a refinement procedure is applied,
 73 leading to high-quality reconstructions.

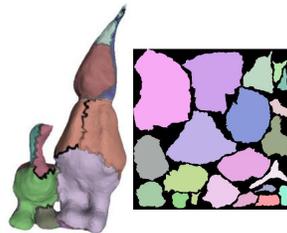


FIGURE 5: Charts defined on the base mesh, and its associated parameterization (image of (Park et al., a)).

74

75 Another relevant approach is proposed in (Pietroni et al., 2011). The authors
 76 present a quadrangular remeshing technique based on a global and low distor-
 77 tion parameterization of different kinds of surfaces (polygonal meshes, point

78 clouds...). The principle, illustrated in figure 6, is to first generate a set of dis-
 79 tance maps U^i of the input data. Then, each image U^i is parameterized into a
 80 2D planar domain, while controlling the resulting distortion at the frontiers of
 81 the images in the final parameterization. Finally, a sampling in the parameteri-
 82 zation domain creates a quadrangular semi-regular mesh.

83

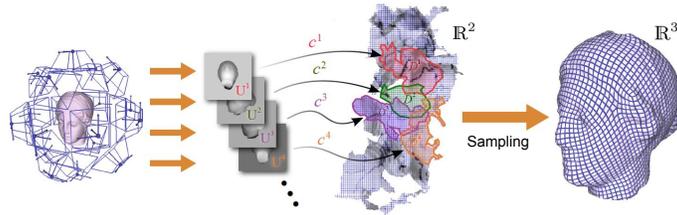


FIGURE 6: Overview of (Pietroni et al., 2011)'s method (image of (Pietroni et al., 2011)).

84 **Discussion** These two parameterization-based methods are reliable. However,
 85 we cannot refer to (Park et al., a) to get a semi-regular mesh directly from
 86 stereoscopic images, as a coarse 3D mesh must be built before creating the
 87 parameterization. The other method, (Pietroni et al., 2011), is closely related
 88 to our goal, it requires a cross-field technique that might be complex and uses
 89 triangles embedded in \mathbb{R}^3 . *A contrario*, our method strives to minimize the
 90 use of the 3D connectivity by using the stereoscopic images as parameterization
 91 domain and a *coarse-to-fine* approach.

92 3. Presentation of our semi-regular reconstruction method

93 3.1. Overview

94 To highlight the interest of our approach, we first present the classical pipe-
 95 line to get a semi-regular mesh of a physical object with a stereoscopic system
 96 (Figure 7(a)).

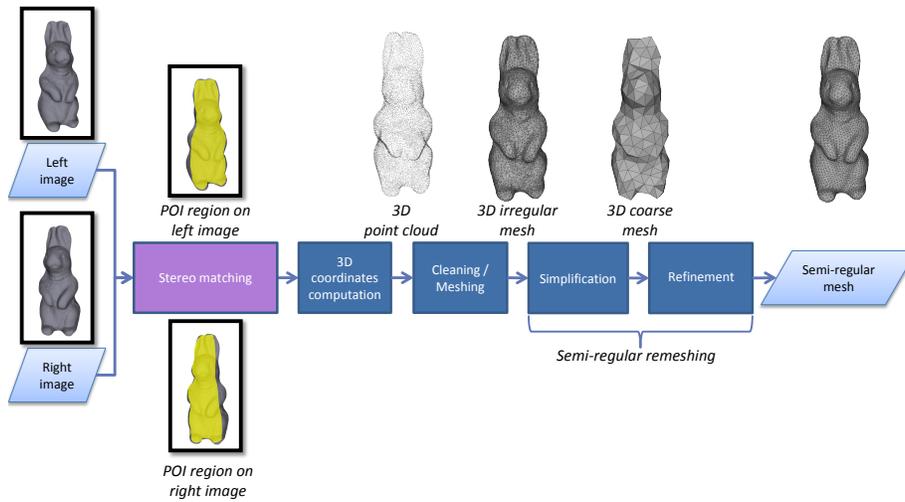
- 97 1. **Stereo matching** The goal is to find the *Pixels Of Interest (POI) region*
 98 in the two images that represents the physical object (Scharstein and Sze-
 99 liski, 2002). The POI region gathers the couples of pixels that correspond

100 to a same point in the 3D space through both cameras (yellow parts on
101 the left and right stereoscopic images). The POI region is only a subset of
102 the stereoscopic images since it is impossible to capture the same set of
103 3D points from two different points of views.

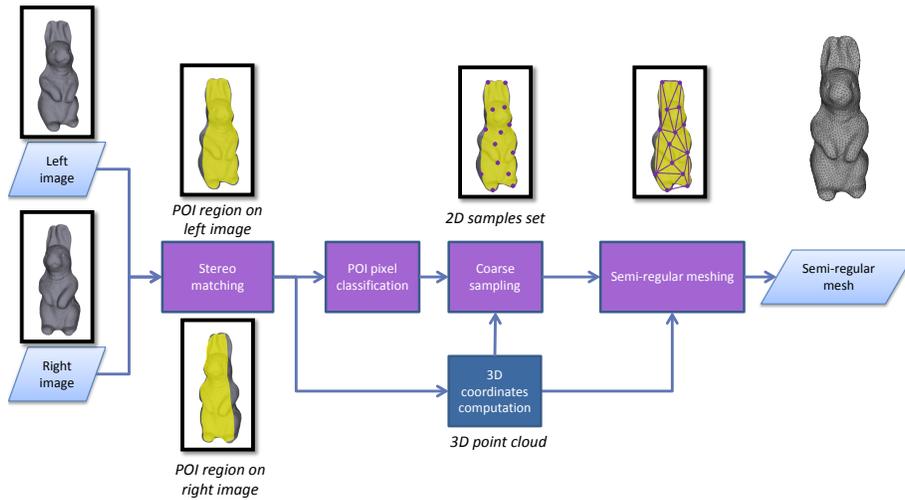
- 104 2. **3D coordinates computation** The coordinates of the 3D points are
105 computed for all the pixels belonging to the POI region (Hartley and
106 Zisserman, 2004). These two first steps are done by the acquisition system.
- 107 3. **Cleaning/meshing** The 3D point cloud must be cleaned, and then trian-
108 gulated, leading to a dense irregular mesh. This is the second independent
109 process.
- 110 4. **Simplification** The semi-regular remeshing can now be done (third inde-
111 dependent process) : the irregular mesh is first simplified to obtain a coarse
112 mesh corresponding to the base mesh of final semi-regular mesh. During
113 this stage, a parameterization of the irregular mesh vertices is generally
114 computed onto this base mesh.
- 115 5. **Refinement** The base mesh is subdivided several times (1 :4 subdivision)
116 to create the different resolutions of the final semi-regular mesh. Generally,
117 the aforementioned parameterization optimizes the positioning of the new
118 vertices added by subdivision.

