3D reconstruction of urban environments based on fisheye stereovision

Julien Moreau, Sébastien Ambellouis, Yassine Ruichek

3D reconstruction of urban environments based on fisheye stereovision

Julien Moreau*, Sébastien Ambellouis† and Yassine Ruichek*
*IRTES-SET, UTBM, 90010 Belfort cedex, France
Email: { julien.moreau, yassine.ruichek }@utbm.fr
†Univ Lille Nord de France- F-59000 Lille, IFSTTAR, LEOST, F-59650 Villeneuve d'Ascq, France
Email: sebastien.ambellouis@ifsttar.fr

***Abstract*—In this paper, we show some results about 3D urban scenes reconstruction using a fisheye stereovision setup. We propose an analytical analysis of epipolar geometry of the system and an analytical description of tools to compute a 3D point cloud from matched pixels. The novelty is that we do not rectify the images and that we match points along 3D or 2D epipolar curves. The matching process is based on a global dynamic programming algorithm that we adapt to take into account continuous epipolar curve equation. We show 3D point cloud in the case of synthetic images.**

***Keywords*-fisheye; spherical; stereovision; 3D; epipolar; urban;**

I. Introduction

This paper presents a work that aims to compute 3D structure of an urban scene by using a stereo sensor based on fisheye cameras. From application point of view, the objective is to increase the accuracy of a satellite based localization system by correcting errors due to the multipath effects caused by reflection of satellite signals on urban objects. For this, we need to gather the structure information in a precise 3D model of the surrounding car environment to calculate pseudorange errors.

We have to model all the buildings and streets around the car. Thus, a wide field of view is required. An obvious solution is to use several pinhole cameras. Other authors have proposed more original sensors like catadioptric sensors. A catadioptric system is a combination of lenses and mirrors. Depending on the shape of the mirror, deformations are different. These sensors are usually used to obtain an omnidirectional view of the environment by using spherical, parabolic or hyperbolic shapes for the mirror.

In [1], Gonzalez-Barbosa develops an omnidirectional stereovision system for the autonomous navigation application of a robot in natural environments. He uses parabolic catadioptric sensors.
Kawanishi *et al.* in [2] propose omnidirectional stereovision from a mobile robot equipped with one catadioptric hyperbolic camera. They are able to compute 3D distances and model the environment by matching the images obtained at different positions with an estimation of the robot's movement. They track interest points and automatically compute the essential matrix.
Ragot in [3] uses two hyperbolic mirror cameras. He mainly proposes calibration method with specific test patterns, and presents a volumetric way to reconstruct the scene in 3D.

A fisheye camera delivers a wide field of view. It is smaller and lighter than a catadioptric sensor and easier to install.

Shah and Aggarwal in [4] present an autonomous mobile robot navigation system in an indoor environment. This system uses two fisheye sensors calibrated using a polynomial distortion model. The robot can detect walls and compute a 3D model of corridors, after images rectification and lines extraction according to geometric knowledge of the environment: horizontal and vertical lines can easily be used to distinguish walls, floor and ceiling.
Mičušìk *et al.* in [5] propose a 3D reconstruction of the surrounding scene with two or more uncalibrated fisheye images. They automatically detect point correspondences to process an autocalibration.
Li in [6] and [7] defines and computes spherical disparity maps, and draws 3D representations from the computed 3D distances. He calibrates his sensors and transforms images to get horizontal epipolar lines in order to apply existing standard matching points process. Considering Li's definition of spherical disparity, our paper clarifies how to compute 3D points' position.
Herrera *et al.* in [8] propose fisheye stereovision in forest environments. The objective is to compute disparity maps with no image rectification. They start by a segmentation of images to separate textures of interest and discard those that are not useful. Then, they match trees by using fuzzy logic with some parameters and smooth the results with a global neuronal network based matching algorithm. A noticeable difference between Herrera *et al.* and previously cited authors, is that they place the cameras in direction of sky, not in a frontal direction. This way, the whole surrounding scene is really acquired, and most important information are not in the center of images but at the periphery. By the way, our work shares this feature with Herrera's work.

