©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037

Blue noise Sampling of surfaces from stereoscopic images

Frédéric Payan, Jean-Luc Peyrot, Marc Antonini; Laboratory I3S - University of Nice - Sophia Antipolis and CNRS (UMR 7271); France

Abstract

We propose an original technique to sample surfaces generated by stereoscopic acquisition systems. Our motivation is to simplify the long and fastidious sampling pipeline, for such acquisition systems. The idea is to make the sampling of the surfaces directly on the pair of stereoscopic images, instead of doing it on the meshes created by triangulation of the point clouds given by the acquisition system. More precisely, we present a feature-preserving sampling, done directly in the stereoscopic image domain, while computing the inter-sample distances in the 3D space, in order to reduce the distortion due the embedding in $\mathbb{R}^3$*. We focus on Poisson-disk sampling, because of its nice blue noise properties. Experimental results show that our method is a good trade-off between the direct sampling methods that are time-consuming, and the methods based on parameterizations that alter the final sampling properties.*

Introduction

Acquisition systems are now able to create massive point clouds, to ensure the preservation of the finest details. Consequently the resulting meshes are also dense, most of times oversampled, and cannot be easily managed by any workstation or mobile device with limited memory and bandwidth.

To overcome this problem, many methods of mesh simplification [1], remeshing [2, 3], resampling [4], etc. have been developed during the last two decades, to reduce the size of geometrical data and to make their process easier, or just... possible. However, it could be relevant to tackle the problem upstream, i.e. during the acquisition process.

The motivation of our current work is to simplify the long and fastidious sampling pipeline, here for the specific case of stereoscopic acquisition systems. In this paper, we generate the sampling pattern of the final surfaces directly from the stereoscopic images, instead of making it on the meshes created by triangulation of the dense point clouds at the output of the stereoscopic systems. Hence, the generated 3D data are usable without any additional resampling stage, and the number of sampling points can be controlled from the beginning of the acquisition.

Challenges and contributions

To enable the sampling of surfaces directly in the stereoscopic image domain, there are two challenges to overcome:

1. The sampling of surfaces directly in the stereoscopic images is similar to a parameterization-based approach, as the stereoscopic images represent a parameterization domain of the surface geometry. We benefit from the advantage of such an approach, which is the implicit connectivity of the sampling domain. The drawback is that parameterization generally leads to distortion, once the sampling embedded in the 3D space. In other words, the quality of sampling patterns in the parameterization domain can be significantly decreased when moving from parameterization domain to the surfaces. To tackle this problem, we propose to compute the distances between samples in the 3D space, as the corresponding 3D coordinates are known for each pair of pixels in the stereoscopic images. Thus, our method can be seen as an *hybrid* blue noise sampling method, driven by the connectivity of the stereoscopic images, but computing the real distances between samples in the 3D space.
2. We would like to preserve the geometrical features of the scanned surfaces in the sampling pattern. Inspired by [5], we propose to extract these features directly from the stereoscopic images, by detecting the corresponding feature lines in the stereo images.

In this paper, we focus on Poisson-disk sampling that received considerable attention in computer graphics because of its nice blue noise properties [6], and its capability to avoid aliasing artifacts [7]. Indeed, a Poisson-disk distribution generates sampling patterns that satisfy a uniform distribution (a minimum distance is ensured between samples), but irregular within the domain (the samples do not lie on a spatial regular lattice). These characteristics are particularly relevant for many applications such as rendering, imaging, texturing, geometry processing and numerical simulations [8], [9], [10]. However, our approach could be easily extended to other sampling patterns.

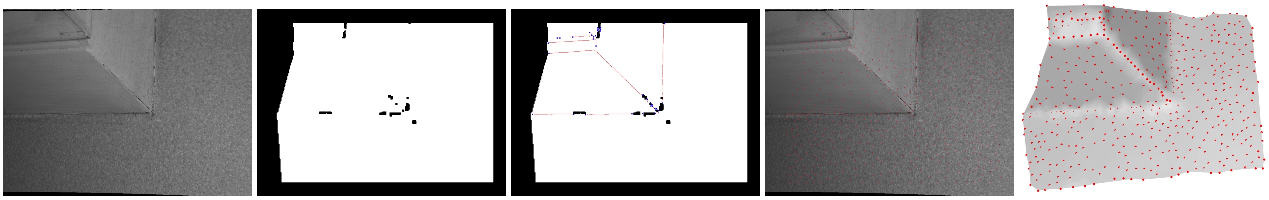
There are several prior works about the blue noise (re)sampling of surfaces. Two kinds of approaches exist: directly on the surfaces [11, 4], or on a 2D parametric domain of the surfaces [12]. As expected, the first approaches are efficient in term of sampling quality but are time-consuming, because of the computation of geodesics over the surface (to estimate the distances between samples). The second approaches overcome the issue relative to the computation of geodesics, but the generated patterns may suffer from distortion as the parameterization must be computed. Finally, our hybrid method benefits from the advantages of these two kinds of approaches.