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

- 122 1. **Stereo matching** This stage is identical to the one in the classical ap-
123 proach.
- 124 2. **POI pixel classification** The goal is to detect the feature lines in the
125 POI region. The creation of the base mesh will be guided by these fea-
126 ture lines to ensure that the geometrical features are preserved on the
127 final semi-regular output. Moreover, such assertion greatly improves the
128 reconstruction quality.



(a) Classical pipeline of semi-regular reconstruction.



(b) Proposed semi-regular reconstruction method.

FIGURE 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.

129 **3. Coarse sampling** A coarse sampling, constrained by the feature lines, is
 130 done to retrieve a set of 2D samples that will be later the vertices of the
 131 base mesh. This stage is based on 2D Poisson-disk sampling, to ensure a
 132 good distribution of the samples over the POI region.

133 **4. Semi-regular meshing** The set of 2D samples is first triangulated to
 134 obtain a 2D base mesh. Then, this 2D base mesh is subdivided several
 135 times, to get a 2D semi-regular mesh of the POI region. Finally, the 3D
 136 semi-regular mesh is obtained by computing the 3D coordinates associated
 137 to the 2D samples.

138 3.2. POI pixel classification

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

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

$$T_{p(u,v)} = \sum_{u'=u-n}^{u'=u+n} \sum_{v'=v-n}^{v'=v+n} \vec{N}^i \cdot \vec{N}^{j^t}, \quad (1)$$

142 where \vec{N}^i is the 3D normal associated to the neighbor pixel $p'(u', v')$, and n
 143 depends on the size of the considered neighbor region of p . The three eigenva-
 144 lues of $T_{p(u,v)}$ are then computed with the Jacobi operator, and a thresholding
 145 operation performs the segmentation of high curvature area. In order to reduce
 146 the runtime and benefit the independence of the operation at each pixel p , this
 147 classification is GPU-parallelized.

148
 149 However, this classification is not precise enough to be exploited as it is.
 150 A parallelized thinning technique (Zhang and Suen, 1984) is thus applied to
 151 the 'high curvature' pixels to finely detect the sharp edges. Thinning a set of

1. In the current version, the curvature values are calculated with the technique of (Park et al., b) 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 (DTA) that computes the 3D normals directly from stereoscopic images.

152 neighbor pixels consists in generating a skeleton (*i.e.* a set of median lines) that
153 presents the same topology as the related shape.

154

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

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

161 Figure 8 depicts several results of classification obtained with this method.

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

165 3.3. Coarse sampling

166 This stage is inspired by the feature-preserving Poisson-disk sampling for
167 surfaces of (Peyrot et al., 2015), which is based on a dart throwing.

168 This approach can be efficiently adapted to our setting : the sampling domain
169 Ω becomes the POI region of the stereoscopic images (instead of a surface mesh
170 in (Peyrot et al., 2015)), and the feature lines detected by the previous stage
171 guide the distribution of 2D samples.

172 However, as the output 2D samples of this stage will define the vertices of
173 the base mesh, they must be consistently distributed *over the surface* of the
174 object (and not especially *over its stereoscopic images*). Therefore the sampling
175 is done onto the POI region, to benefit from its implicit 2D connectivity, but
176 the distances between samples are computed in the 3D space with Dijkstra’s
177 algorithm (Dijkstra, 1959).

178 The principle of the dart throwing on a 2D image is the following : i) one
179 pixel in the 2D domain Ω is chosen randomly, ii) a disk is computed around
180 it, according to a radius R that depends on the target number N of samples
181 and a density function, iii) this pixel is considered as a valid sample if the disk