In our work, we use fisheye lenses which optical axis are vertical and mounted on cameras placed along the longitudinal axis of the car. This stereo rig allows us to reach 3D information 360 degrees around the vehicle by applying a stereo matching algorithm on a single image

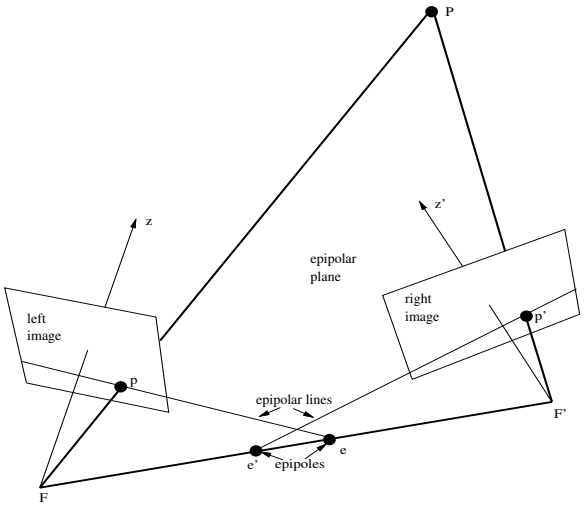


Figure 1. Epipolar and point correspondence geometry. F and F' are indicating the centers of the two cameras. The camera baseline FF' and the point P form an epipolar plane. Epipolar lines are in the epipolar plane and intersect the baseline FF' at the epipoles. p and p' are the corresponding image points of P. Figure issued from the Horaud and Monga's book [9].

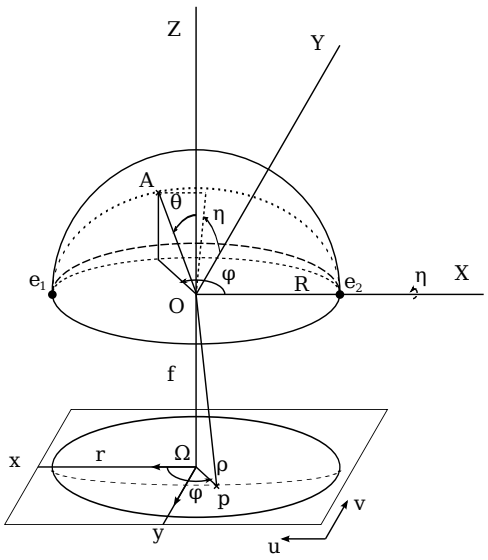


Figure 2. Spherical projection of a fisheye lens mounted on a camera. The point A corresponds to the intersection of the ray passing through a viewed point of the scene and the optical center with the hemisphere's (lens) surface, and p its projection on the image plane.

pair. High resolution cameras are used to compensate for the optical effect of the fisheye projection process.

The paper is organized as follows. The section II presents an analytical description of the epipolar geometry for a calibrated fisheye stereovision system, and how to retrieve 3D points coordinates from matched points. The section III describes the matching algorithm we use to compute 3D point cloud and presents some results on synthetic images.

II. Epipolar geometry

A. Generalities

Epipolar geometry is described by Hartley and Zisserman in [10] as the "intrinsic projective geometry between two views". Epipolar geometry for two pinhole cameras is illustrated in figure 1. This property is very interesting and largely exploited in most of the matching algorithms to reduce the search space of homologous pixels in several images. For an image pair, the homologous point of each pixel in the first image is located on a line in the second image. Often, a calibration and a rectification steps are computed to make epipolar lines parallel and coincident with the rows of the images. In this configuration, a scan-line matching process can be used.

B. Fisheye case study

As illustrated in figure 2, fisheye projection model is a spherical projection that has two epipoles e_1 and e_2, contrary to conventional pinhole model. Many authors like Li in [6] and Ramalingam in [11] show the main models used for fisheye lens. Table I gives spherical projection models that can be used to model fisheye lens. Each model connects the angle of the incident ray and the radius of the projected point's position on the image plane. In this table, θ is the colatitude angle, ρ is the radius on the image plane and f is the focal length of the lens (see figure 2).

Table I
Main fisheye projection models.