Overview

Figure 1 gives an overview of our hybrid sampling method. Figure 2 shows the data produced all along the process. Let us acquire a pair of stereoscopic images (Figure 2(a)). Our method consists in four stages, described below.

1. **Stereo matching** The goal of this stage is to determine the *Pixels Of Interest (POI) region*, which gives the part of the physical object that can be reconstructed from the pair of images [13]. The POI region is defined by a mask *I* (Figure

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037



(a) Left stereoscopic image. (b) Mask of the POI region. (c) Detected feature lines. (d) 2D sampling pattern. (e) Final 3D sampling pattern (500 samples).

Figure 2. *From stereo images to 3D sampling pattern.*

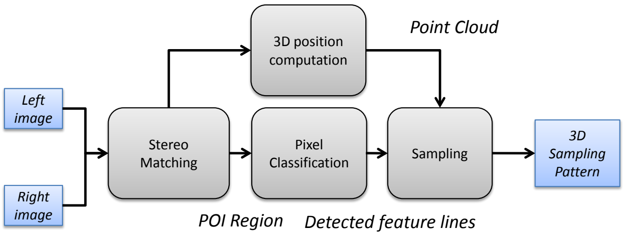


Figure 1. *Overview of our hybrid sampling method.*

2(b)), which is a binary image where the white pixels define the POI. The POI region is only a subset of the stereoscopic images as it is impossible to capture the same set of 3D points from two different points of view.

2. **Pixel classification** Its goal is to detect the feature lines in the POI region (Figure 2(c)). The subsequent sampling will be driven by these feature lines to ensure that corners and sharp features are preserved in the sampling pattern.
3. **3D position computation** This stage consists in computing the position, in the 3D space, of the pixels belonging to the POI region, by triangulation [14].
4. **Sampling** This step distributes the 3D samples on the scanned surface. The selection of samples is done in the stereoscopic image domain (Figure 2(d)), but the distances between them are computed in the 3D space to ensure the blue noise properties of the final 3D sampling pattern (Figure 2(e)). Its geometrical features are also preserved because the sampling is done on the feature lines first, and then on the other parts of the POI region.

Description of our algorithm

Stereo matching

The stereo matching is a well-known technique to gather the couples of pixels that correspond to a same point in the 3D space through two cameras (see Figure 3). This set of pixels defines the *POI region*, shown in green on Figure 3(c). In our algorithm, this POI region allows us to generate a mask I, shown on Figure 3(d), where the white pixels and the black pixels represent the pixels belonging to the POI region, or not, respectively. This mask I determines the surface that can be reconstructed in 3D space or, in our case, sampled.

Pixel Classification

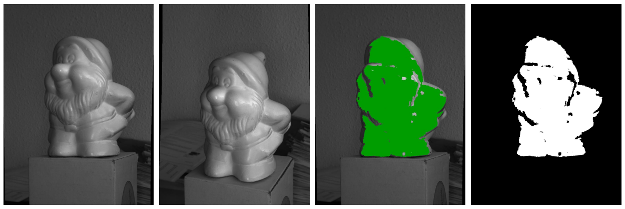
This classification allows us to detect the sharp features of the acquired surface, in order to preserve them during its sampling. As the sampling will be done on the stereoscopic images, the idea is to classify the pixels in function of the curvature values of the corresponding 3D points. In function of its associated curvature values, we check if a given pixel is considered as *sharp* (belonging to a sharp feature on the surface), as a *corner* (belonging to two sharp features) or as *smooth* (not belonging to a sharp feature). Currently, the curvature values are calculated from the 3D normals in the 3D space with [5]. However, to operate on the images directly, this technique could be replaced by the technique of [15] that computes the 3D normals directly from stereoscopic images.

For each pixel $p(u,v)$ of the POI region, a tensor $T_{p(u,v)}$ is computed according to

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

$\overrightarrow{N'}$ 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. Considering the three eigenvalues of this tensor sorted by decreasing order of amplitude, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, the maximum and minimum curvature values associated to each pixel are given by the two smaller eigenvalues λ_2 and λ_3, respectively. A thresholding is then used to detect the high curvature areas: if $\frac{\lambda_2+\lambda_3}{\lambda_1} \geq 0.05$ for a given pixel, its associated point belongs to a high curvature area. This stage is GPU-parallelized in our algorithm.

At this step we only know if a pixel belongs to a high curvature area: see the second row of Figure 4. We now have to detect the sharp edges precisely. For this, we compute the skeleton (*i.e.* the set of median lines) of the set of pixels belonging to the high curvature areas, thanks to a parallelized technique based on [16]. Finally, a given pixel is classified in



(a) Left image. (b) Right image. (c) POI region. (d) Mask I.