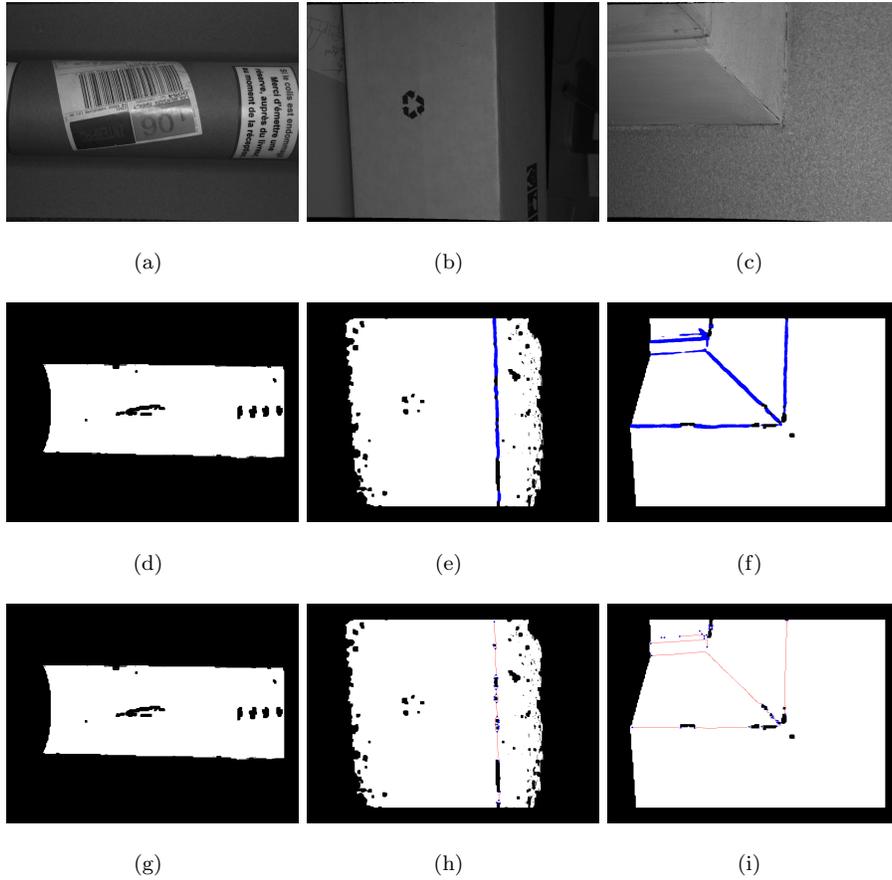


FIGURE 8: Detection of the feature lines *via* our classification technique. First row : left stereoscopic images obtained with our scanner (models PIPE, BOX and WALL). Second row : detection of high curvature areas (in blue). Third row : resulting classification after thinning : white, red and blue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

182 does not intersect the disks relative to the samples already accepted (ensuring
 183 a minimal distance between the samples).

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

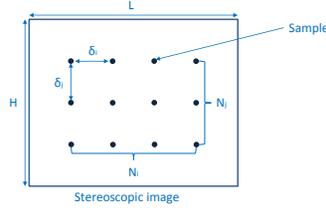


FIGURE 9: Example of uniform sampling performed on one stereoscopic image.

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

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

196 where S_r is the spatial resolution of the scanner (0.3mm in our case). With some
 197 objects, it can be convenient to realize an adaptive sampling, to better preserve
 198 the geometrical features for instance. In that case, the radius will depend on the
 199 surface curvature according to the following equation (Peyrot et al., 2015) :

$$R = \frac{1}{3} \cdot \max(\delta_i, \delta_j) \cdot S_r \cdot (1 + e^{C \cdot \lambda_2} + e^{C \cdot \lambda_3}). \quad (3)$$

200 Empirically, we put $C = -8.0$ for the pixels of the class *sharp features*, and
 201 $C = -6.0$ for the pixels of the class *smooth regions*. λ_2 and λ_3 are the eigenvalues

202 of the tensor $T_{p(u,v)}$ computed in section 3.2. In this formulation, the *corners*
 203 pixels keep the minimum radius given by equation (2). To determine the disks
 204 associated to the samples in function of the radius R , we recall that we use
 205 Dijkstra’s algorithm to compute geodesic distances between 3D points, while
 206 using the connectivity of the 2D sampling domain Ω . Therefore, a disk does
 207 not depend on the Euclidean distance between two given 2D samples, but on
 208 the sum of the lengths of the 3D segments defined by the shortest path in the
 209 POI region, as shown in figure 10. As output of this stage, we get a set of 2D
 210 samples, that ensures a good distribution of the vertices of the 3D base mesh
 211 all over the scanned surface.

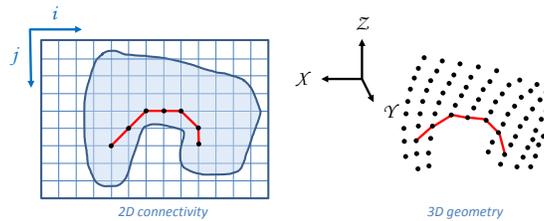


FIGURE 10: Computation of a geodesic distance between two points of the surface in \mathbb{R}^3 (right image), driven by the shortest path between the associated pixels in the POI region (light blue region, left image).

212 3.4. Semi-regular meshing

213 We now present how to generate a semi-regular mesh directly from the set
 214 of 2D samples defined previously. It is a three-stage process : creation of the 3D
 215 base mesh from the set of 2D samples, refinement by iterative subdivisions to
 216 get a 2D semi-regular mesh, and fitting in 3D space.

217 *Creation of the base mesh.* The base mesh is obtained *via* a constrained Voronoi
 218 relaxation (Lloyd, 1982) of the samples in the stereoscopic image domain,
 219 followed by the triangulation of the relaxed samples (via the dual of the Voronoi
 220 diagram (Rong et al.)). In our context, the Voronoi relaxation consists in first
 221 computing a Voronoi diagram of the pixels in function of the set of samples, and
 222 then displacing each sample to the centroid of its cell. This process is iteratively

223 repeated until convergence. The relaxation greatly improves the mesh quality,
 224 when comparing with the triangulation that we could obtain directly from the
 225 initial voronoi diagram.

226 In this work, the Voronoi diagrams are generated with (Munshi et al., 2011),
 227 that is a GPU implementation of popular Dijkstra’s algorithm (Dijkstra, 1959).
 228 We had to adapt this algorithm to process stereoscopic images. Moreover, to
 229 preserve the feature lines on the created base mesh, we added a constraint
 230 during the relaxation : the new samples must belong to the same class than the
 231 initial samples (*corners, sharp features or smooth regions*). In other words, if
 232 after relaxation a sample is moved to a pixel which does not belong to the same
 233 class, then the sample is displaced to the closest pixel of the same class. This
 234 technique is straightforward, but produces nice triangulations, while preserving
 235 geometrical features of the scanned object, as shown in figure 11. This figure
 236 shows also the poor triangulation obtained if the constraint is not included.

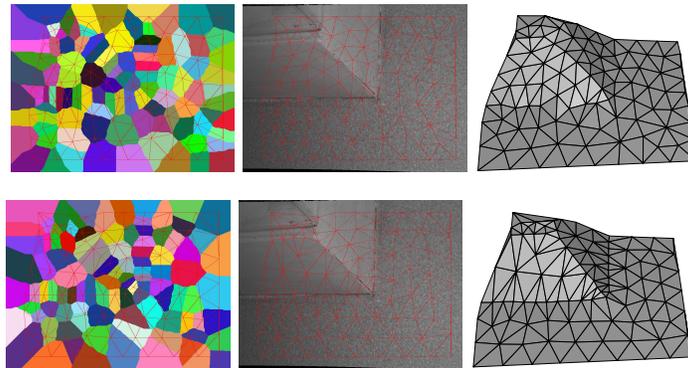


FIGURE 11: Base mesh generated by our Voronoi relaxation without (first row) or with (second row) the constraint on the feature lines. Left : final Voronoi diagram and triangulation. Middle : the same triangulation on the left stereoscopic image. Right : the resulting 3D base mesh.