Projection	Expression of radius $\rho = \mathrm{proj}_f(\theta)$	Expression of angle $\theta = \mathrm{proj}_f^{-1}(\rho)$
Equidistant	$\rho = f\theta$	$\theta = \frac{\rho}{f}$
Orthographic	$\rho = f\sin(\theta)$	$\theta = \arcsin(\frac{\rho}{f})$
Equisolid	$\rho = 2f\sin(\frac{\theta}{2})$	$\theta = 2\arcsin(\frac{\rho}{2f})$
Stereographic	$\rho = 2f\tan(\frac{\theta}{2})$	$\theta = 2\arctan(\frac{\rho}{2f})$

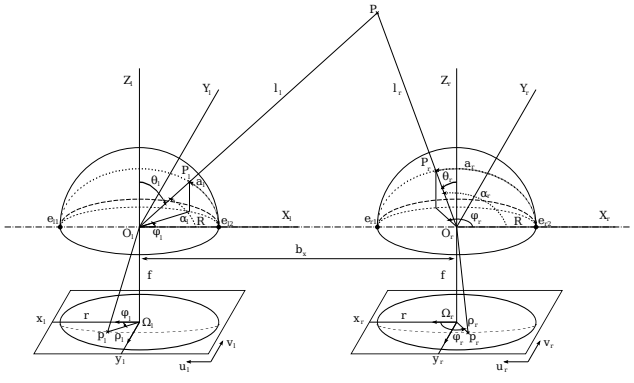


Figure 3. Calibrated fisheye stereo sensor. The point P of the scene intersects left and right hemispheres respectively in P_l and P_r, and is viewed as p_l and p_r in the recorded images of both cameras.

According to Arican and Frossard in [12], the principle of epipolar geometry remains true for others projection models like the fisheye projection model. In this case, epipolar lines are transformed in 3D epipolar curves described on the lens surface i.e. circle arcs. As in the case of pinhole cameras, a calibration and a rectification steps allow to make 3D epipolar curves identical for both lenses.

In this configuration, illustrated in figure 3, the points e_{l1}, O_l, e_{l2}, e_{r1}, O_r and e_{r2} lie on the same line

and the epipolar curves are the intersection of the plane (O_l, P, O_r) and both fisheye lenses. Both epipolar curves are projected on the camera sensors using one of the projection model presented in Table I to yield two identical 2D epipolar curves which is possible to match points using adapted scan-line techniques.

C. Epipolar curve calculation

Let p_i be a pixel of the first image (in our case i can be l for the left sensor or r for the right sensor) which coordinates are (x_i, y_i) in the image centered coordinates system. In the image centered polar reference frame, coordinates of p_i are given by:

$$p_i \begin{pmatrix} \rho_i = \sqrt{x_i^2 + y_i^2} \\ \varphi_i = \arctan(\frac{y_i}{x_i}) \end{pmatrix}_{\text{polar frame}}$$

The projection P_i of p_i on the lens' hemisphere is obtained thanks to one of the projection models previously described:

$$P_i \begin{pmatrix} R \\ \theta_i = \text{proj}_f^{-1}(\rho_i) \\ \varphi_i \end{pmatrix}_{\text{spherical frame}}$$

The conversion to 3D Cartesian coordinates gives:

$$P_i \begin{pmatrix} X_i = R\sin(\theta_i)\cos(\varphi_i) \\ Y_i = R\sin(\theta_i)\sin(\varphi_i) \\ Z_i = R\cos(\theta_i) \end{pmatrix}_{\text{Cartesian frame}}$$

The 3D epipolar curve is described by the circle arc passing through P_i and the epipoles e_{i1} and e_{i2} and which center is the optical center of the image. Its equation is:

$$\begin{pmatrix} R\cos(\gamma) \\ R\cos(\eta_i)\sin(\gamma) \\ R\sin(\eta_i)\sin(\gamma) \end{pmatrix}_{\text{Cartesian frame}}$$

where R is the radius of the hemisphere, η is the rotation angle around the x axis, γ is varying in $[0; \pi]$ and

$$\cos(\eta_i) = \frac{\sin(\theta_i)\sin(\varphi_i)}{\sqrt{1 - \sin(\theta_i)^2\cos(\varphi_i)^2}} \quad (1)$$

$$\text{and } \sin(\eta_i) = \frac{\cos(\theta_i)}{\sqrt{1 - \sin(\theta_i)^2\cos(\varphi_i)^2}} \quad (2)$$

In the spherical coordinates system, the equation is given by:

$$\begin{pmatrix} R \\ \Theta = \arccos(\sin(\eta_i)\sin(\gamma)) \\ \Phi = \pm\arccos(\frac{\cos(\gamma)}{\sqrt{1-(\sin(\eta_i)\sin(\gamma))^2}}) \end{pmatrix}_{\text{spherical frame}}$$

Finally, the equation of the epipolar curves projected on the camera plane is given by following expression:

$$\begin{pmatrix} \text{proj}_f(\Theta) \\ \Phi \end{pmatrix}_{\substack{\text{polar} \\ \text{frame}}} = \begin{pmatrix} x_i = \text{proj}_f(\Theta).\cos(\Phi) \\ y_i = \text{proj}_f(\Theta).\sin(\Phi) \end{pmatrix}_{\substack{\text{centered} \\ \text{frame}}}$$

Note that in the calibrated case illustrated in figure 3, η_i are the same angle. In the following, $\eta_i = \eta$ for all i values.