Figure 3. *Creation of the mask I, determining the POI region of a scanned surface from the pair of stereoscopic images.*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037

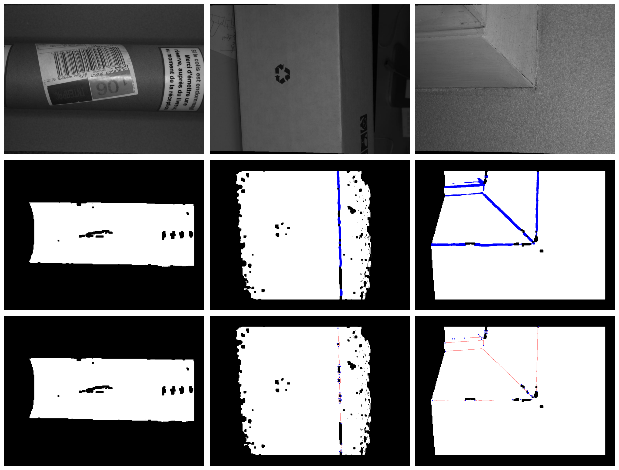


***Figure 4.** Detection of the feature lines* via *our pixel classification. First row: left stereoscopic images of three acquisitions called* PIPE, BOX *and* WALL. *Second row: detection of high curvature areas (in blue). Third row: resulting classification after thinning. White, red and blue pixels represent, respectively, the smooth regions, the sharp features, and the corners.*

- *corner* if it is an intersection of several median lines;
- *sharp* if it lies on one median line;
- *smooth* otherwise.

The third row of Figure 4 depicts several results of classification obtained with this method. We observe that globally our method detects well the sharp features of the scanned surfaces.

3D position computation

This stage just computes the coordinates in $\mathbb{R}^3$ of the points associated to the pair of pixels belonging to the POI region. It can be done simply by triangulation [14], by using the position of the pixels into the two stereoscopic images, and the parameters of the projection function ($3D \to 2D$) for the two cameras.

Sampling

As explained in Section *Challenges and contributions*, we focus in this paper on the Poisson-disk sampling, also called blue noise sampling. Poisson-disk sampling is a random process for selecting a set of samples on a specific domain Ω. Each point must be located at a minimum distance d from any previous point (see Figure 5). One popular technique for Poisson-disk sampling is the *Dart throwing* (DT) [11]. Considering $d = 2R$ the minimum distance required between two samples, the DT consists in: i) picking out randomly a sample s on the domain; ii) drawing a disk of radius R around it; iii) verifying if this disk intersects another disk. If no disk is intersected, the sample s is kept. Otherwise, the sample is discarded (as the red disk in Figure 5). This process is iterated until no more sample can be added on the domain, or until a user-given number of samples is reached. If the radius of the disks is constant, it is called a *uniform* sampling (Figure 5). If the radius depends on a local function relative to the samples, the curvature for instance, it is called an *adaptive* sampling (Figure 6).

Our method is inspired by the *Dart Throwing* for surfaces proposed in [4]. This method is considered as a direct 3D sampling method, because the sampling domain Ω is the triangular mesh associated to the surface sampled (no parameterization). This method also takes into account the sharp features in order to preserve them during sampling, and uses geodesics to guarantee the output sampling quality. To preserve efficiently the sharp features, this method puts samples on the pixels classified as *corners*, then distributes samples among the pixels classified as *sharp*, and finally distributes samples among the set of pixels classified as *smooth*.

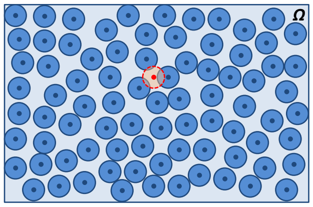


***Figure 5.** Uniform Poisson-disk sampling of a 2D domain. The red disk shows a non-valid sample.*

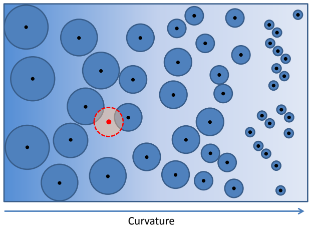


***Figure 6.** Adaptive Poisson-disk sampling of a 2D domain. The red disk shows a non-valid sample.*

The method of [4] can be efficiently adapted to our setting: the sampling domain Ω becomes the POI region of the stereoscopic images, and the feature lines detected during the pixel classification can likewise guide the distribution of 2D samples. Nevertheless, even if the process is done in the 2D domain, we still use the 3D coordinates associated to the pixels (computed in a previous stage) to compute accurately the distances between samples. Moreover, the radius associated to each sample must be now computed in the image domain.

Therefore, to estimate as finely as possible the radius associated to a pixel in function of the number N of samples requested, we first calculate the horizontal and vertical deviations, δ_i and δ_j respectively, between samples when a uniform sampling pattern is realized on an image. It generates a grid of samples of dimension $\mathbb{N}_{\delta_i} \times \mathbb{N}_{\delta_j}$, as depicted in Figure 7. N_i and N_j represent the number of samples per row and per column, respectively ($N_i \times N_j = N$). As the sampling domain Ω is restricted to the POI region, the distances δ_i and δ_j between samples along each dimension, are shrunk by a factor $\frac{L \times H}{\#POI}$, with $\#POI$ the number of pixels of the POI region.