237 *Refinement.* A 2D semi-regular mesh is first obtained by applying several mid-
 238 point subdivisions (Chen and Prautzsch, 2012) to the base mesh of the left
 239 stereoscopic image (see figure 12). Then, the surface fitting will embed the semi-

240 regular mesh in the 3D space.

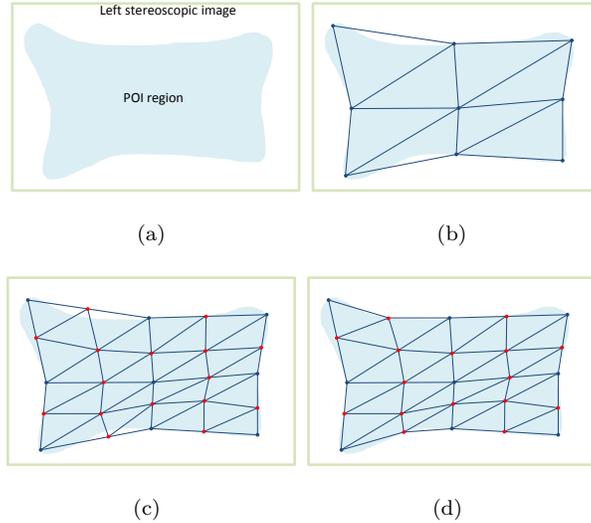


FIGURE 12: Generation of the 2D semi-regular mesh. (a) Left stereoscopic image and its POI region; (b) 2D base mesh; (c) Subdivision; (d) Displacement of the new vertices (red ones).

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

246 In the second case, if we use the same technique, the resulting triangles will
247 be badly shaped (see figure 13(a)), which globally decreases the mesh quality.
248 To reduce such artifacts, we choose to keep the vertices "fallen in a hole" in the
249 2D domain, and so in the 3D space. Figure 13(b) shows that, with this technique,
250 the triangles filling the holes are better shaped. Note that instead of using this
251 simple scheme, one could use an interpolating scheme such as *Butterfly* (Egli
252 and Dussault, 2001).

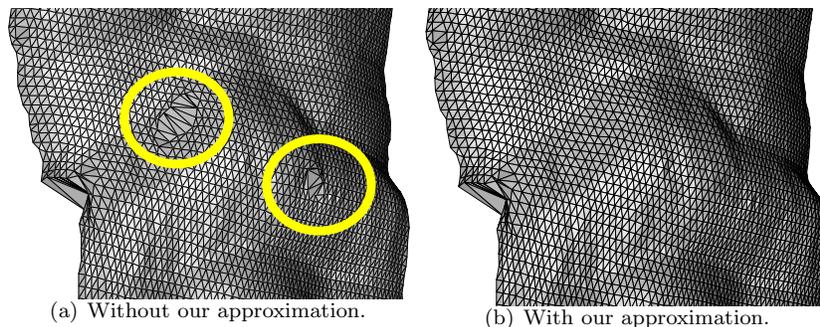


FIGURE 13: Technique proposed to fill the holes during the refinement.

253 **4. Experimental results**

254 *4.1. Visual results*

255 All our results are generated with a single pair of stereoscopic images ob-
 256 tained with a hand-held scanning system. Figure 14 gives an overview of our
 257 method on the model FACE.

258 From the stereoscopic images (a), the POI region (b) is defined, and the base
 259 mesh (resolution 0) is created (c). Then, our coarse-to-fine approach generates
 260 several resolutions (d, e, f). At resolution 5, our semi-regular reconstruction,
 261 with only 43k vertices, is already a good approximation of the original cloud of
 262 250k points given by the stereoscopic system. This is promising in terms of both
 263 compactness and compression. Figure 14(g) also shows the textured semi-regular
 264 mesh, produced in a very simple way, with one stereoscopic image. No additional
 265 texturing technique is necessary as the connectivity of the semi-regular mesh is
 266 generated directly on the image domain. This is another great advantage of our
 267 approach.

268 *4.2. Uniform Vs adaptive sampling*

269 We now study the efficiency of our feature-preserving technique, and the
 270 difference in terms of triangle quality, between the meshes produced with the
 271 uniform/adaptive samplings during the creation of the base mesh (Section 3.3).

272

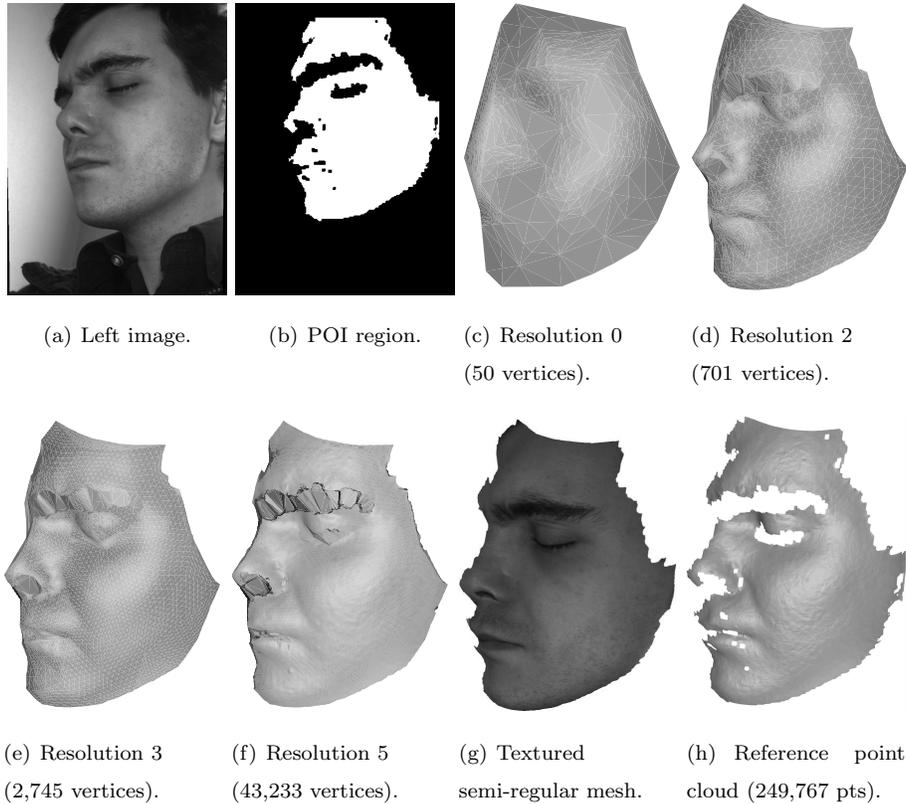


FIGURE 14: Semi-regular reconstruction of the model FACE.

