Published in Image Processing On Line on 2018–01–02.
Submitted on 2016–08–17, accepted on 2017–12–18.
ISSN 2105–1232 © 2018 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2018.187

The Production of Ground Truths for Evaluating High Accurate Stereovision Algorithms

Tristan Dagobert

CMLA, ENS Cachan, France
tristan.dagobert@cmla.ens-cachan.fr

Abstract

The conception and improvement of algorithms for subpixel stereovision requires very precise test databases. The state of the art on the sets of images used extensively by the scientific community shows that they are often incomplete and imprecise compared to the dataset goals. We will present a method based on image synthesis to produce stereoscopic pairs with ground truths such as disparity and occlusion maps reaching an accuracy of about 10^{-6} pixels. The a priori noise estimate is also taken into account. This process allows us to deliver a new image database consisting of 66 stereo pairs together with their ground truths.

Keywords: ground truths; disparity map; stereovision; synthetic images

Source Code

We provide the code of a program that computes the 3D coordinates of the points associated with each ground truth in order to view them with appropriate software. We also provide a modified code of the algorithm from Lisani et al. [15]. It allows to process images in floating point format such as TIF or EXR.

Supplementary Material

The database is available at this address[1]. The use of these images for scientists is permitted provided that this article is mentioned as well as the designers of the scenes.

[1] https://doi.org/10.5201/ipol.2018.187

1 Introduction

The field of stereo vision is vast and its applications have developed considerably over the last twenty years: satellite photogrammetry or robotic navigation are the best examples. In its most general formulation (see Hartley et al. [10, parts 2 to 4]), stereo reconstruction involves rebuilding a 3D model of the scene from two or more 2D views. This technology can be divided into two groups of complementary algorithms (cf. Szeliski [24, p. 19]): the algorithms estimating the position of the geometric points from the pixels, resulting in point clouds, and algorithms reconstructing the forms from such clouds. One of the basic needs for the development, improvement and objective comparison of such algorithms is to have reliable ground truths which are as accurate as possible; Scharstein et al. [21] were among the first to make such calibrated ground truths.

The matching algorithms are faced with two recurring problems which new approaches seek to overcome: the influence of noise in stereo pairs and the fattening effect. Noise due to the sensor technology is inherent to the images. However images used as ground truths must contain a noise which is negligible or at least quantified, to compare algorithm efficiency objectively. The second problem analyzed by Delon et al. [4, 5] is the fattening effect that appears along contrasted edges of the image as a dilation of the 3D model along the upper or lower part of the edges. To precisely measure the gap between the contours induced by this fattening and the exact edges, it is necessary to have a subpixel knowledge of the position of the edges. For the stereo matching algorithms computing a disparity map from a pair of images (see Szeliski [24, Chapter 11]), the following information is added to the ground truths:

- the transformation matrices between the view and the 3D frame of the scene to reconstruct the epipolar geometry (see Hartley et al. [10, Chapter 6], Szeliski [24, Chapter 2]);
- the disparity map which, for each pixel of an image I_1, indicates its position in the image I_2;
- the occlusion map which, for each pixel of I_1, indicates if it is visible or not in I_2.

State of the art datasets for stereo algorithms are sometimes incomplete. This is why we present in this paper an approach to ground truths creation filling some of the gaps. To this aim, we use synthetic scenes produced by a renderer where the geometry of objects in the scene, the optical characteristics of cameras and the variety of scenes are controllable. We tried to make the most of the geometric information obtained by ray tracing by exploiting the ray tracing spatial oversampling.

This article is organized as follows. Section 2 is devoted to an analysis of the state of the art of databases widely used by the scientific community for stereo algorithms evaluation. Having motivated our approach in Section 3 and briefly described the principle of ray tracing in Section 5 we detail in Section 6 the noise estimation method in image pairs, in Sections 7, 8 and 9 the creation of point clouds, disparity maps and occlusion maps, respectively. Finally, Appendix B describes the features of the image database files.

2 State of the Art

A large number of databases on computer vision are accessible to the scientific community. Riemenschneider [19] drew up an almost exhaustive list. We shall limit our study to databases relating primarily to image matching algorithms and 3D reconstruction algorithms. Regarding matching, two types of ground truths databases were devised over the recent years. The first are derived from images acquired in a real environment while the latter are produced from synthetic scenes.

2.1 Stereo Ground Truths in Real Environment

The Middlebury Dataset. Scharstein et al. [21] of Middlebury College were among the first to generate stereo pairs accompanied by ground truth. They have published five datasets with disparity maps over the last 15 years. From 2003 Scharstein et al. [22] produced pairs whose disparity estimation is obtained by illuminating the scene with a coherent light. Specifically the scene was illuminated several times by projecting different bar patterns. As a result each pixel of the stereo pairs was marked with a unique multi-spectral signature. Measurement of disparities using an ad hoc algorithm was then much easier. The authors did not, however, describe this registration algorithm.

Until 2013 the disparity maps are accurate up to one pixel. By improving the illumination device, including the projection of colored bar patterns defined by the method of Gupta et al. [9], and post-processing of the acquired images, Scharstein et al. [20] have provided a set of 33 scenes with subpixel disparities, some of which achieve 0.2 pixel accuracy. This method, however, due to the complexity of its implementation, encounters a number of problems [22, Section 4.1] such as:

- some pixels have partial occlusion;
- some pixels have no signature because of shadow and reflection effects;
- the presence of aliasing or blurring in signatures;
- some signatures are inconsistent because of the illumination changes;

so that it takes at least twenty steps to process the raw data [20, Figure 3] to obtain the disparity maps.

The KITTI Vision Benchmark Suite. Geiger et al. [7] proposed an image database acquired from a vehicle with different sensors namely two high resolution cameras, a laser scanner and a GPS location system. This database serves as a tool for benchmarking and ranking matching algorithms. They provide a training set of nearly 200 stereo pairs with ground truths composed of disparity, occlusion and optical flow maps.

However, these ground truths were obtained from the scans of a rotating 3D laser scanner, so that the laser sampling does not correspond to the pixel sampling of the image. Nearly half of the pixels of the image have no ground truth, and it must be deduced by interpolation. In addition, disparities and optical flow maps have integer values.

Image Sequence Analysis Test Site (EISATS). Reinhard Klette et al. [13] proposed ten stereo sequences acquired from cameras and a laser scanner mounted on vehicles. Most of these scenes are grayscale and made in a real environment. Among them, the sequence 1 has car kinematics ground truths, sequence 2, in synthetic images, has temporal optical flow between frames, and sequence 6 has disparity and depth maps deduced from the scan, but relatively noisy and incomplete.

HCI Robust Vision. The Heidelberg Collaboratory for Image Processing project [14, 17] proposed a set of stereo sequences in road urban environment. The dataset does not contain ground truths because it is primarily a project for the final evaluation of algorithms.

2.2 Stereo Ground Truths in a Virtual Environment

University of Tsukuba Stereo Flow. Martull et al. [16] produced a 1800 stereo pairs dataset of 640×480 photo-realistic images with ground truths including: disparity, occlusion and discontinuity maps as well as the position and orientation of the cameras. The pairs were extracted from a 3D

synthetic scene representing an office created with the Autodesk Maya 2012 software then textured using real and synthetic textures. These maps are pixel-accurate. It is not possible to use such a database for a tenth-pixel or even a quarter-pixel benchmark because this operation would imply reducing the images' size, which is already small.

MPI Sintel Flow. The MPI Sintel database [28, 2] is a set of sequences and images picked from an animated film containing varied and realistic environments. Its features are: long movements, non-rigid moving objects, specular reflections, camera shake and other atmospheric effects. It is mainly dedicated to the optical flow evaluation: objects and characters are moving from one image to another. The database consists of 35 excerpts split into a training set (23 sequences) and an evaluation set (12 sequences). Optical flow, (poorly) estimated edges, occlusion maps and rendering effects are indicated there. This database cannot be used for the evaluation of stereo algorithms which assume rigid deformations between images.

2.3 Ground Truths Dedicated to 3D Reconstruction.

The following bases cannot be used for stereo matching in itself, but for the next step which, is to build surfaces from point clouds estimated by matching.

Middlebury College. Middlebury College [23] proposes two scanned objects (a Roman temple and a dinosaur) through 395 different points of view but whose acquisition was not coupled with a camera. The laser scanner used was moved to cover a hemisphere. However, only 80% of the hemisphere of the Roman Temple object is exploitable.

Stanford University. The Stanford 3D Scanning Repository database [3] is a set of a dozen 3D scans of objects which contains the coordinates of 3D points and the triangulations of the mesh.

University of Utah. Berger et al. [1] studied the problem of surfaces reconstruction. To this end they simulated the acquisition of data from a laser scanner to reproduce realistic point clouds. They propose a set of 5 items scanned synthetically and for each of these, 48 point clouds. However, the views of these clouds are always the same, only the sampling changes.

Institut Farman. Digne et al. [6] have produced a 3D points dataset composed of nearly 200 scenes of items that have been both scanned by a 3D high precision scanner laser and photographed by a CCD camera. Each of the 11 items presented was scanned under 18 views. The high-precision images are accompanied by 3D point coordinate files.