A uniform sampling can be finally realized using the following formulation for the radius.

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

where S_r is the spatial resolution of the scanner ($0.3mm$ in our case). To better preserve the geometrical features of the scanned surface, we choose to make an adaptive sampling, with a radius

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037

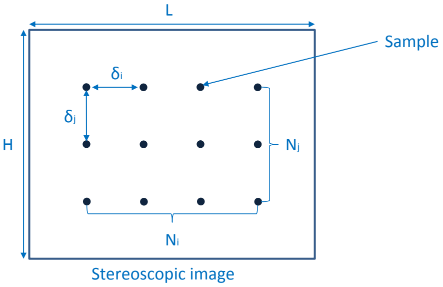


Figure 7. *Example of uniform sampling performed on one stereoscopic image.*

depending on the curvature, according to the following equation:

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

Empirically, we put $C = -8.0$ for the *sharp* pixels, and $C = -6.0$ for the *smooth* pixels. In this formulation, the *corner* pixels keep the minimum radius given by equation (2).

To determine the disk areas associated to the samples in function of the radius R, we use Dijkstra's algorithm [17] to compute geodesic distances between 3D points, while using the connectivity of the 2D sampling domain Ω. Hence, 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 Figure 8. This approach leads to better sampling distributions, as shown in the experimental results.

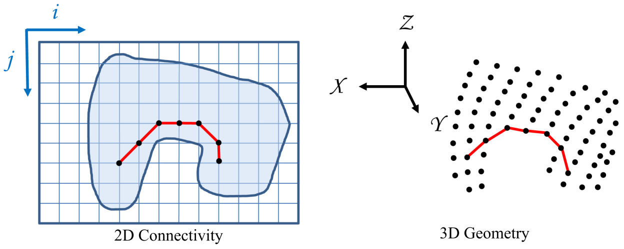


Figure 8. *Computation of a geodesic distance on the surface defined by 3D points, driven by the implicit connectivity of the image domain.*

Experimental results

Visual results

We first present two sampling patterns obtained with our hybrid approach. Figures 9 and 10 show the results obtained on two models Pipe and Box. For each model, we can see the distribution of 500 samples (b) and 1k samples (c) superposed on the left images, and the resulting 3D sampling pattern on the acquired surface: (e) and (f). We observe on Pipe that the samples are distributed all over the surface as all the pixels belong to the same class *smooth*. Note that some regions are not sampled because of the reflectance, which prevents the matching of the pixels in these regions. Concerning Box, we observe that many samples have been distributed along the feature lines, which allows us to preserve the geometrical features efficiently. We can also see the interest of our method on Figure 2(e) (page 2), with the model Wall: corners and sharp features are globally well preserved.

Analysis of the sampling quality

We now evaluate the efficiency of our *hybrid* method, in term of sampling quality. We compare the properties of the sampling patterns generated with our *hybrid* method, and with the 3D direct sampling method of [4] that we consider as ground truth. We also compare our method with a "naive" 2D method that consists in distributing the samples on the stereoscopic images without taking into account the 3D geometry (when the distances between samples are computed). It will highlight the interest of our hybrid method that takes into account the 3D geometry even though the sampling is performed in the 2D domain.

To evaluate the spectral quality of sampling patterns, a power spectrum is usually used. It represents the distribution of distances between samples. From a power spectrum, two statistics are then computed: the radially averaged power spectrum (RAPS) and the anisotropy. The RAPS assesses the radial distribution of the distances between samples, whereas the anisotropy evaluates the radial uniformity of the sampling pattern.

An ideal blue noise or Poisson-disk distribution has a RAPS similar to a step function: a zero-region at low frequencies and a flat high-frequency region. Between them, a sharp transition at the cut-off frequency, proving that a minimum distance between samples is respected. The ideal anisotropy is constant and low beyond the cut-off frequency, meaning that the sampling is directionally independent.