273 Figure 15 shows a reconstruction of a surface having sharp features called
 274 DOOR.

275 Subfigures 15(*b*), (*c*) and (*d*) present the results with the uniform sampling,
 276 whereas subfigures 15(*f*), (*g*) and *h* present the results with the adaptive sam-
 277 pling. We observe on the smooth shadings that the features are globally well
 278 preserved whatever the sampling. Some artifacts along them are visible, but
 279 they are due to the holes in the POI region that generate notches along features
 280 when the base mesh is created (these artifacts would be removed by improving
 281 the stereo matching in the scanning system).

282

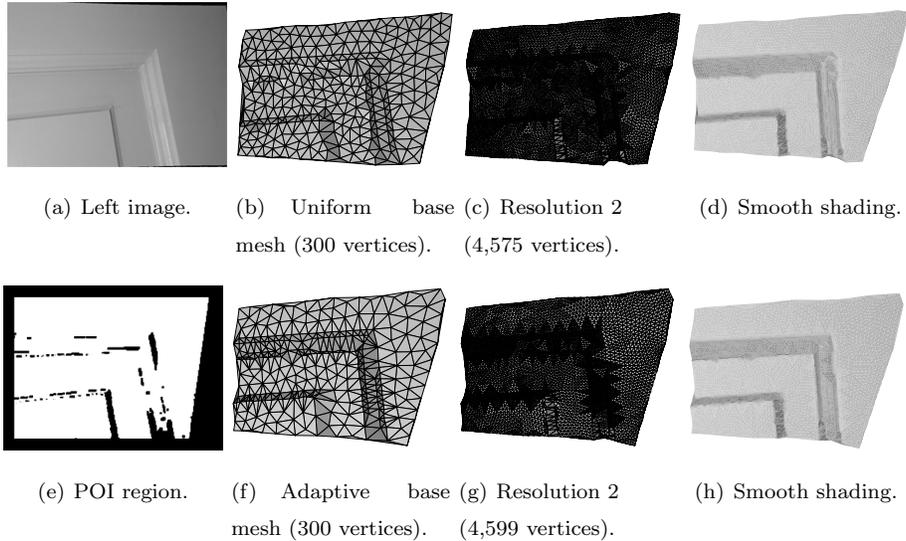


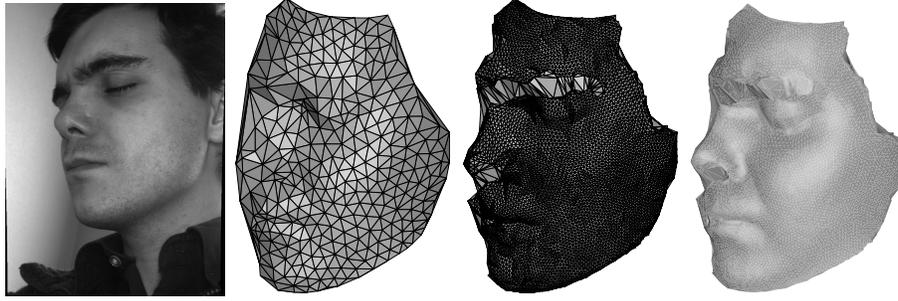
FIGURE 15: Preservation of the geometrical features of DOOR, with the uniform or the adaptive sampling.

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

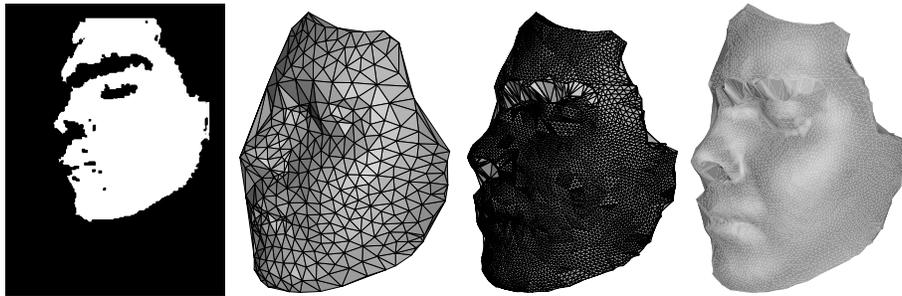
290

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

296



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



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

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

297 *4.3. Runtime*

298 We now evaluate the runtime of our semi-regular meshing on five surfaces
 299 shown in Figure 17. The results have been obtained with an Intel Core i3 CPU
 300 2.30 GHz processor, associated to a 4 GB RAM.

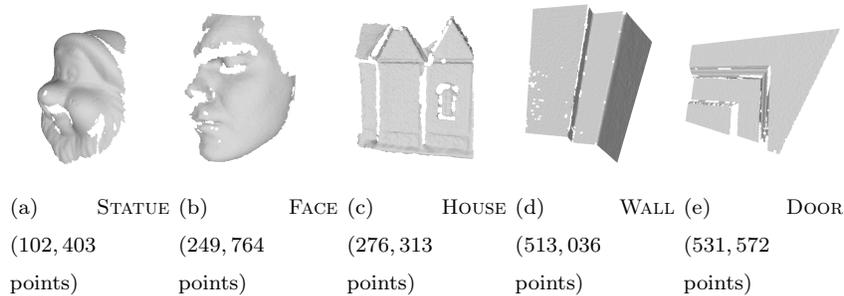


FIGURE 17: Database used to compute the runtimes of Figure 19.