3 Our Approach

As we can see, databases acquired in a real environment suffer from a lack of accurate information. This is due to difficulties in handling and synchronizing acquisition devices. On the contrary, images obtained by synthesis like for Sintel or Tsukuba provide accurate and complete sets. These however do not reach subpixel precision regarding the edges and they are more oriented towards optical flow estimation. Nevertheless the use of synthetic scenes takes on its full meaning because it can precisely control many parameters and work on more realistic scenarios.

Our approach is to use images produced by the renderer and also to exploit the data generated during rendering. When creating an image, the renderer generates, for each pixel p, a number $N(p)$ of

rays that will intersect the objects in the scene at 3D points $\mathbf{P}(p,n)$ with depth r_n for $n = 1,\ldots,N(p)$. The color characteristics of these points are then averaged to determine the final color of p. We will give more details in Section 5 on the calculation of these colors and the underlying mathematical model. These contributions thus form for each pixel a cloud of 3D points and therefore, in their entirety, an oversampling of the objects in the scene much more important than the oversampling of the image itself. This set of data is the starting point to create precisely all ground truths that we need and that are the subject of the next sections. Table 1 summarizes the main properties of the bases previously mentioned as well as ours, entitled "CMLA dataset".

Table 1: Comparison of ground truths present in the bases described above, including ours (CMLA) for the evaluation of stereovision algorithms.

	Middlebury	KITTI	EISATS	HCI	Tsukuba	Sintel	Inst. Farman	CMLA
images resolution	8 bits	8 bits	8 bits	12 bits	8 bits	8 bits	8 bits	floating 16 bits
rigid deformations	Y	Y	N	N	Y	N	Y	Y
noise estimate	N	N	N	N	N	N	N	Y
noise estimate of 3D points position	N	N	N	N	N	N	N	Y[1]
optical flow map	N	Y	Y	N	N	Y		
depth map	N	Y	Y	N	N	N	N	Y
disparity map precision	$\simeq 0.2$ pix.	1 pix.	$\ll 1$ pix.	1 pix.	1 pix.		N	$\simeq 10^{-6}$ pix.[1]
occlusion map	partially	Y	N	N	Y	Y	N	Y

The creation of synthesized images and ground truths led us to choose the rendering engine LuxRender [27]. This software presents many advantages. It is a clone of the software PBRT developed by Pharr et al. [27] whose design is very detailed and whose interest is to rely on physical characteristics of materials such as metals and photometric features of the light sources so that the rendering is very realistic. It also provides natively floating 16-bit EXR or 16-bit PNG image formats as well as a depth map. Finally, it may be coupled to the 3D builder software Blender[2].

4 Notations

Table 2 presents the list of notations used in this article. The points may be indexed A, I or C depending on whether they are considered into the absolute $\mathfrak{R}_A$, the camera $\mathfrak{R}_C$ or the $\mathfrak{R}_I$ image frame (see Figure 1). By convention, the parallel planes $(O_I,\vec{x}_I,\vec{y}_I)$ and $(O_C,\vec{x}_C,\vec{y}_C)$ are oriented in the opposite direction (i.e. $\vec{x}_I = -\vec{x}_C$ and $\vec{y}_I = -\vec{y}_C$).

[1]Precision is only limited by the numbers representation in single-precision floating-point format.
[2]Blender - a 3D modelling and rendering package [11], Blender Foundation, http://www.blender.org.

Table 2: Main notation.

$\|.\|_2$	ℓ^2 norm
$\propto$	proportional to
$\mathfrak{R}_A = (O_A, \vec{x}_A, \vec{y}_A, \vec{z}_A)$	absolute and direct frame, of origin O_A, corresponding to a 3D scene (see Figure 1)
$\mathfrak{R}_C = (O_C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$	direct frame related to the camera defined in the synthetic 3D scene, whose origin O_C is its focal point
$\mathfrak{R}_I = (O_I, \vec{x}_I, \vec{y}_I)$	direct frame related to image Ω whose origin O_I is the image upper left corner
$\mathbf{P} = (x, y, z)^T$	3D geometric point of the synthetic 3D scene
$\mathbf{p} = (x, y)^T$	point belonging to an image
Ω	the digital image considered as a matrix of pixels (in the electronic sense), that is to say the rectangular surface of width $L_I = L_I \Delta l$ and height $H_I = H_I \Delta l$ with $\Delta l = 1$ by convention
$I = [\![0, \ldots, L_I - 1]\!] \times [\![0, \ldots, H_I - 1]\!]$	the discretized digital image regarded as the matrix of the sampling points of Ω
L_I	number of columns of the image
H_I	number of rows of the image
$N_I = L_I \times H_I$	image size
i	integer index along $\vec{x}_I$ axis
j	integer index along $\vec{y}_I$ axis
u	the ideal digital image considered in terms of RGB colors defined on I with values in $\mathbb{R}^3_+$
$\tilde{u}$	u estimate obtained by ray tracing
$\Omega(i,j) = [(i,j), (i,j+1), (i+1,j+1), (i+1,j)]$	the square surface, strictly speaking, the pixel
$p = (i, j)$	the pair of integer indices, appointed pixel for convenience associated with the point $\mathbf{p} = (i,j)^T$ of image I and to surface $\Omega(i,j)$ of image Ω
$N(p)$	the number of contributions associated with the pixel p
$\mathbf{c}_n = (x_n, y_n)^T$	position, expressed in the reference frame $\mathfrak{R}_I$, of the nth contribution of p
r_n	depth of the nth contribution of p
$\upsilon_n = (\upsilon_n^R, \upsilon_n^G, \upsilon_n^B)$	color of the nth contribution of p
$\mathcal{V}(p)$	neighborhood of p corresponding to the smallest square containing contributions and center $(i + 0.5, j + 0.5)^T$

5 Principle of Ray Tracing

In its general representation, a 3D scene is described by a set of meshes to which are associated photometric properties, one or more light sources, one or more cameras characterized by their optical properties, and optionally air or dynamic characteristics. The photorealistic rendering consists in calculating within each pixel a color as accurate and realistic as possible on the basis of the above parameters. Rendering is an attempt to solve the light transport equation as it was formulated by Kajiya [12]. Veach [25, Chapter 8] demonstrated that this equation could be reformulated by the

integration problem

$$u(x,y) = \int_{\omega} f(\gamma, x, y) d\mu(\gamma), \tag{1}$$

where u is the image, (x, y) the position expressed in image frame $\mathfrak{R}_I$, ω is the set of paths of all possible lengths carrying the light, μ is a measure on ω and f is the function of light contribution. The function f depends on photometric parameters (scattering, absorption and reflection spectra) associated with the objects encountered on path γ. With this formulation, color $u(x, y)$ can be approximated by the equation

$$\frac{1}{N(x,y)} \sum_{n=1}^{N(x,y)} \upsilon_n(x_n, y_n), \tag{2}$$

where $\upsilon_n(x_n, y_n) = f(\gamma_n)$ is the color obtained by (x_n, y_n) the starting point of path γ_n and $N(x, y)$ is the number of rays. This quantity is computed by an iterative method of the type of the Monte-Carlo integration.

More precisely (see Figure 1) from each pixel p, a number $N(p)$ of rays are sent that intersect the objects in the scene at 3D points $\mathbf{P}(p, n)$ for $n = 1, \ldots, N(p)$. These rays originate in the focal plane at positions $\mathbf{c}_n = (x_n, y_n)$ located in a neighborhood $\mathcal{V}(p)$ related to pixel p and pass through the focal point O_C of the camera. We define the neighborhood $\mathcal{V}(p)$ as the smallest square containing all the points $\mathbf{c}_n$ and centered at $(i + 0.5, j + 0.5)^T$ where (i, j) are the integer coordinates of the upper left pixel corner. Every contribution $\mathbf{c}_n$ is associated to a color υ_n derived from the photometric parameters of the scene and to a depth r_n defined as the distance $\|\mathbf{P}(p, n) - O_C\|_2$.

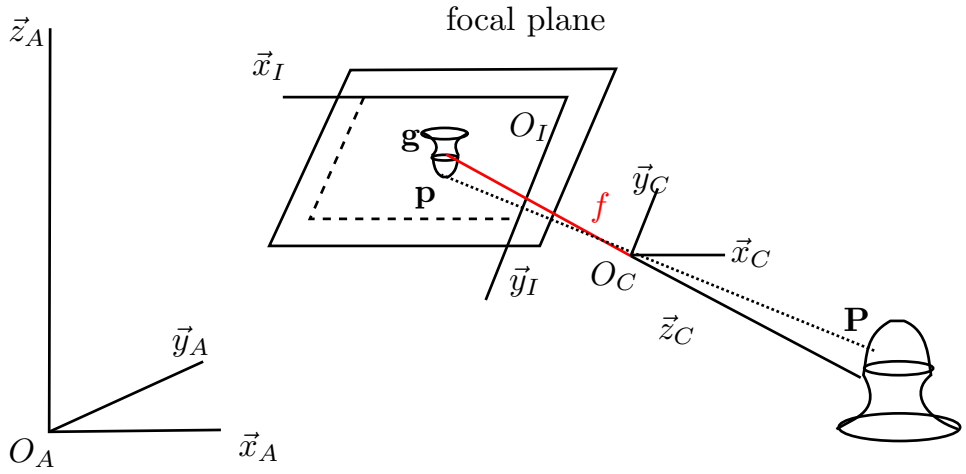


Figure 1: Schematic diagram of the formation of an image by a pinhole camera. $\mathfrak{R}_A = (O_A, \vec{x}_A, \vec{y}_A, \vec{z}_A)$ is the absolute coordinate frame of the 3D scene, the direct frame $\mathfrak{R}_C = (O_C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$ is linked to the camera and $\mathfrak{R}_I = (O_I, \vec{x}_I, \vec{y}_I)$ to the image. The red line, perpendicular to both the focal plane $(O_I, \vec{x}_I, \vec{y}_I)$ and plane $(O_I, \vec{x}_C, \vec{y}_C)$, is the focal length f connecting focal point O_C with the image center $\mathbf{g}$. The camera's line of sight is $(O_C, \vec{z}_C)$. The 3D geometric point $\mathbf{P}$ is projected on the image point $\mathbf{p}$.