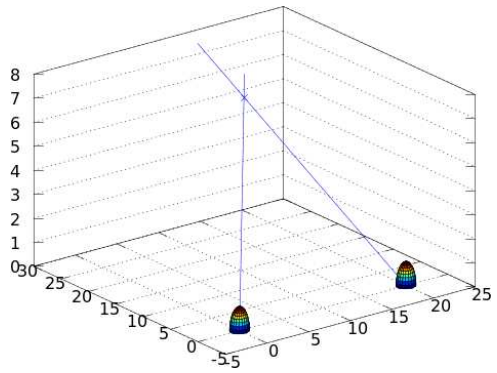


Figure 4. The colored half-spheres represent the two fisheye lenses. Two straight lines issued from the optical centers and passing through the projection of the scene point on the hemisphere are merging in the exact position of the observed scene point. This figure has been obtained by applying the described principles in a mathematics software.

D. Position of points in space

This section describes two solutions to compute a 3D point cloud knowing the points pairs matched along the epipolar curves, which equations are given in the previous section.

1) From the projections on both hemispheres: As it is illustrated in figure 4, the projected straight lines of a pair of corresponding points on both hemispheres cross themselves in the position of the original scene's point in space. The coordinates of P, in the left camera Cartesian frame (O_l, x_l, y_l, z_l), can be expressed in terms of the coordinates of its projections $P_l(R, \theta_l, \varphi_l)$ and $P_r(R, \theta_r, \varphi_r)$ (see figure 3).

In the left camera frame (O_l, x_l, y_l, z_l), the coordinates of optical centers are:

$$O_l \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{\text{left frame}} \quad \text{and } O_r \begin{pmatrix} b_x \\ 0 \\ 0 \end{pmatrix}_{\text{left frame}}$$

where b_x is the baseline of the stereo system i.e. the distance between optical centers.

The O_lP and O_rP are given by:

$$O_lP \begin{pmatrix} t\sin(\theta_l)\cos(\varphi_l) \\ t\sin(\theta_l)\sin(\varphi_l) \\ t\cos(\theta_l) \end{pmatrix}_{\text{left frame}} \text{ with } t \in [0; +\infty[$$

$$O_rP \begin{pmatrix} b_x + t'\sin(\theta_r)\cos(\varphi_r) \\ t'\sin(\theta_r)\sin(\varphi_r) \\ t'\cos(\theta_r) \end{pmatrix}_{\text{left frame}} \text{ with } t' \in [0; +\infty[$$

P is the intersection of O_lP and O_rP and its coordinates are given by replacing t or t' by:

$$t = \frac{b_x}{\sin(\theta_l)\cos(\varphi_l) - \cos(\theta_l)\tan(\theta_r)\cos(\varphi_r)}$$

or:

$$t' = \frac{b_x}{\cos(\theta_r)\tan(\theta_l)\cos(\varphi_l) - \sin(\theta_r)\cos(\varphi_r)}$$

2) From α_i angles: In [7], Li proposes to compute spherical disparities (spherical normalized disparity is defined as $\alpha_r - \alpha_l$), and the distance between the point P and the center of both hemispheres. Let l_l and l_r be these two distances respectively from the center of the left and right lens. As illustrated in the figure 3, by applying the sine law, we get (see figure 3):

$$l_l = \frac{b_x \sin(\alpha_r)}{\sin(\alpha_r - \alpha_l)} \text{ and } l_r = \frac{b_x \sin(\alpha_l)}{\sin(\alpha_r - \alpha_l)}$$

Hence, one can exploit and extend these results to compute coordinates of scene's points. α_i values can be obtained by two ways:

- with the relation $\alpha_i = \arccos(\sin(\theta_i)\cos(\varphi_i))$,
- or by using γ; in fact α and γ are the same angles, and when we look through the epipolar curve by varying γ, we obtain the α of a specific point.