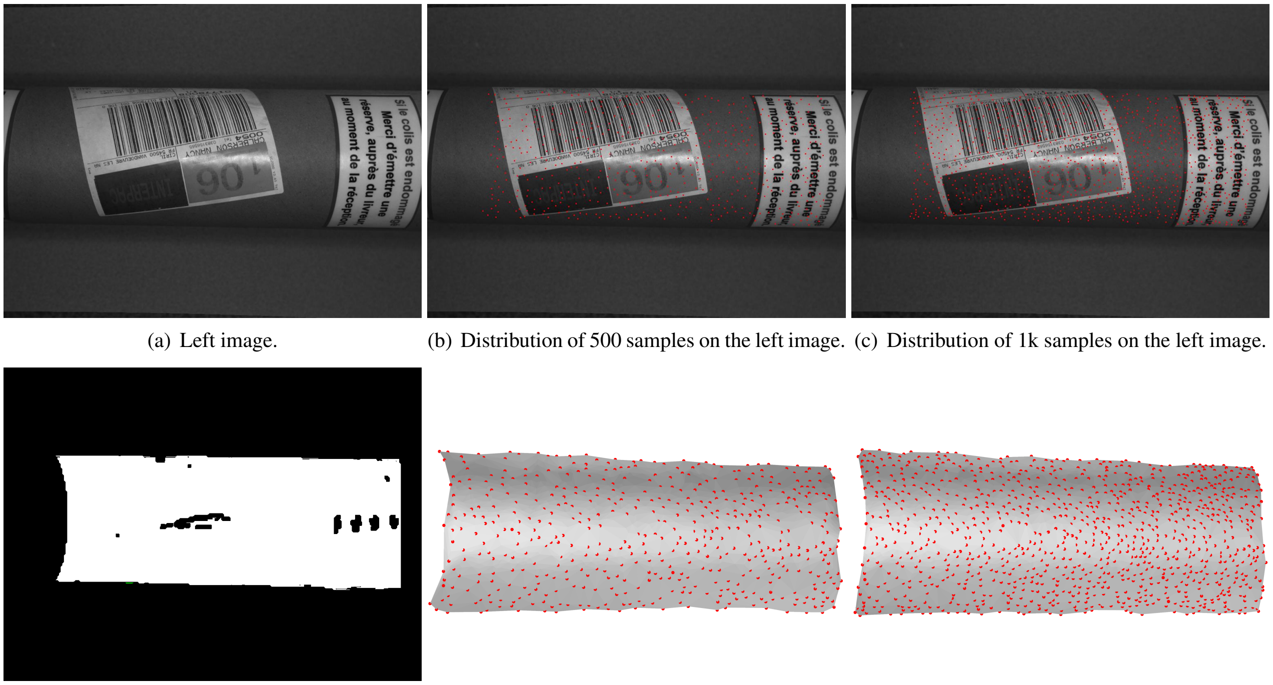
Figures 11 and 12 show the RAPS and the anisotropy of the samples distributed with the three methods on Pipe and Wall. The RAPS and the anisotropy are computed with the tool of [4]. We observe that a 2D naive sampling leads to patterns with poor blue noise properties. The RAPS is far from a step function (a minimal distance is not respected), and the anistropy is not flat.

The sampling properties of the patterns generated with our hybrid method are more satisfactory. Their RAPS are close to the RAPS of the 3D method and respect a minimal distance efficiently. On the other hand, the RAPS of our method oscillate after the cut-off frequency (contrary to the RAPS of the 3D direct method). This is mainly due to the acquisition noise that biases the 3D normals and the size of the disks, consequently. But the RAPS remain satisfactory. Moreover, the anisotropy curves of our method remain flat, as the 3D direct method. On the other hand, their magnitudes are higher.

Despite the higher anisotropy, this analysis shows that our hybrid method generates sampling distributions with nice blue noise properties, close to the 3D direct method, while preserving the sharp features as well.