The integration process is iterative. At each iteration k, (i.e. each pass k) a non accumulated image u_k is computed and averaged with the preceding images u_{k-l} for $l = 1, \ldots, k$. We have for all p, $N(p) = \sum_{k=1}^{K} N^k(p)$ where $N^k(p)$ is the number of rays shot at k. Note that, in practice, the renderers sometimes use anti-aliasing, motion blur or defocus filters which involve neighboring pixels of p, induce a change in the final color $\tilde{u}(p)$ and adapt the effective size of neighborhood $\mathcal{V}(p)$. We consider two types of distributions within the pixel: a pseudo-random distribution of the contributions (see Figure 2-left) which is used to render the color image, and a distribution on a regular grid (Figure 2-right) which is used to produce disparity and occlusion maps. In the first case,

the law defining the distribution might be specified as a scene parameter (e.g. PBRT [18, Chapter 7]). In the second case, N is an integer square fixed for all p which implies that the contribution positions into the pixel $p = (i, j)^T$ are given by

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} i \\ j \end{pmatrix} + \frac{1}{\sqrt{N}} \begin{pmatrix} \lambda + 0.5 \\ \mu + 0.5 \end{pmatrix}, \forall (\lambda, \mu) \in \{0, \ldots, \sqrt{N} - 1\}^2. \tag{3}$$

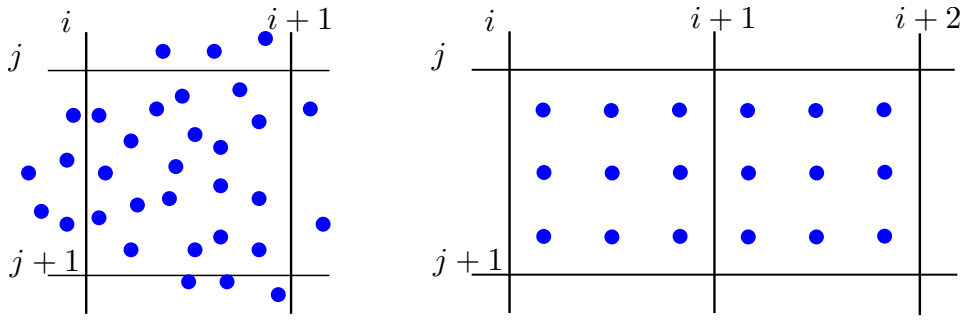


Figure 2: Left: an example of a pixel p such that $N = 30$, where the contributions $\mathbf{c}_n$ are distributed according to a random process simulated by the renderer. This kind of distribution is used for the creation of the high resolution image. Right: an example of a pixel whose contributions are evenly distributed when $N = 9$. This regular distribution is used for the creation of ground truths as disparity maps, occlusion maps and points clouds.

6 Noise Estimation

We present in this section several noise estimators applied to the raw images or to the images requantized between 0 and 255. We consider the general case, that is to say when the distribution of contributions is random. If the number of passes of the rendering is insufficient, the rendering noise is important, especially in dark areas (see Figure 3). However, it is possible to quantify the average noise of the pixels and to deduce the computation time required to obtain an image with a given average noise variance. Indeed, unbiased renderers are designed so that at the end of the last pass each pixel p has received approximately the same number of $N(p)$ contributions. We see the image as a random vector and define the average noise variance as follows.

Definition 1. *The average noise variance for image* $u = (\mathcal{U}^1, \ldots, \mathcal{U}^{3N_I})$ *where* $\mathcal{U}^n$ *are the random variables associated with the channels pixels is*

$$var(u) = \frac{1}{3N_I} \sum_{n=1}^{3N_I} var(\mathcal{U}^n). \tag{4}$$

Let $v_k = (\mathcal{X}_k^1, \ldots, \mathcal{X}_k^{3N_I})$ be the random vector representing the non accumulated image at the k^{th} iteration. It is assumed that the renderer has the following statistical properties:

- the iterations are homoscedastic (all random variables in the sequence have the same finite variance)
$$\forall n, \forall k, \forall l \neq k, \mathrm{var}(\mathcal{X}_k^n) = \mathrm{var}(\mathcal{X}_l^n), \tag{5}$$
- the iterations are time-independent
$$\forall n, \forall k, \forall l \neq k, \mathrm{cov}(\mathcal{X}_k^n, \mathcal{X}_l^n) = 0, \tag{6}$$

- the iterations are spatially independent

$$\forall k, \forall n, \forall m \neq n, \mathrm{cov}(\mathcal{X}_k^n, \mathcal{X}_k^m) = 0. \tag{7}$$

During rendering, images v_k are filtered by an anti-aliasing filter modeled by a convolution with a normalized kernel h leading to image u_k, namely

$$u_k = v_k * h, \tag{8}$$

and denoted by the same convention $u_k = (\mathcal{Y}_k^1, \ldots, \mathcal{Y}_k^{3N_I})$. We assume that rendering is linear and unbiased, that is to say that the image $\bar{u}_k$ obtained after k passes as the output of ray tracing, is equal to the average of k images that would have been calculated independently during a single pass by k ray tracings. This leads to

$$\bar{u}_k = \frac{u_1 + \cdots + u_k}{k}. \tag{9}$$

By the variance additivity of temporally independent random variables we deduce that each pixel n of $\bar{u}_k = (\bar{\mathcal{Y}}_k^1, \ldots, \bar{\mathcal{Y}}_k^{3N_I})$ is subjected to a noise variance

$$\mathrm{var}(\bar{\mathcal{Y}}_k^n) = \frac{\mathrm{var}(\mathcal{Y}_k^n)}{k}, \tag{10}$$

and at infinite time, by the law of large numbers, $\bar{u}_k$ converges towards the non noisy image u. The noiseless image being inaccessible over time, one can nevertheless estimate the noise variance of the final image $\bar{u}_K$ from the mean squared error (MSE) compared to an intermediate image $\bar{u}_k$ under the following proposition whose proof is given in Appendix A.1:

Proposition 1. *Let $\bar{u}_k$ and $\bar{u}_K$ be two images rendered respectively after k and K iterations, $k < K$. The estimated average variance of noise for the rendered image $\bar{u}_K$ corresponds to*

$$var(\bar{u}_K) \simeq \frac{k}{K-k} MSE(\bar{u}_k, \bar{u}_K), \tag{11}$$

where the mean square error (MSE) is defined by

$$MSE(\bar{u}_k, \bar{u}_K) = \frac{1}{3N_I} \mathbb{E}\|\bar{u}_k - \bar{u}_K\|_2^2. \tag{12}$$

Since the obtained raw image $\bar{u}_K$ is expressed directly from the spectral properties of materials, the values of its pixels are not limited to $[0, 255]$ but belong to $[0, +\infty)$ so that the value of the empirical variance (11) is not very intuitive. We therefore propose a normalized representation of the variance with respect to the $[0, 255]$ range based on the coefficient of variation [26, p. 22]:

Definition 2. *The normalized variance with respect to the range $[0, 255]$, of image $\bar{u}_K$ is defined by*

$$\varsigma^2(\bar{u}_K) = \frac{127.5 \cdot var(\bar{u}_K)}{\mu(\bar{u}_K)}, \tag{13}$$

where $\mu(\bar{u}_K)$ is the mean of $\bar{u}_K$.

To provide stereo algorithms with images whose dynamics are contained in the range $[0, 255]$ while avoiding saturation, we applied to the raw image $\bar{u}_K$ the histogram equalization algorithm by Lisani et al. [15] whose interest is to use only three input parameters; we note by f_L this transformation. We propose the following relation for the noise estimate for the new $f_L(\bar{u}_K)$ image (see the proof in Appendix A.2):

Proposition 2. *The variance of image $\bar{u}_K$ requantized by Lisani et al.'s transformation f_L corresponds to*

$$var(f_L(\bar{u}_K)) = \frac{k}{3N_I(K-k)} \sum_{n=1}^{3N_I} \alpha_n^2 (\bar{u}_k(n) - \bar{u}_K(n))^2, \tag{14}$$

where α_n is the slope of f_L (denoted as m_k in [15, Section 2]).

Finally, we define the signal to noise ratio of the images $\bar{u}_K$ and $f_L(\bar{u}_K)$ from the definition by Gonzalez et al. [8, Equation 5.8-5], replacing the estimated variance of denominator of Equation 5.8-5 by the estimated variance (11):

Definition 3. *The signal to noise ratio (SNR) of a given image v is defined as*

$$SNR(v) = \frac{\|v\|_2}{\sqrt{3N_I var(v)}}. \tag{15}$$

Figure 3: Example of residual noise in a shaded portion of the scene made for different durations. From left to right, the images were generated respectively for the periods $T = 1$, $T = 4$ and $T = 16$. Between two successive images the noise standard deviation is divided by a factor of 2.