301 Figure 18 shows the runtime in function of the resolution, when the base
 302 mesh has 50 vertices, and eight resolutions. The inferior part of this figure is a
 zoom of the graphic, where the Y-axis spans from 0 to 1.

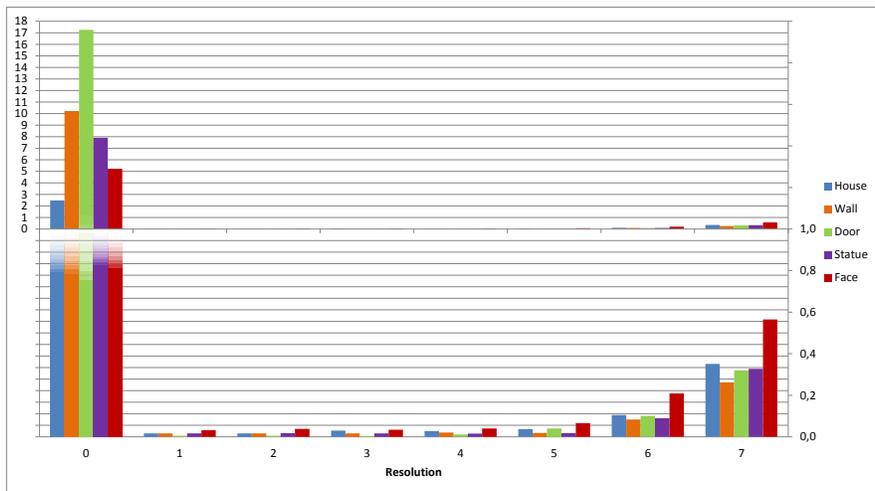


FIGURE 18: Runtime in seconds of our method, per resolution. The inferior part is a zoom of the superior one where the Y-axis spans from 0 to 1.

303
 304 As expected, the most "greedy" resolution is the first one, during which the
 305 base mesh is generated (including the Poisson-disk sampling, the constrained
 306 2D Voronoi relaxation, and the 2D Delaunay triangulation). The obtaining of
 307 the other resolutions is much faster, partly because these steps have been pa-

308 rallelized on GPU. As a proof, the generation of the seventh resolution defined
 309 by 705,281 vertices (528,768 vertices are added) requires less than 0.6 seconds
 310 in the worst case. The higher total runtime to create our semi-regular mesh is
 311 around 17 seconds, for the model DOOR. Nevertheless, this remains very fast.

312 Now, Figure 19 shows the runtime in function of the base mesh density.
 313 For each surface, the curve is obtained by averaging the runtime of five tests.
 314 Indeed, our algorithm is not deterministic (because of the dart throwing), and
 315 the relaxation time depends on the initial sampling. So, we can obtain slightly
 differences at each resolution for a same surface.

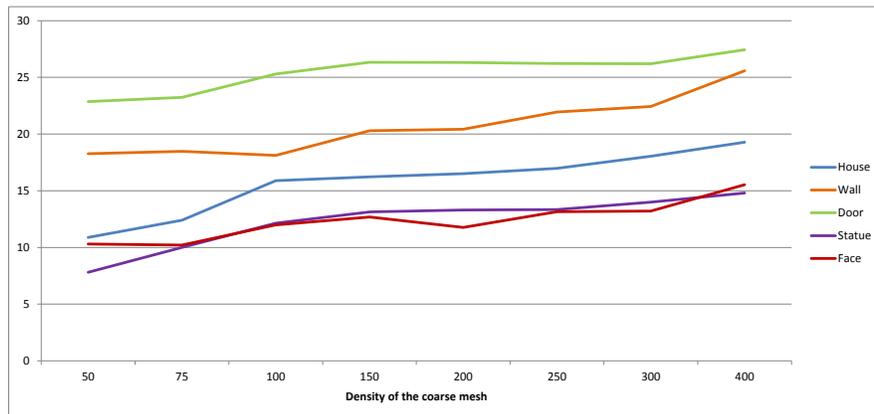


FIGURE 19: Runtimes (in sec.) of our method in function of the number of vertices of the base mesh.

316

317 We observe the linearity of the total runtime with respect to the number of
 318 vertices of the base mesh. The differences between the models is due to the ori-
 319 ginal point cloud density (ranging from 102,403 points for STATUE to 531,572
 320 points for DOOR). The curvature of the scanned surface also influences the run-
 321 time. For instance, WALL contains around 18k points less than DOOR, but it
 322 contains much more pixels classified as *sharp features*. Finally, the mean sam-
 323 pling runtime is slower : 5.70 seconds for WALL, while it amounts to 10.90
 324 seconds for DOOR.

325

326 4.4. Comparison with the classical pipeline

327 We now compare our direct semi-regular meshing with the classical acquisition
328 pipeline to get a semi-regular mesh (*point cloud generation* \rightarrow *triangulation*
329 \rightarrow *semi-regular remeshing*). We use a Voronoi-based triangulation technique to
330 generate the original irregular reference mesh M_{ori} from the point cloud provided
331 by our stereoscopic system. This irregular mesh is then remeshed semi-regularly
332 either with the SDK *SmartMesh* based on the patent Fonteles et al. (2014) and
333 developed by the company (Cintoo3D) (a free trial is available on the website),
334 or with *Trireme* (Guskov, 2007). To our knowledge, they are the only semi-regular
335 remeshing techniques available on the web. Unfortunately, we found out that
336 *Trireme* is not a suitable tool to remesh our data. Indeed, we could not produce
337 any semi-regular meshes without severe degeneracies and outlier triangles. In the
338 contrary, *SmartMesh* always provides manifold semi-regular meshes, in particular
339 because it does not use any parameterization, unlike *Trireme*. Consequently,
340 we only compare the reconstruction errors and runtimes relative to our semi-regular
341 meshing and to *SmartMesh*.