Finally, the coordinates of P in the left reference frame are defined by:

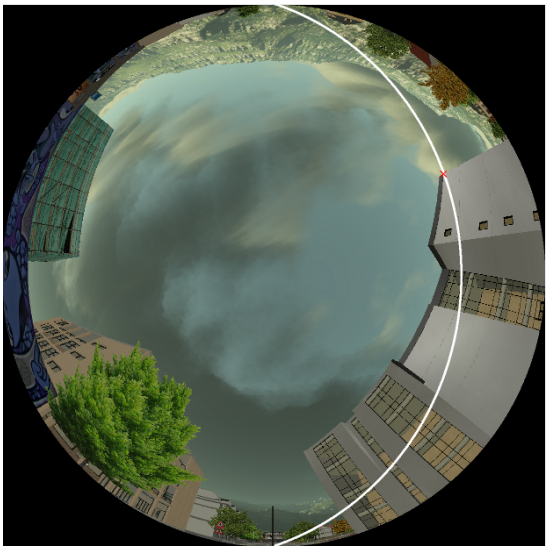
$$P \begin{pmatrix} l_l \cos(\alpha_l) \\ l_l \sin(\alpha_l)\cos(\eta) \\ l_l \sin(\alpha_l)\sin(\eta) \end{pmatrix}_{\text{left frame}}$$

III. Experimental results

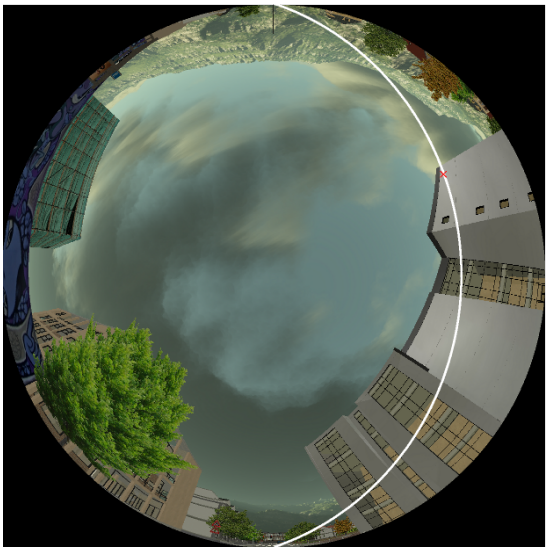
In this section, we present 3D point cloud obtained on a synthetic image sequence. Figure 5 present an image pair of the sequence. The height and the width of the images are respectively 1235 pixels and 1235 pixels. The fisheye cameras cover a 180 degree field of view. The baseline is equal to 2 meters.

We use the global dynamic programming algorithm proposed in Forstmann's work in [13] to find dense correspondences. It takes advantage of epipolar constraint described in the previous section to match points along epipolar curves.
Dynamic programming is a graph-based method. We compute a graph for each couple of epipolar curves. Most of the existing works ([4], [7]) transform the spherical images to yield rectified images where epipolar lines are horizontals in order to apply a standard and well-known matching points algorithm. We do not choose to follow the same approach, because rectification generally induce a loss of information due to interpolation steps of the rectification process. We propose to scan epipolar curves by varying γ in range $[0; \pi]$.

To get curves passing through all the pictures' pixels, we could compute curves for each pixel. It is unnecessary because a pixel can be crossed by many curves, and we would compute several times same curves. To avoid a major part of unnecessary computation, we compute curves by choosing specific reference points. The solution is to compute curves for each points of the vertical line in the middle of the projected image disc. By this way, we are sure to get a maximum number of points crossed



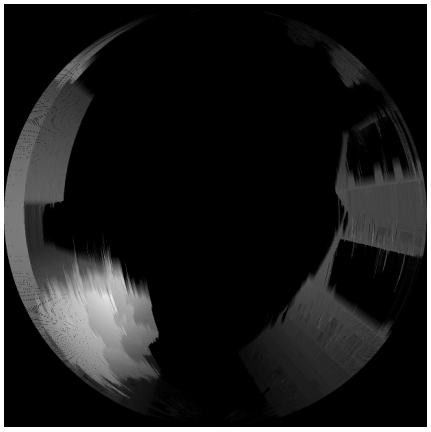
(a) Left image (back).