7 Generation of the 3D Point Clouds

We describe in the following paragraph the geometric relationship between the point of the 3D scene $\mathbf{P}$ and its projection $\mathbf{p}$ on the image plane according to the pinhole camera model [25, Chapter 2]. The relationship between these two points is given by two transformation matrices whose dimensions are 4×4 by convention. The matrix $\mathbf{R}_{AC}$ of inverse $\mathbf{R}_{CA}$ is called the rigid displacement matrix and is used to express the frame $\mathfrak{R}_A$ compared to the camera frame $\mathfrak{R}_C$. Its coefficients depend on three rotations of the axes of the camera in the frame $\mathfrak{R}_A$, represented by the matrix $\mathbf{R}_{CA}$ and the translation $\mathbf{T}_{CA} = (t_x, t_y, t_z)^T$ of focal point O_C relatively to O_A. Its homogeneous formulation is

$$\mathbf{R}_{AC} = \left(\begin{array}{c|c} \mathbf{R}_{CA} & -\mathbf{T}_{CA} \\ \hline \mathbf{0} & 1 \end{array}\right) \text{ and its inverse is } \mathbf{R}_{CA} = \left(\begin{array}{c|c} \mathbf{R}_{CA}^{-1} & 0 \\ \hline \mathbf{0} & 1 \end{array}\right) \left(\begin{array}{c|c} \mathbf{I} & +\mathbf{T}_{CA} \\ \hline \mathbf{0} & 1 \end{array}\right). \tag{16}$$

The second invertible matrix corresponds to the calibration matrix $\mathbf{R}_{CI}$ of the camera. This matrix is upper triangular, and in its simplest formulation [24, Equation 2.59], depends only on the focal

length f and the coordinates $\mathbf{g}_{|I} = (g_x, g_y)^T$ of the center of the image, when it originates from the top left corner O_I. Its homogeneous formulation is

$$\mathbf{R}_{CI} = \begin{pmatrix} f & 0 & g_x & 0 \\ 0 & f & g_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and its inverse is } \mathbf{R}_{IC} = \begin{pmatrix} 1/f & 0 & -g_x/f & 0 \\ 0 & 1/f & -g_y/f & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{17}$$

Point $\mathbf{P}_A = (x_A, y_A, z_A)^T$ is expressed in homogeneous coordinates as $\tilde{\mathbf{P}}_A = (x_A, y_A, z_A, 1)^T$. Similarly $\mathbf{p}_I = (x_I, y_I)^T$ is associated with $\tilde{\mathbf{p}}_I = (x_I, y_I, 1, \alpha)^T$. By convention (see [18, p. 75]) the point $\tilde{\mathbf{p}}_I$ has a weighted homogeneous coordinates representation denoted $\tilde{\mathbf{p}}_I^w = (w_I x_I, w_I y_I, w_I, w_I \alpha)^T$ where w_I and α are not zero (the term α does not play a direct role in the transformations below).

Calculating $\tilde{\mathbf{p}}_I$ knowing $\tilde{\mathbf{P}}_A$ is as follows. Denoting $\tilde{\mathbf{P}}_C = (x_C, y_C, z_C, 1)^T = \mathbf{R}_{AC}\tilde{\mathbf{P}}_A$ yields

$$\tilde{\mathbf{p}}_I \propto \tilde{\mathbf{p}}_I^w = \mathbf{R}_{CI}\mathbf{R}_{AC}\tilde{\mathbf{P}}_A, \tag{18}$$

$$= \mathbf{R}_{CI}\tilde{\mathbf{P}}_C. \tag{19}$$

Developing (19) from the definition of $\mathbf{R}_{CI}$ (17) allows us to see that $w_I = z_C$ and to deduce that

$$\tilde{\mathbf{p}}_I = \frac{1}{z_C}\mathbf{R}_{CI}\tilde{\mathbf{P}}_C. \tag{20}$$

To find the 3D geometric point $\mathbf{P}_A$ related to contribution $\mathbf{p}_I$, of which we know the position in the image and distance $r = \|\mathbf{P}_C\|_2$, one uses Thales' theorem which applies here to the pinhole cameras

$$z_C = \frac{rf}{\|f^2 + \langle \mathbf{p}_I - \mathbf{g}_I, \mathbf{p}_I - \mathbf{g}_I \rangle\|_2}. \tag{21}$$

By inverting Equation (20) then applying the transformation $\mathbf{R}_{CA}$ (16) we finally obtain

$$\tilde{\mathbf{P}}_A = \frac{rf}{\|f^2 + \langle \mathbf{p}_I - \mathbf{g}_I, \mathbf{p}_I - \mathbf{g}_I \rangle\|_2}\mathbf{R}_{CA}\mathbf{R}_{IC}\tilde{\mathbf{p}}_I. \tag{22}$$

The choice of the distribution of contributions to be taken into account a priori depends of the application for which point clouds are intended. The case of regular and single pixel distributions ($N = 1$) seems better suited to serve as ground truth for stereovision algorithms. Indeed, the cloud of 3D points represents the ideal cloud that we can hope to rebuild from a stereoscopic pair. The cases with regular or irregular $N \gg 1$ that result in an over-sampling of the volume of the scene seem more intended to mesh or 3D surface reconstruction algorithms tests.

Figure 4 (right) shows an example of a 3D reconstruction without artifacts from the image of the depths of contributions shot from the pixel center (left). Note that the calculation of the average depths of contributions of a pixel is to be avoided and is only of interest to provide noisy data. Indeed, it induces in the 3D reconstruction significant errors for the pixels on the boundaries representing close objects placed in front of a long shot because their 3D points have no physical reality (see Figure 4, center of the picture).

8 Construction of the Disparity Map

In the case of two points of views, the map of disparities D_1 on the image I_1 of the camera C_1 measures for each of its pixels p, the displacement $D_1(p) = (d_x(p), d_y(p))^T$ in image I_2 associated with camera

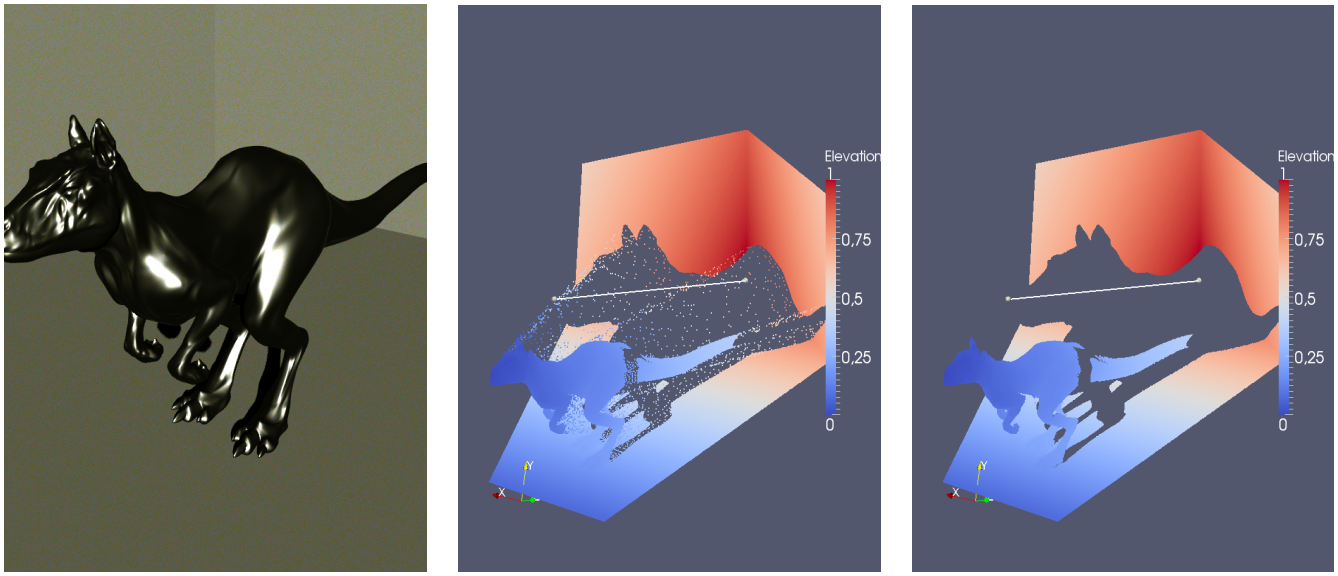


Figure 4: Example of 3D reconstruction of a scene from the image depths (left). Rebuilding from the average of the depths of a pixel produces artifacts (middle image). Using regular ray tracing in each pixel defined by Equation (3) eliminates these artifacts (right).

C_2 (see Figure 6). Frames associated with I_1, I_2, C_1 and C_2 are noted $\mathfrak{R}_a = (O_a, \vec{x}_a, \vec{y}_a, \vec{z}_a)$ for all $a \in \{I_1, I_2, C_1, C_2\}$.