342
343 We first compute the symmetric root mean square distance between the set of
344 vertices of our semi-regular meshes, and the set of vertices of M_{ori} (the original
345 point cloud provided by the acquisition system). It permits to assess the fidelity
346 of our sampling to the reference point cloud. The same distance is calculated with
347 the set of vertices of the semi-regular meshes produced by *SmartMesh*. Figure 20
348 shows the evolutions of these distances depending on the resolutions : the X-axis
349 indicates the associated number of points. We observe that our method presents
350 lower distances than *SmartMesh*. It was expected as our method is approximating,
351 contrary to *SmartMesh* that is interpolating (it optimizes the positions of the
352 vertices such as its semi-regular mesh is close to the reference mesh). Our method
353 has the advantage to determine the majority of vertices among the original point
354 cloud, as the vertices are selected *via* the pixels of the POI region in the image
355 domain. The only vertices that do not exist in the original point cloud are
356 associated to pixels selected outside the POI region

357 during subdivision. Thus, our method is more accurate when considering only
 358 the geometry of the initial surface.

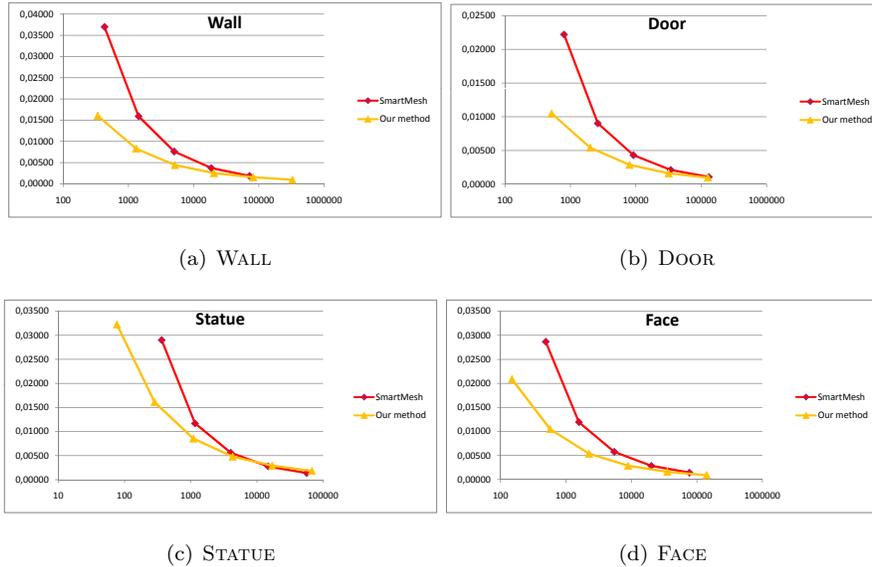


FIGURE 20: Comparison of the geometry sampling obtained with our method and with *SmartMesh* (Cintoo3D), depending on the vertex density of the semi-regular meshes.

358

359 We now assess the fidelity of our semi-regular meshes with respect to the
 360 reference mesh M_{ori} . To achieve this goal, we compute the symmetric Root
 361 Mean Square (RMS) distance between M_{ori} and our semi-regular meshes M_{sr}
 362 (normalized by the diagonal length of the bounding box), which is widespread
 363 used in the state-of-the-art (Payan et al., 2015). However, in our context, this
 364 measure is not suited. Indeed, as explained in section 3.4, our method fills the
 365 holes, in order to make the texturing easier and to enhance the mesh quality.
 366 As a consequence, when measuring the symmetric RMS distances between our
 367 semi-regular meshes and the reference irregular mesh M_{ori} , the distances bet-
 368 ween the triangles filling the holes and the original surface are inevitably high.
 369 It severely corrupts the comparison with *SmartMesh*, as this latter has been ini-
 370 tially developed to preserve the potential borders of a surface and consequently
 371 the holes. So, to make fairly comparisons, we compute the asymmetric RMS

372 distance $RMS(M_{ori} \rightarrow M_{sr})$, which excludes the filled holes : see Figure 21
 373 (the X-axis still indicates the number of points of each resolution). Globally, we
 374 observe that our method is better than *SmartMesh* in the first resolutions. It
 375 can be explained by the fact that our method tends to preserve the geometrical
 376 features in the base mesh, and that it is approximating (Payan et al., 2015). On
 377 the other hand, *SmartMesh* becomes better on the highest resolutions, because
 378 it minimizes the geometric distortion directly onto the original surface, without
 379 any parameterization, which avoids the relative distortion. Furthermore, our
 380 method is penalized by the fact that it works in the image domain, but also by
 381 our feature preservation that positions more vertices on them.

382

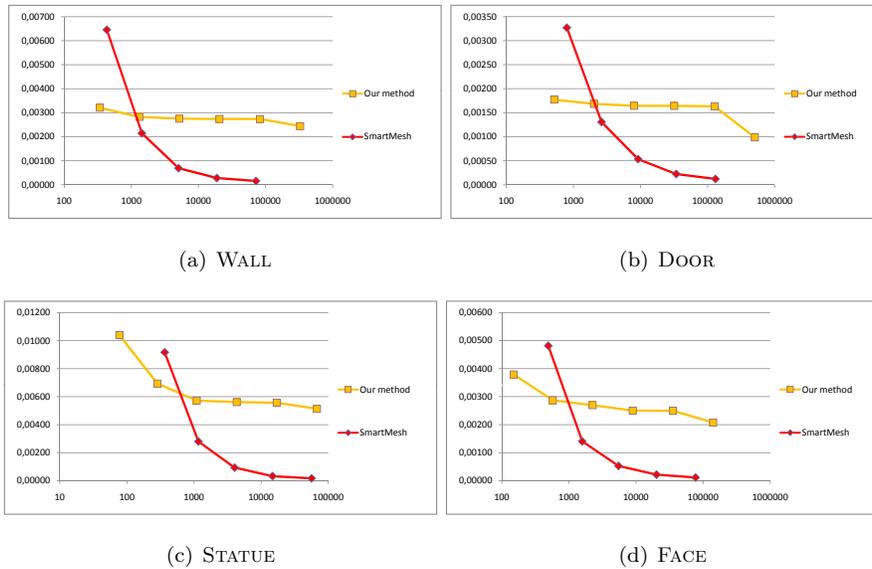


FIGURE 21: addAsymmetric RMS distance $RMS(M_{ori} \rightarrow M_{sr})$ obtained with our method and with *SmartMesh* (Cintoo3D) in function of the resolution.