Time complexity

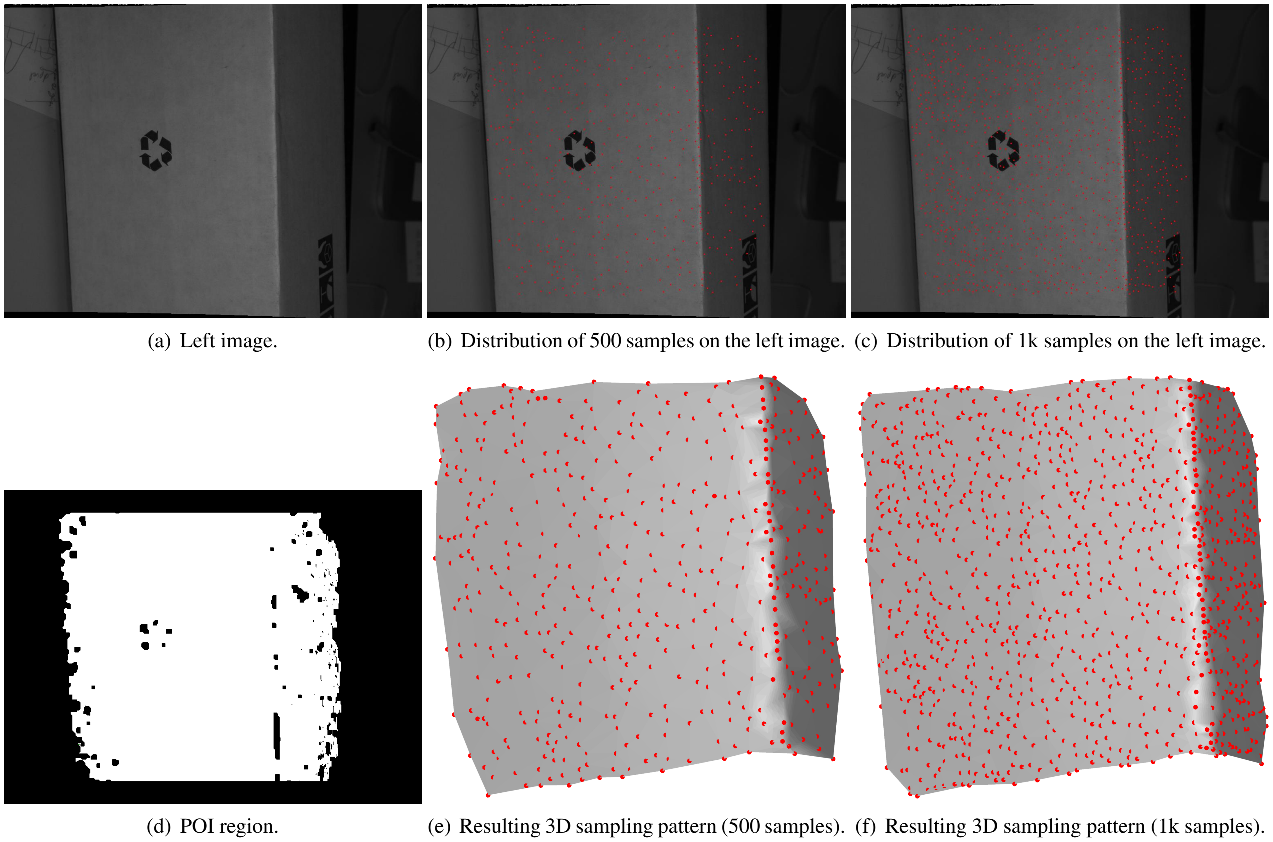
In order to evaluate the time complexity of our method, the runtime of the stages *pixel classification* and *sampling* for the three approaches is given in Table 1 for the models Pipe and Box. The values are obtained by averaging the runtime of 8 tests for each object, with an Intel Core i3 (2.30GHz, 4GB RAM), and a graphic card NVIDIA GeForce 610M (900MHz, 2GB $VRAM$). We observe that the classification of our approach is as fast as the 2D approach (same technique, [5]), but also faster than the 3D approach.



(a) Left image. (b) Distribution of 500 samples on the left image. (c) Distribution of 1k samples on the left image.

(d) POI region. (e) Resulting 3D sampling pattern (500 samples). (f) Resulting 3D sampling pattern (1k samples).

Figure 9. *Distribution obtained with our hybrid method on* PIPE.

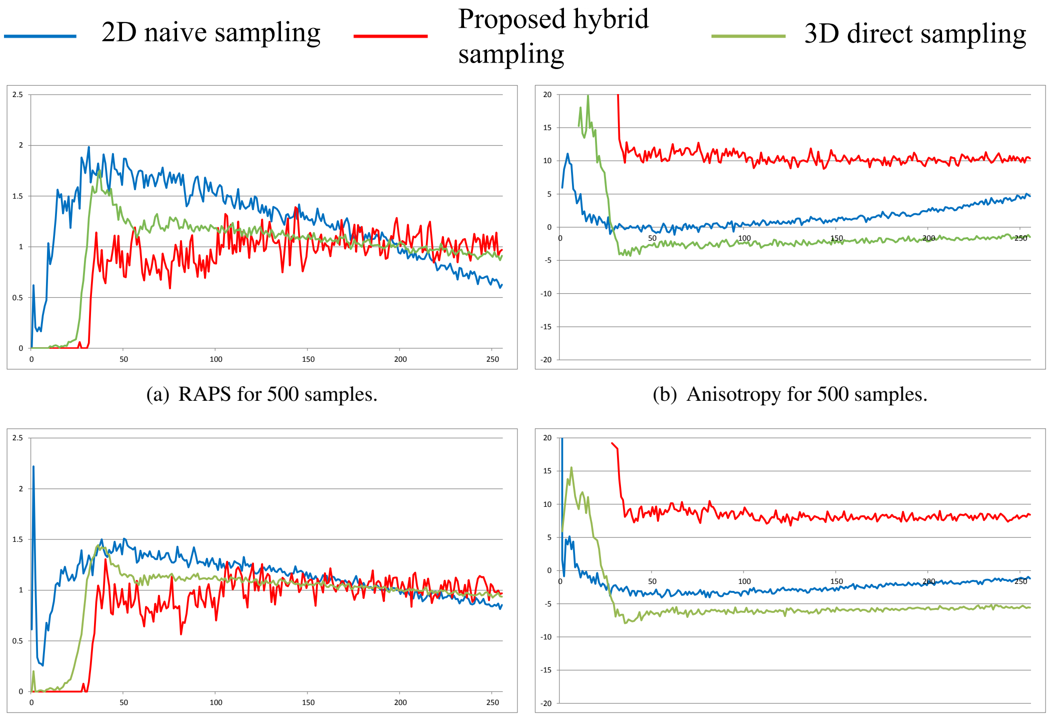


(a) Left image. (b) Distribution of 500 samples on the left image. (c) Distribution of 1k samples on the left image.