The knowledge of matrices $\mathbf{R}_{AC_1}, \mathbf{R}_{C_1I_1}, \mathbf{R}_{AC_2}, \mathbf{R}_{C_2I_2}$ and of the distance $r = \|\mathbf{P}_{C_1}\|_2$ of the 3D point $\mathbf{P}$ associated to $\mathbf{p}_{I_1}$ allows to precisely calculate $D_1(p)$ by back-projecting $\mathbf{P}$ on I_2 (see Figure 5). Considering the ray shot from the center of $\Omega(p)$ that is to say $\mathbf{p}_{I_1} = (i+0.5, j+0.5)^T$ and denoting $\tilde{\mathbf{q}}_{I_2} = (x_{I_2}, y_{I_2}, w_{I_2}, \alpha_{I_2})^T$ the projection of $\mathbf{P}$ on I_2, we obtain from (18) and then from (22)

$$\tilde{\mathbf{q}}_{I_2} \propto \tilde{\mathbf{q}}^w_{I_2} = \mathbf{R}_{C_2I_2}\mathbf{R}_{AC_2}\tilde{\mathbf{P}}_A, \tag{23}$$

$$= \frac{rf}{\|f^2 + \langle \mathbf{p}_{I_1} - \mathbf{g}_{I_1}, \mathbf{p}_{I_1} - \mathbf{g}_{I_1} \rangle\|_2} \mathbf{R}_{C_2I_2}\mathbf{R}_{AC_2}\mathbf{R}_{C_1A}\mathbf{R}_{I_1C_1}\tilde{\mathbf{p}}_{I_1}, \tag{24}$$

$$\tilde{\mathbf{q}}_{I_2} = \frac{1}{w_{I_2}}\tilde{\mathbf{q}}^w_{I_2}. \tag{25}$$

This defines the disparity map calculated at the center of the pixel

$$D_1(i,j) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} (\tilde{\mathbf{p}}_{I_2} - \tilde{\mathbf{p}}_{I_1}), \forall i, \forall j. \tag{26}$$

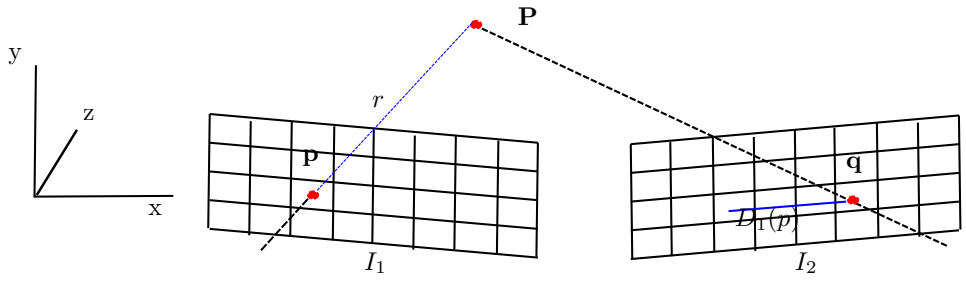


Figure 5: Computation of the disparity of $\mathbf{p}$ from the back-projection of $\mathbf{P}$ on I_2.

In the particular case where there is a fronto-parallel displacement of the camera, the map has no d_y component. This displacement corresponds to a translation in $\vec{x}_{C_1}$ of frame $\mathfrak{R}_{C_1}$ thus giving

frame $\mathfrak{R}_{C_2}$. In order to set a priori the disparity $d_x(p)$ of a pixel p whose associated distance r is known, one must calculate the O_{C_2} position. This change only affects the translation vector $\mathbf{T}_{C_2A}$ component of matrix $\mathbf{R}_{C_2A}$ since the rotation matrix $\mathbf{R}_{C_2A}$ remains unchanged. Under the Thales theorem applicable to pinhole cameras we have

$$\frac{x_{O_2}}{d_x(p)} = \frac{r}{\|f^2 + \langle \mathbf{p}_{I_1} - \mathbf{g}_{I_1}, \mathbf{p}_{I_1} - \mathbf{g}_{I_1} \rangle\|_2}, \tag{27}$$

where x_{O_2} is the abscissa of O_{C_2} in $\mathfrak{R}_{C_1}$, therefore

$$\tilde{O}_{C_2C_1} = \left(\frac{r d_x(p)}{\|f^2 + \langle \mathbf{p}_{I_1} - \mathbf{g}_{I_1}, \mathbf{p}_{I_1} - \mathbf{g}_{I_1} \rangle\|_2}, 0, 0, \alpha \right)^T, \tag{28}$$

for $\alpha \neq 0$. According to the definition of $\mathbf{T}_{C_2A}$, the relation

$$\mathbf{T}_{C_2A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{R}_{C_1A} \tilde{O}_{C_2C_1} \tag{29}$$

then allows to construct the appropriate matrix $\mathbf{R}_{C_2A}$ to shift $d_x(p)$.

In this case the stereo pairs were generated from a fronto-parallel movement. From a reference view, three shifted points of view were created so that the disparities d_x are of maximum value of 1, 10 and 50 pixels respectively. Note that the limitations of digital precision in matrix calculations induce the appearance of tiny displacements in y. Table 6 draws up the maximum amplitude of such displacement. Disparity maps in y were included in the data set although they may be neglected in a first approximation as they do not exceed $1.722 \cdot 10^{-4}$ pixels.

For practical purposes we indicate in the ground truth the focal f and the median baseline B/H of each stereo pair, which is calculated from the Thales relationship

$$\frac{x_{O_2}}{z_C(p)} = \frac{d_x(p)}{f}, \tag{30}$$

applicable to the pinhole case with fronto-parallel displacement and the following applies

$$\frac{B}{H} = \frac{\text{median}(|D_1|)}{f}. \tag{31}$$

left image I_1 right image I_2 disparity map D_1

Figure 6: Example of disparity map obtained with a stereo pair (I_1, I_2) for which the camera made a maximal fronto-parallel displacement of 50 pixels.

9 Construction of the Occlusion Map

The occlusion map O_1 shows for each pixel p of image I_1 if the area observed from this pixel has been occulted or not by another area in front of it, in the I_2 image. Knowing the disparity map D_2 of I_2 over I_1, the occlusion map O_1 of view I_1 is classically defined (see Figure 7) from the following boolean formulation

$$O_1(p) = \begin{cases} 1 \text{ if } \exists q / \lfloor D_2(q) + q \rfloor = p, \\ 0 \text{ if } p \text{ is occulted.} \end{cases} \tag{32}$$

However if one is limited to using only one contribution per pixel, i.e. $N = 1$, as in the case of the

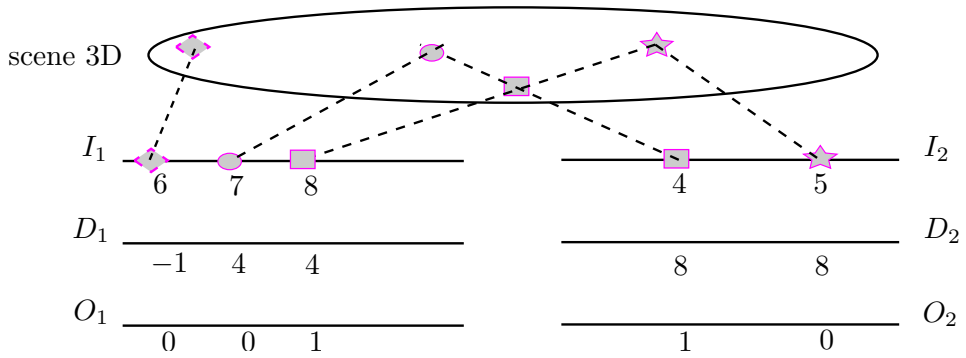


Figure 7: Creation of the occlusion map from the disparity map. The values shown in the line I_1 (resp. I_2) correspond to the scene positions of objects projected on image I_1 (resp. I_2), those below the line D_1 (resp. D_2) to their position in D_2 (resp. D_1). Boolean values of the image O_1 (resp. O_2) indicate whether the objects in I_1 (resp. I_2) are seen or not in I_2 (resp. I_1).

disparity, to compute the backward projection $\mathbf{P}$ on I_2, one experimentally observes (see Figure 8) a number of I_1 pixels that are not considered as visible as they should be. The reason is that in this case spatial sampling is not dense enough: two different pixels of I_1, $\mathbf{p}_1$ and $\mathbf{p}_2$, can have the same projected integer coordinates $\mathbf{q}$ on I_2. The solution to this problem is to use a spatial oversampling, i.e. $N \gg 1$ in back-projecting exhaustively all the contributions of each p of I_1 on I_2 then to accumulate. In this way the occlusion map is obtained

$$\tilde{O}_1(p) = \begin{cases} \sum\limits_q 1 \quad \forall q / \lfloor D_2(q) + q \rfloor = p, \\ 0 \text{ if } p \text{ is occulted.} \end{cases} \tag{33}$$

This formulation has the advantage of offering flexibility in how to create a Boolean occlusion map because it is sufficient to apply to it a threshold s. We propose finally the relationship

$$O_1(p) = \begin{cases} 1 \text{ if } \sum\limits_q 1 \geq s, \quad \forall q / \lfloor D_2(q) + q \rfloor = p, \\ 0 \text{ if } p \text{ is occulted.} \end{cases} \tag{34}$$

Note that if we limit ourselves to a $s = 1$ threshold, we will tend to over-estimate the visibility because it only takes one back-projected contribution reaching the pixel to be considered visible in both images. We generated the Boolean occlusion maps according to Equation (34) setting $s = \lfloor N(p)/2 \rfloor$ where $N(p) = 100$ for any p (see Figure 6).