383 On the other hand, our algorithm is direct and thus significantly faster than
 384 *SmartMesh*. The runtime comparison is summarized in table 1. *SmartMesh* in-
 385 deed takes several minutes to produce the semi-regular meshes : from 2 to 7
 386 minutes in function of the data, excluding the triangulation time, whereas our

387 method needs always less than one minute. Finally, it shows that the proposed
 388 pipeline is a promising alternative to the classical one.

Models	Wall	Door	Statue	Face
Base mesh density	431	800	365	492
<i>SmartMesh</i>	≥ 3 min.	≥ 7 min.	≥ 2 min.	≥ 6 min.
Our method	19.1 sec.	27.7 sec.	10.1 sec.	10.3 sec.

TABLE 1: Runtime comparison in seconds between our method and *SmartMesh* (Cintoo3D),
 to generate 5 resolutions.

388

389 5. Conclusion and perspectives

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

397 The first contribution is an original sampling that creates a base mesh of the
 398 scanned surface. We show that the Poisson-disk sampling developed by (Peyrot
 399 et al., 2013) can be extended to a stereoscopic system, while retrieving the 3D
 400 information necessary to preserve features all along the process. This allows to
 401 take into account the surface geometry, although the sampling is realized on
 402 the stereoscopic images. The second contribution concerns our *coarse-to-fine*
 403 approach that allows to get a semi-regular mesh preserving the geometrical
 404 features as output of our acquisition system, by working mainly in the image
 405 domain.

406 Our pipeline could be easily included into any stereoscopic acquisition sys-
 407 tem. It also has the advantage to create semi-regular output that can be directly

408 textured with the stereoscopic image, and is also much more faster and conve-
409 nient than the classical pipeline.

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

421

422 **Acknowledgements** This work was supported by a grant from *Région Pro-*
423 *vence Alpes Côte d’Azur* in France.

424 Bibliography

425

426 Q. Chen and H. Prautzsch. General midpoint subdivision. *CoRR*,
427 abs/1208.3794, 2012.

428 Cintoo3D. SDK smartmesh. <http://www.cintoo3D.com>.

429 E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische*
430 *Mathematik*, 1 :269–271, 1959.

431 R. Egli and J.-P. Dussault, editors. *Technique butterfly généralisée*, Dijon,
432 France, p. 133-136, 2001. Actes de Compression et Représentation des Si-
433 gnaux Audiovisuels.

- 434 L. Hidd Fonteles, A. Meftah, M. Antonini, and F. Payan. Method, system and
435 computer program product for 3d objects graphical representation. FR-13, n
436 4152728.3-1502, 2014, CNRS et Universit de Nice - Sophia Antipolis, 2014.
- 437 I. Guskov. Manifold-based approach to semi-regular remeshing. *Graphical Mo-*
438 *dels*, 69(1) :1–18, 2007.
- 439 R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*.
440 Cambridge University Press, second edition, 2004.
- 441 E. C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo.
442 *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3), March 2008.
- 443 S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Infor-*
444 *mation Theory*, 28 :129–137, 1982.
- 445 M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for sur-
446 faces of arbitrary topological type. *ACM Trans. Graph.*, 16(1), January 1997.
- 447 A. Munshi, B.-R. Gaster, T.-G. Mattson, J. Fung, and D. Ginsburg. *OpenCL*
448 *Programming Guide*. Prentice Hall, 2011.
- 449 J. Park, S. N. Sinha, Y. Matsushita, Y.-W. Tai, and I. S. Kweon. Multiview
450 photometric stereo using planar mesh parameterization. *Computer Vision,*
451 *IEEE International Conference on*, 0, a.
- 452 M. K. Park, S. J. Lee, and K. H. Lee. b.
- 453 F. Payan and M. Antonini. Mean square error approximation for wavelet-based
454 semiregular mesh compression. *Transactions on Visualization and Computer*
455 *Graphics (TVCG)*, 12, 2006.
- 456 F. Payan, C. Roudet, and B. Sauvage. Semi-regular triangle remeshing : a
457 comprehensive study. *Computer Graphics Forum*, 34 :86–102, February 2015.
- 458 J.-L. Peyrot, F. Payan, and M. Antonini. Feature-preserving Direct Blue Noise
459 Sampling for Surface Meshes. In *Eurographics (Short Papers)*, pages 9–12,
460 2013.

- 461 J.-L. Peyrot, F. Payan, N. Ruchaud, and M. Antonini. Stereo reconstruction of
462 semiregular meshes, and multiresolution analysis for automatic detection of
463 dents on surfaces. In *Proceedings of IEEE International Conference in Image*
464 *Processing (ICIP)*, Paris, France, october 2014.
- 465 J.-L. Peyrot, F. Payan, and M. Antonini. Direct blue noise resampling of meshes
466 of arbitrary topology. *The Visual Computer*, 31 :1365–1381, september 2015.
- 467 N. Pietroni, M. Tarini, O. Sorkine, and D. Zorin. Global parametrization of
468 range image sets. *ACM Trans. Graph.*, 30(6), December 2011.
- 469 G. Rong, T.-S. Tan, T.-T. Cao, and Stephanus. In Eric Haines and Morgan
470 McGuire, editors, *SI3D*.
- 471 D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame
472 stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3) :7–42, April
473 2002. ISSN 0920-5691.
- 474 S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison
475 and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings*
476 *of the 2006 IEEE Computer Society Conference on Computer Vision and*
477 *Pattern Recognition - Volume 1*, CVPR '06, 2006.
- 478 N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism : Exploring photo
479 collections in 3d. *ACM Trans. Graph.*, 25(3) :835–846, July 2006.
- 480 T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital
481 patterns. *Commun. ACM*, 27(3), March 1984.
- 482 K. Zhou, J. Synder, B. Guo, and H.-Y. Shum. Iso-charts : Stretch-driven mesh
483 parameterization using spectral analysis. Eurographics, July 2004.