(b) Right image (front).

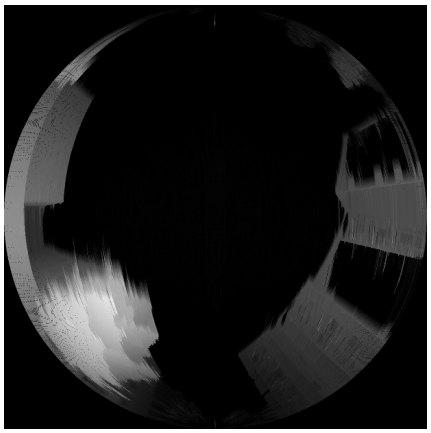
Figure 5. Examples of a pair of synthetic fisheye images. (a) and (b) are the viewed images. The red cross at position (387,233) shows a chosen point, used to calculate the epipolar curve drawn in white. In a calibrated configuration, the curve is the same on both images and passes through the same points.

by at least one curve (in practice, few points are missing but the impact is negligible), and we compute only as many curves as the number of pixels of the image disc's diameter. We can note that points near epipoles will be part of more curves than the others.
Each epipolar curve does not pass through the same number of pixels in the pictures. The longest curves are these which pass at the edge of projected image disc, and the shortest are lines between both epipoles.
For practical reasons, we choose to set the same pixel length to all indexed curves. That means that shorter epipolar curves will contain many successive times same points in their indexed version. It is not an issue if we ensure that these points correspond exactly to same angles, hence, correspondences are not altered. In fact,



(a) Longitude disparity map.

(b) Colatitude disparity map.



(c) Spherical disparity map.

Figure 6. Examples of disparity maps issued from the couple of synthetic fisheye images shown in figure 5. (a) is the longitude disparity map (φ angles), (b) the colatitude disparity map (θ angles) and (c) the spherical disparity map (α angles).

we set the pixel length of indexed curves at $\frac{\pi}{2}$ times the pixel diameter of the image disc rounded up, this value corresponds to the length of the half perimeter of the disc.

Knowing the total pixel length, we can do varying γ with a step of $\frac{\pi}{\text{indexed curve's pixel length}}$ to get indexed curves from epipolar curves (π being the total variation of γ). Then, for each obtained curves' point, we apply a specific sampling. We want to get values for positions centered in pixels, so the sampling consists of taking the entire part of computed coordinates and to add 0.5. We use these subpixel coordinates for the computation of points' associated angles (longitude φ and colatitude θ).

To illustrate the results, we compute longitude (based on φ angles), colatitude (based on θ angles) and spherical (based on α angles) normalized disparities, which is the difference between right and left angles for a given point in space (angles are drawn in figure 3). Disparity maps in figure 6 take about 50 seconds to compute with our algorithm run in a desktop computer equipped with a Core i3-2120 CPU. The proposed approach is different from Li's approach in [7], who defines and uses only the spherical disparity.

Figures 7a, 7b and 7c show the 3D point cloud we obtained by applying equations described in section II-D and by mapping the texture extracted from the images.

IV. Conclusion

This paper deals with 3D point cloud computation of urban areas using a fisheye stereovision system. After a brief state of the art of the domain, we propose an analytical study of the epipolar geometry for fisheye stereo. Matching process is based on the adaptation of the global dynamic programming algorithm proposed by Forstmann in [13]. Image pairs are not rectified and epipolar curves are not the rows of the images. We show results on synthetic images. In future works, we will focus on 3D cloud generation from fisheye images acquired in real conditions and on VRML model computation.

Acknowledgment

The authors would like to thank Dominique Gruyer (IFSTTAR LIVIC) for having provided computer-generated fisheye images.

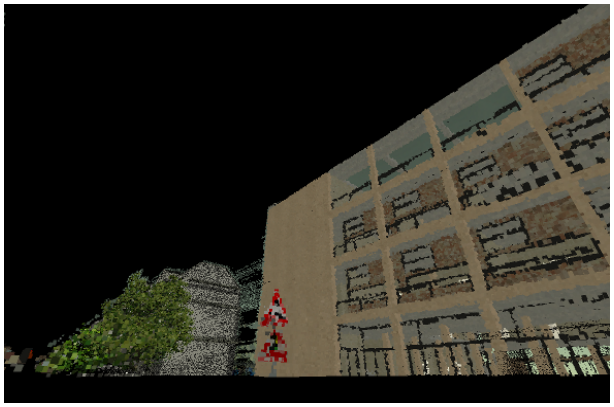
This work has been carried out as part of CAPLOC (*Combinaison de l'Analyse d'image et de la connaissance de la Propagation des signaux pour la LOCalisation*), a french project supported by PREDIT (program of research, experimentation and innovation in land transport) and funded by the french ministry MEDDTL.