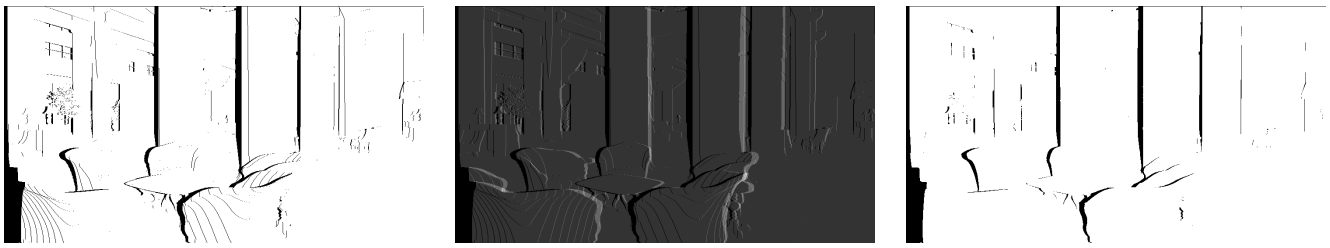


Figure 8: Example of occlusion map obtained for a stereo pair (I_1, I_2) for which the camera made a maximum fronto-parallel displacement of 50 pixels. The left map obtained with $N = 1$ is not dense enough because level lines of occulted pixels appear. The middle map obtained by summation from the relation (33) and $N = 100$, allows the creation of a dense Boolean map, right after thresholding with $s = 50$.

10 Conclusion

The qualitative evaluation of very accurate stereoscopic algorithms requires having at one's disposal stereo pairs with minimal and quantified noise, accompanied by ground truths as accurate as possible. We presented a new method of generating ground truths with synthetic images. This approach, which takes advantage of spatial oversampling ray tracing, allows to make maps of disparities and occlusions with a level of accuracy which has never been reached before. Furthermore, the iterative creation of image synthesis during rendering allows us to make an analytical estimate of the residual noise. This method allows us to offer to the scientific community, a new dataset of 66 reference stereo pairs, directed to evaluating stereoscopic and 3D reconstruction algorithms.

Acknowledgements

This work was partly funded by the Centre National d'Études Spatiales (CNES, MISS Project), the European Research Council (advanced grant Twelve Labours), the Office of Naval research (ONR grant N00014-14-1-0023), the DGA Stéréo project, the ANR-DGA (project ANR-12-ASTR-0035) and the Institut Universitaire de France.

Image Credits

Images by the author (license CC BY-NC-SA 4.0).
Designers cf. Table 3, except

 Copyright [18].

A Demonstrations

In the following, we denote respectively $\mathcal{X}^n$ and $\mathcal{Y}^n$ the parent random variables of the random variables $\mathcal{X}_k^n$ and $\mathcal{Y}_k^n$ for $k = 1, \ldots, K$.

A.1 Proof of Proposition 1

Proof. From definition (12) of MSE we have

$$\begin{aligned}
MSE(\bar{u}_k, \bar{u}_K) &= \frac{1}{3N_I}\mathbb{E}\|\bar{u}_k - \bar{u}_K\|_2^2 \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathbb{E}[(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n)^2], \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n) + [\mathbb{E}(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n)]^2 \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n) + [\mathbb{E}(\bar{\mathcal{Y}}_k^n - \mathcal{Y}^n) - \mathbb{E}(\bar{\mathcal{Y}}_K^n - \mathcal{Y}^n))]^2 \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}\left[\frac{1}{k}\sum_{i=1}^{k}\mathcal{Y}_i^n - \frac{1}{K}\sum_{i=1}^{K}\mathcal{Y}_i^n\right] + [\mathbb{E}(\bar{\mathcal{Y}}_k^n - \mathcal{Y}^n) - \mathbb{E}(\bar{\mathcal{Y}}_K^n - \mathcal{Y}^n))]^2.
\end{aligned}$$

The hypothesis of an unbiased rendering and relation (6) apply to images $(v_k)_k$ as well as to images $(u_k)_k$. They imply that

$$\begin{aligned}
MSE(\bar{u}_k, \bar{u}_K) &= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}\left[\left(\frac{1}{k} - \frac{1}{K}\right)\sum_{i=1}^{k}\mathcal{Y}_i^n - \frac{1}{K}\sum_{i=k+1}^{K}\mathcal{Y}_i^n\right] + 0 \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\left(\frac{1}{k} - \frac{1}{K}\right)^2\mathrm{var}\left(\sum_{i=1}^{k}\mathcal{Y}_i^n\right) + \frac{1}{K^2}\mathrm{var}\left(\sum_{i=k+1}^{K}\mathcal{Y}_i^n\right) \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\left(\frac{1}{k} - \frac{1}{K}\right)^2 k\mathrm{var}(\mathcal{Y}^n) + \frac{1}{K^2}(K-k)\mathrm{var}(\mathcal{Y}^n) \\
&= \frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}(\mathcal{Y}^n)\left[\left(\frac{1}{k} - \frac{1}{K}\right)^2 k + \frac{1}{K^2}(K-k)\right] \\
&= \left(\frac{1}{k} - \frac{1}{K}\right)\frac{1}{3N_I}\sum_{n=1}^{3N_I}\mathrm{var}(\mathcal{Y}^n) \\
&= \left(\frac{1}{k} - \frac{1}{K}\right)\mathrm{var}(u_t). \qquad (35)
\end{aligned}$$

As $u_t = v_t * h$ and from property (7), we have

$$\begin{aligned}
\mathrm{var}(\mathcal{Y}^n) &= \mathrm{var}\left[\sum_{m=-\infty}^{\infty}\mathcal{X}^{n-m}h(m)\right] \\
&= \sum_{m=-\infty}^{\infty}\mathrm{var}(\mathcal{X}^{n-m})h^2(m) + 2\sum_{m=-\infty}^{\infty}\sum_{l=m+1}^{\infty}h(m)h(l)\mathrm{cov}(\mathcal{X}^{n-m}, \mathcal{X}^{n-l}) \\
&= \sum_{m=-\infty}^{\infty}\mathrm{var}(\mathcal{X}^{n-m})h^2(m) + 0. \qquad (36)
\end{aligned}$$

Introducing $\mathrm{var}(v_t)$ from Definition 1 and using (36) then

$$\begin{aligned} MSE(\bar{u}_k, \bar{u}_K) &= \left(\frac{1}{k} - \frac{1}{K}\right) \frac{1}{3N_I} \sum_{n=1}^{3N_I} \sum_{m=-\infty}^{\infty} \mathrm{var}(\mathcal{X}^{n-m}) h^2(m) \\ &= \left(\frac{1}{k} - \frac{1}{K}\right) \sum_{m=-\infty}^{\infty} h^2(m) \frac{1}{3N_I} \sum_{n=1}^{3N_I} \mathrm{var}(\mathcal{X}^{n-m}) \\ &= \left(\frac{1}{k} - \frac{1}{K}\right) \|h\|_2^2 \mathrm{var}(v_t). \end{aligned} \tag{37}$$

According to the definition of the statistics (9) and to the equivalence between (35) and (37) we have

$$\begin{aligned} \mathrm{var}(\bar{u}_K) &= \mathrm{var}\left(\frac{u_1 + \cdots + u_K}{K}\right) \\ &= \frac{1}{K}\mathrm{var}(u) \\ &= \frac{1}{K}\|h\|_2^2 \mathrm{var}(v). \end{aligned}$$

Thus,

$$\begin{aligned} \mathrm{var}(\bar{u}_K) &= \frac{1}{K}\left(\frac{1}{k} - \frac{1}{K}\right)^{-1} EQM(\bar{u}_k, \bar{u}_K) \\ &= \frac{k}{K-k} EQM(\bar{u}_k, \bar{u}_K). \end{aligned}$$

□

A.2 Proof of Proposition 2

Proof. According to Definition 1, and since the histogram modification corresponds to an affine transformation of slope α_n and bias β_n, applied to each pixel, we have

$$\begin{aligned} \mathrm{var}(f_L(\bar{u}_K)) &= \frac{1}{3N_I} \sum_{n=1}^{3N_I} \mathrm{var}(f_L(\bar{\mathcal{Y}}_K^n)) \\ &= \frac{1}{3N_I} \sum_{n=1}^{3N_I} \mathrm{var}(\alpha_n \bar{\mathcal{Y}}_K^n + \beta_n) \\ &= \frac{1}{3N_I} \sum_{n=1}^{3N_I} \alpha_n^2 \mathrm{var}(\bar{\mathcal{Y}}_K^n). \end{aligned}$$