(d) POI region. (e) Resulting 3D sampling pattern (500 samples). (f) Resulting 3D sampling pattern (1k samples).

Figure 10. *Distribution obtained with our hybrid method on* BOX.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037



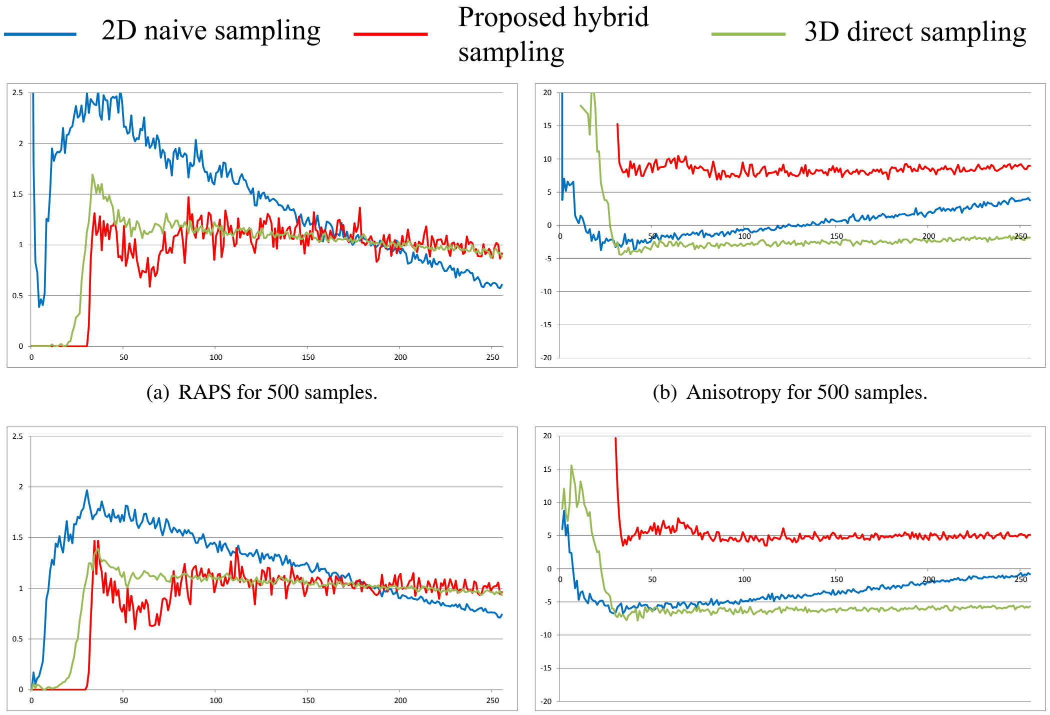
(a) RAPS for 500 samples.

(b) Anisotropy for 500 samples.

(c) RAPS for 1k samples.

(d) Anisotropy for 1k samples.

Figure 11. *Quality of the sampling distribution generated with a naive 2D method (blue), with our hybrid method (red), and with the 3D direct method of [4] for* PIPE. *RAPS and anisotropy (in dB) are given for two densities:* 500 *and* 1*k samples.*



(a) RAPS for 500 samples.

(b) Anisotropy for 500 samples.

(c) RAPS for 1k samples.

(d) Anisotropy for 1k samples.

Figure 12. *Quality of the sampling distribution generated with a naive 2D method (blue), with our hybrid method (red), and with the 3D direct method of [4] for* WALL. *RAPS and anisotropy (in dB) are given for two densities:* 500 *and* 1*k samples.*

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037

Concerning the sampling, the runtime of our method is lower than the 3D method for the densities of 500 and $1k$ samples, but not for a higher density ($5k$). Nevertheless, when we consider the global pipeline (from the acquisition to the sampling pattern), our method is less complex than the 3D method as we do not have to create the full point cloud and to triangulate it before applying the 3D method that needs a mesh as input.

Table 1: Running time (in seconds) for performing classification and sampling with different methods.