References

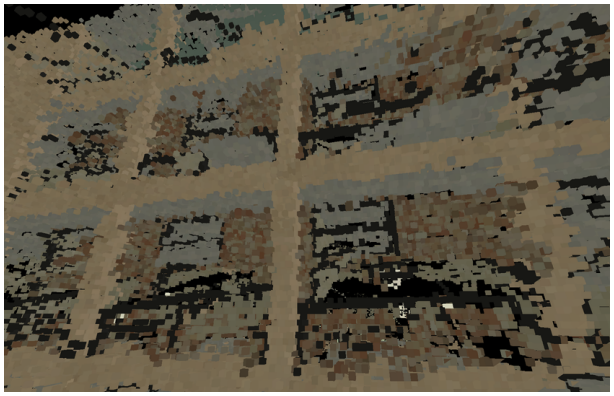
[1] J.-J. Gonzalez-Barbosa, "Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images," Ph.D. dissertation, École doctorale en informatique et télécomunications, Laboratoire d'Analyse et d'Architecture des Systèmes, Université Toulouse III, 2004.

[2] R. Kawanishi, A. Yamashita, and T. Kaneko, "Three-dimensional environment model construction from an omnidirectional image sequence," Department of Mechanical Engineering, Shizuoka University, Shizuoka, Japan, Tech. Rep., 2009.

(a) Reconstructed view of the scene shown in figure 5.



(b) Reconstructed view of the scene taken for an other position of the vehicle (an other images pair).



(c) A detail with the point of view near the building reconstructed in (b).

Figure 7. Sample of 3D point clouds. Generated points are corresponding points of the scene observed by the fisheye stereovision sensor. In original images, distant entities are smaller, i.e. they are described with less information. Consequently, these structures are represented by less points in the 3D reconstruction too, and gaps appear.

[3] N. Ragot, "Conception d'un capteur de stéréovision omnidirectionnelle : architecture, étalonnage et applications à la reconstruction de scènes 3d," Ph.D. dissertation, École Doctorale Sciences Physiques, Mathématiques et de l'Information pour l'Ingénieur, Université de Rouen, Institut de Recherche en Systèmes Electroniques EMbarqués, 2009.

[4] S. Shah and J. K. Aggarwal, "Mobile robot navigation and scene modeling using stereo fish-eye lens system," *Machine Vision and Applications*, vol. 10, pp. 159–173, 1997.

[5] B. Mičušík, D. Martinec, and T. Pajdla, "3d metric reconstruction from uncalibrated omnidirectional images," in *Asian Conference on Computer Vision*. Jeju Island, Korea: Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, 2004.

[6] S. Li, "Monitoring around a vehicle by a spherical image sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 541–550, 2006.

[7] ——, "Binocular spherical stereo," *IEEE Transactions On Iintelligent Transportation Systems*, vol. 9, pp. 589–600, 2008.

[8] P. J. Herrera, G. Pajares, M. Guijarro, J. J. Ruz, and J. M. De La Cruz, "A stereovision matching strategy for images captured with fish-eye lenses in forest environments," *Sensors*, vol. 11, pp. 1756–1783, 2011.

[9] R. Horaud and O. Monga, *Vision par ordinateur : outils fondamentaux*, 2nd ed. Éditions Hermès, 1995. [Online]. Available: http://perception.inrialpes.fr/people/Horaud/livre-hermes.html

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[11] S. Ramalingam, "Generic imaging models: Calibration and 3d reconstruction algorithms," Ph.D. dissertation, Institut National Polytechnique de Grenoble, Laboratoire GRAVIR-IMAG, Department of Computer Science, University of California, Santa Cruz, 2006.

[12] Z. Arican and P. Frossard, "Dense depth estimation from omnidirectional images," *Journal of Computer Vision and Image Understanding*, 2010.

[13] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, and A. Schmitt, "Real-time stereo by using dynamic programming," in *Computer Vision and Pattern Recognition Workshop*, 2004.