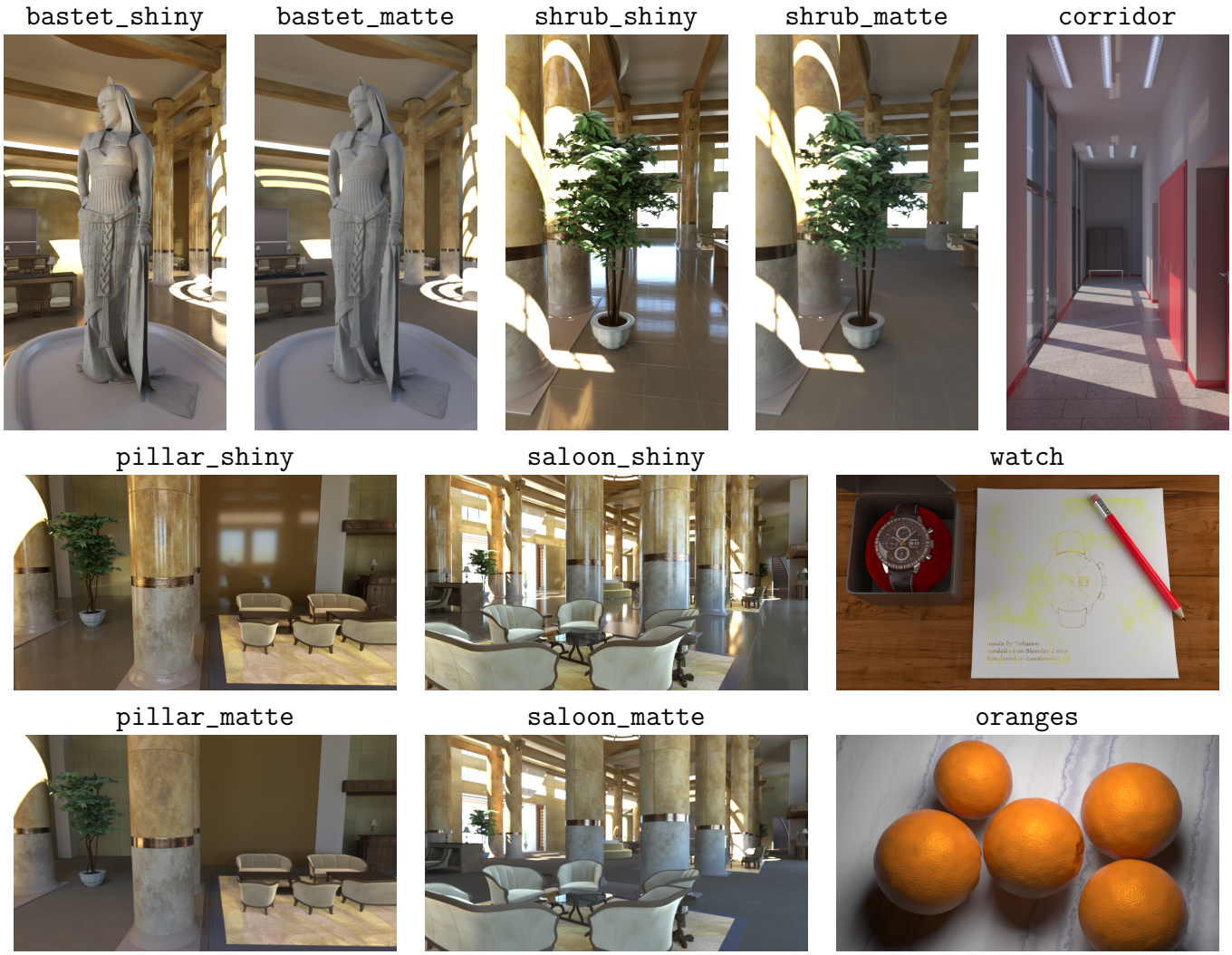
From (10) and relationship $\mathbb{E}[(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n)^2] = (\frac{1}{k} - \frac{1}{K})\mathrm{var}(\bar{\mathcal{Y}}^n)$ seen in A.1 we have $\mathrm{var}(\bar{\mathcal{Y}}_K^n) = \frac{k}{K-k}\mathbb{E}[(\bar{\mathcal{Y}}_k^n - \bar{\mathcal{Y}}_K^n)^2]$ which is then estimated by $\frac{k}{K-k}(\bar{u}_k(n) - \bar{u}_K(n))^2$. □

B Constitution of the Database

We used seven different synthetic scenes observed from four points of view and named $v_{+000}, v_{+001}, v_{+010}$ and v_{+050}. Four of these scenes were considered with two different color characteristics: with and

without reflective materials. In the end, the base consists of 66 stereo pairs. Figure 9 shows the different scenes for view v_{+000}. The images produced have a size of 960×540 in portrait mode and of 540×960 in landscape mode. They are offered in three formats, namely: EXR 16-bit, 16-bit PNG and TIF 16 bit float. The first two mentioned formats come directly from the rendering engine LuxRender while the third was obtained by a post-processing of EXR images detailed in Section B.3. Each pixel received an average of over 2 million contributions. The noise estimate was made by creating, for each of the 11 scenes u_K, a identical u_k scene but with a rendering duration k much lower by around 100 000 contributions per pixel and then by applying formula (11). In practice the rendering duration is the number of contributions per pixel.

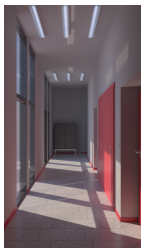
Figure 9: All the scenes constituting the dataset. The images of this illustration reflect the views v_{+000} in PNG 16 bits created by the render.



B.1 Scene Designers

Table 3 lists the scene designers who must be credited when using the CMLA database.

Table 3: 3D scenes designers.

Designers	Scenes
Peter Sandbacka	hotel lobby 
Simon Wendsche	school corridor 
Tahseen Jamal	sketch watch  , oranges 

B.2 File Naming Convention

`scene_+000_1.cbs`	Binary file containing the set of contributions expressed in frame $\mathfrak{R}_A$ obtained after regular sampling of pixels in the image ($N = 1$)
`scene_+000_100.cbs`	Binary file containing the set of contributions expressed in frame $\mathfrak{R}_A$ obtained after regular sampling of pixels in the image ($N = 100$)
`scene_+000.mat`	Binary file containing coefficients of transformation matrices $\mathbf{R}_{IC}$ and $\mathbf{R}_{CA}$
`scene_+000.exr`	Image of view v_{+000} of scene `scene` provided by renderer in EXR format
`scene_+000.png`	Image of view v_{+000} of scene `scene` provided by renderer in PNG 16 bits format
`scene_+000.tif`	Equalized image between $[0, 255]$, according to the modified algorithm of Lisani et al.
`scene_+000.txt`	Text file that contains various informations
`scene_dispx_+000_to_+050.tif`	Disparity map along x from view v_{+000} to view v_{+050}
`scene_dispy_+000_to_+050.tif`	Disparity map along y from view v_{+000} to view v_{+050}
`scene_occu_+000_to_+050.tif`	Occlusion map obtained from (33) after regular sampling $N = 100$
`scene_occu_+000_to_+050.png`	Occlusion map obtained from (34) after thresholding $s = 50$ of the previous map

B.3 TIF Format Images

The images produced natively by LuxRender are either EXR, or PNG format. But in the first case the color values are not necessarily limited to $[0, 255]$ and the dynamic is not updated; in the

second case values are discretized in $[\![0, 65535]\!]$ and possibly saturated. In order to best preserve all chromatic information and to offer stereo algorithms the range $[0, 255]$ conventionally used, we adapted the dynamics of EXR images using a piecewise affine histogram adjustment [15]. It originally processes only input PNG images in $[\![0, 255]\!]$. Changes to this algorithm concern the addition of the library `iio.h` allowing image processing in floating point format and normalization in $[0, 255]$ of the EXR input image. More specifically, the linear transformation applies

$$\check{v} = \frac{255 \cdot v}{\max_{RGB}(v)}, \tag{38}$$

to the input image EXR v, where $\max_{RGB}(v)$ is the maximum of the three channels, then image $\check{v}$ is treated with the original algorithm. We take into account this transformation in computing (14). To apply the same contrast change to all images v of a same scene, we created the image union $V = v_{+000} \cup v_{+001} \cup v_{+010} \cup v_{+050}$ then dealt with V using the modified algorithm. The result $\tilde{V}$ is then split to retrieve images TIF $\tilde{v}_{+000}, \tilde{v}_{+001}, \tilde{v}_{+010}$ and $\tilde{v}_{+050}$. The three parameters of the equalization method, applied to the different scenes are shown in Table 4.

Table 4: Parameter values used by the modified version of histogram equalization algorithm by Lisani et al., applied to different EXR images.

Scene	minimum slope	maximum slope	number of control points
`bastet_matte`	0	20	1000
`bastet_shiny`	0	100	1000
`corridor`	0	5	10
`oranges`	0	2	10
`pillar_matte`	0	20	1000
`pillar_shiny`	0	40	1000
`saloon_matte`	0	20	1000
`saloon_shiny`	0	40	1000
`shrub_matte`	0	20	1000
`shrub_shiny`	0	40	1000
`watch`	0	5	1000

B.4 Noise Estimates File

The text file `scene_+000.txt` contains the various noise estimates explained in Section 6 in the format described in Table 5.

B.5 Disparity Maps in the Vertical Direction

Table 6 shows the maximum amplitude of disparity maps along the y axis. They do not exceed $1,722.10^{-4}$ pixels.

B.6 Binary File Content

Contribution file. The file contains information about the image dimensions, then, for each pixel, its coordinates, the number N of its contributions $\mathbf{c}_n$, finally their x_n and y_n coordinates and $d_n = \|\mathbf{P}(p, n) - O_C\|$. The x_n and y_n coordinates are already expressed in the image reference frame and not in the camera frame, so it is not necessary to apply the inverse transformation $\mathbf{R}_{IC}^{-1}$ to them.