Model	Density	Method	Classification	Sampling
Pipe	500	Naive 2D	0.4496	0.53
		Hybrid	0.4177	2.309
		Direct 3D	0.748	3.931
	1k	Naive 2D	0.4336	0.577
		Hybrid	0.4174	3.869
		Direct 3D	0.733	6.146
	5k	Naive 2D	0.4028	0.967
		Hybrid	0,4334	19.610
		Direct 3D	0.7334	8,143
Box	500	Naive 2D	0.7221	0.966
		Hybrid	0.7393	4.415
		Direct 3D	0.905	5.428
	1k	Naive 2D	0.6909	1.046
		Hybrid	0.7379	7.722
		Direct 3D	0.920	7.270
	5k	Naive 2D	0.7233	1.825
		Hybrid	0.7380	35.864
		Direct 3D	0.905	12.231

Conclusion

We presented in this paper a novel approach to sample surfaces acquired with stereoscopic systems. The novelty is to sample the surfaces directly from the stereoscopic images, instead of making it on the reconstructed surfaces at the output of the acquisition systems. To the best of our knowledge, no prior work proposes such an approach, despite the advantage of controlling the number of sampling points at the beginning of the sampling/reconstruction process, to avoid oversampled data at the end. We developed a *hybrid* method: the sampling is performed in a 2D domain (as parameterization-based techniques) whereas geodesic distances between samples are computed in 3D space. Experimental results prove that our hybrid scheme produces sampling patterns with blue noise properties, comparable to those generated by 3D direct methods, and that the sharp features are well preserved. Moreover, when considering the whole pipeline from stereoscopic images to the final distributions, our method is simple, and faster than 3D direct methods that take as input a surface mesh, generated by triangulation of the point cloud, itself given by acquisition systems.

References

[1] P. Cignoni, C. Montani, and R. Scopigno, “A comparison of mesh simplification algorithms,” *Computers and Graphics*, vol. 22, no. 1, pp. 37 – 54, 1998.

[2] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene, “Recent advances in remeshing of surfaces,” in *Shape Analysis and Structuring, Mathematics and Visualization*. 2008, Springer.

[3] F. Payan, C. Roudet, and B. Sauvage, “Semi-regular triangle remeshing: a comprehensive study,” *ACM Transactions on Graphics*, 2014.

[4] Jean-Luc Peyrot, Frédéric Payan, and Marc Antonini, “Direct blue noise resampling of meshes of arbitrary topology,” *The Visual Computer*, vol. 31, no. 10, pp. 1365–1381, 2015.

[5] Min Ki Park, Seung Joo Lee, and Kwan H. Lee, “Multiscale tensor voting for feature extraction from unstructured point clouds.,” *Graphical Models*, vol. 74, no. 4, pp. 197–208, 2012.

[6] Daniel Heck, Thomas Schlömer, and Oliver Deussen, “Blue noise sampling with controlled aliasing,” *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 25:1–25:12, June 2013.

[7] Franklin C. Crow, “The aliasing problem in computer-generated shaded images,” *Commun. ACM*, vol. 20, no. 11, pp. 799–805, Nov. 1977.

[8] Yan Fu and Bingfeng Zhou, “Direct sampling on surfaces for high quality remeshing,” in *Proceedings of the ACM Symposium on Solid and Physical Modeling*, New York, NY, USA, 2008, p. 115–124, ACM.

[9] Matt Pharr and Greg Humphreys, *Physically Based Rendering: From Theory to Implementation*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[10] Yin Xu, Ruizhen Hu, Craig Gotsman, and Ligang Liu, “Blue noise sampling of surfaces,” *Computer and Graphics*, vol. 36, pp. 232–240, 2012.

[11] David Cline, Stefan Jeschke, Anshuman Razdan, Kenric White, and Peter Wonka, “Dart throwing on surfaces,” *Computer Graphics Forum*, vol. 28, no. 4, pp. 1217–1226, 6 2009.

[12] Hongwei Li, Li-Yi Wei, Pedro V. Sander, and Chi-Wing Fu, “Anisotropic blue noise sampling,” in *ACM SIGGRAPH ASIA papers*. 2010, p. 167:1–167:12, ACM.

[13] Daniel Scharstein and Richard Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.

[14] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[15] Vincent Daval, Frédéric Truchetet, and Olivier Aubreton, “Primitives extraction based on structured-light images,” in *QCAV’13*, Fukuoka, Japon, June 2013, p. 45.

[16] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.

[17] Edsger. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-037

Author Biography

Frédéric Payan received the PhD degree in signal and image processing in December 2004 from the University of Nice - Sophia Antipolis (UNS, France). He is now assistant professor at the UNS, in the I3S/CNRS Laboratory. His research interests include geometry processing, in particular compression, remeshing, and sampling of surfaces.

Jean-Luc Peyrot studied computer sciences, signal and image processing at the engineering school ESCPE Lyon, and graduated in 2011. In December 2014, he defended his PhD in signal and image processing under supervision of Dr. Marc Antonini and Dr. Frédéric Payan. In January 2015, he began a postdoctoral stage in the field of geosciences data compression at the IFPEN (France).

Marc Antonini received the Ph.D degree in electrical engineering from the University of Nice-Sophia Antipolis (France) in 1991. He joined the CNRS in 1993 at the I3S laboratory where he is "Directeur de Recherche" since 2004. His research interests include image, video and geometry coding using wavelet analysis. He is the author of more than 200 papers and 10 patents. He is a co-founder of Cintoo3D, a Start-Up specialized in 3D streaming solutions.