Table 5: Content of text file `scene_+000.txt`.

Field	Meaning
`nlin`	image rows number
`ncol`	image column number
`fov`	image field of view expressed in degrees
`baseline_with_+001`	baseline according to (31) between v_{+000} and v_{+001} etc.
`var_uk`	variance according to (11)
`sigma2`	normalized variance according to (13)
`var_fluk`	variance of the quantified image on $[0, 255]$ according to (14)
`snr_uk`	SNR of $\bar{u}_K$ according to (15)
`snr_fluk`	SNR of $f_L(\bar{u}_K)$ according to (15)

Table 6: Maximum amplitude of disparity maps along the y axis.

Scene	v_{+001}	v_{+010}	v_{+050}
`shrub_matte,shrub_shiny`	$3,197.10^{-5}$	$3,958.10^{-5}$	$1,122.10^{-4}$
`bastet_matte,bastet_shiny`	$1,331.10^{-4}$	$2,602.10^{-5}$	$1,722.10^{-4}$
`corridor`	$2,810.10^{-5}$	$2,816.10^{-5}$	$2,992.10^{-5}$
`watch`	$1,518.10^{-16}$	$1,527.10^{-16}$	$1,485.10^{-16}$
`oranges`	$4,836.10^{-6}$	$7,254.10^{-6}$	$4,836.10^{-6}$
`pillar_matte,pillar_shiny`	$4,807.10^{-6}$	$1,421.10^{-7}$	$4,246.10^{-6}$
`saloon_matte,saloon_shiny`	$4,383.10^{-6}$	$5,707.10^{-5}$	$1,001.10^{-4}$

Both coordinates are decimal because they are subpixel. The format of data in the binary file is the following, knowing that i and j correspond respectively to the i$^{\text{th}}$ line and j$^{\text{th}}$ column of $nrow \times ncol$ size image and $ncbs$ represents the maximum number of contributions in the area of the pixel:

$$\overbrace{nrow}^{int}\ \overbrace{ncol}^{int}\ \overbrace{ncbs}^{int}\ \overbrace{0}^{int}\ \overbrace{0}^{int}\ \overbrace{N}^{int}\ \overbrace{x_1}^{float}\ \overbrace{y_1}^{float}\ \overbrace{d_1}^{float} \ldots\ldots \overbrace{x_N}^{float}\ \overbrace{y_N}^{float}\ \overbrace{d_N}^{float}\ \overbrace{0}^{int}\ \overbrace{1}^{int} \ldots\ldots \overbrace{i}^{int}\ \overbrace{j}^{int} \ldots$$

Transformation matrices file. The binary file contains two matrices $\mathbf{R}_{IC}$ and $\mathbf{R}_{CA}$ stored one after the other in the format:

$$\overbrace{\text{nb rows}}^{int}\ \overbrace{\text{nb cols}}^{int}\ \overbrace{R_{IC}[0][0]}^{float}\ \overbrace{R_{IC}[0][1]}^{float}\ \overbrace{R_{IC}[0][2]}^{float}\ \overbrace{R_{IC}[0][3]}^{float}\ \overbrace{R_{IC}[1][0]}^{float} \ldots \overbrace{R_{IC}[3][3]}^{float}\ \overbrace{\text{nb rows}}^{int}\ \overbrace{\text{nb cols}}^{int}\ \overbrace{R_{CA}[0][0]}^{float} \ldots$$

where int and $float$ have a size of 4 bytes.

References

[1] M. Berger, J.A. Levine, L.G. Nonato, G. Taubin, and C.T. Silva, *A benchmark for surface reconstruction*, ACM Transactions on Graphics (TOG), 32 (2013), pp. 20:1–20:17. http://doi.acm.org/10.1145/2451236.2451246.

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, *A naturalistic open source movie for optical flow evaluation*, in European Conference on Computer Vision (ECCV), A. Fitzgibbon et al. (Eds.), ed., Part IV, LNCS 7577, Springer-Verlag, Oct. 2012, pp. 611–625.

[3] B. Curless and M. Levoy, *A volumetric method for building complex models from range images*, in Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, New York, NY, USA, 1996, ACM, pp. 303–312. http://doi.acm.org/10.1145/237170.237269.

[4] J. Delon and B. Roug, *Le phénomène d'adhérence en stéréoscopie dépend du critère de corrélation*, GRETSI 2001, 2001.

[5] ——, *Small baseline stereovision*, Journal of Mathematical Imaging and Vision, 28 (2007), pp. 209–223. http://dx.doi.org/10.1007/s10851-007-0001-1.

[6] J. Digne, N. Audfray, C. Lartigue, C. Mehdi-Souzani, and J-M. Morel, *Farman institute 3D point sets - high precision 3D data sets*, Image Processing On Line, 1 (2011). https://doi.org/10.5201/ipol.2011.dalmm_ps.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, *Vision meets robotics: The KITTI dataset*, International Journal of Robotics Research (IJRR), (2013).

[8] R.C. Gonzalez and R.E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 013168728X.

[9] M. Gupta, A. Agrawal, A. Veeraraghavan, and S.G. Narasimhan, *A practical approach to 3D scanning in the presence of interreflections, subsurface scattering and defocus*, International Journal of Computer Vision, 102 (2013), pp. 33–55. http://dx.doi.org/10.1007/s11263-012-0554-3.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, New York, NY, USA, 2 ed., 2003. ISBN 0521540518.

[11] R. Hess, *Blender Foundations: The Essential Guide to Learning Blender 2.6*, Focal Press, 2010. ISBN 0240814304, 9780240814308.

[12] J.T. Kajiya, *The rendering equation*, in Computer Graphics, 1986, pp. 143–150.

[13] R. Klette, N. Kruger, T. Vaudrey, K. Pauwels, M. van Hulle, S. Morales, F.I. Kandil, R. Haeusler, N. Pugeault, C. Rabe, and M. Lappe, *Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database*, IEEE Transactions on Vehicular Technology, 60 (2011), pp. 2012–2026. http://dx.doi.org/10.1109/TVT.2011.2148134.

[14] D. Kondermann, S. Abraham, G. Brostow, W. Frstner, S. Gehrig, A. Imiya, B. Jhne, F. Klose, M. Magnor, H. Mayer, R. Mester, T. Pajdla, R. Reulke, and H. Zimmer, *On performance analysis of optical flow algorithms*, in Outdoor and Large-Scale Real-World Scene Analysis, F. Dellaert, J-M. Frahm, M. Pollefeys, L. Leal-Taix, and B. Rosenhahn, eds., vol. 7474 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 329–355. http://dx.doi.org/10.1007/978-3-642-34091-8_15.

[15] J.L. Lisani, A.B. Petro, and C. Sbert, *Color and contrast enhancement by controlled piecewise affine histogram equalization*, Image Processing On Line, 2 (2012), pp. 243–265. https://doi.org/10.5201/ipol.2012.lps-pae.

[16] S. Martull, M. Peris, and K. Fukui, *Realistic CG stereo image dataset with ground truth disparity maps*, Technical report of IEICE. PRMU, 111 (2012), pp. 117–118. http://ci.nii.ac.jp/naid/110009482347/en/.

[17] S. Meister, B. Jähne, and D. Kondermann, *Outdoor stereo camera system for the generation of real-world benchmark data sets*, Optical Engineering, 51 (2012), p. 021107.

[18] M. Pharr and G. Humphreys, *Physically Based Rendering, Second Edition: From Theory To Implementation*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd ed., 2010. ISBN 0123750792, 9780123750792.

[19] H. Riemenschneider, A. Bodis Szomoru, J. Weissenberg, and L.J. Van Gool, *Learning where to classify in multi-view semantic segmentation*, 2014, pp. V: 516–532.

[20] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nei, X. Wang, and P. Westling, *High-resolution stereo datasets with subpixel-accurate ground truth*, in Pattern Recognition, X. Jiang, J. Hornegger, and R. Koch, eds., Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 31–42. http://dx.doi.org/10.1007/978-3-319-11752-2_3.

[21] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 47 (2001), pp. 7–42.

[22] ——, *High-accuracy stereo depth maps using structured light*, in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, Washington, DC, USA, 2003, IEEE Computer Society, pp. 195–202. http://dl.acm.org/citation.cfm?id=1965841.1965865.

[23] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, *A comparison and evaluation of multi-view stereo reconstruction algorithms*, in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, June 2006, pp. 519–528. http://dx.doi.org/10.1109/CVPR.2006.19.

[24] R. Szeliski, *Algorithms and Applications*, Springer International Publishing, 2011, ch. Geometric primitives and transformations, pp. 27–52.

[25] E. Veach, *Robust Monte Carlo methods for light transport simulation*, PhD thesis, Stanford University, 1997.

[26] R. Veysseyre, *Statistique et probabilités*, Dunod, 3rd ed., 2014, ch. 17, pp. 330–334.

[27] T.J. Victorino, *Luxrender*, Log Press, 2012. https://books.google.fr/books?id=CiyLMAEACAAJ.

[28] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, *Lessons and insights from creating a synthetic optical flow benchmark*, in ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation, A. Fusiello et al. (Eds.), ed., Part II, LNCS 7584, Springer-Verlag, Oct. 2012, pp. 168–177.