EQUISOLID FISHEYE STEREOVISION CALIBRATION AND POINT CLOUD COMPUTATION

Julien Moreau[1,2], Sébastien Ambellouis[2] and Yassine Ruichek[1]

[1]IRTES-SET, UTBM, 90010 Belfort CEDEX, France
[2]IFSTTAR, LEOST, Univ Lille Nord de France - F-59000 Lille, F-59650 Villeneuve d'Ascq, France
{julien.moreau, yassine.ruichek}@utbm.fr, sebastien.ambellouis@ifsttar.fr

KEY WORDS: fisheye stereovision, binocular, 3D reconstruction, point cloud, dynamic programming matching

ABSTRACT:

This paper deals with dense 3D point cloud computation of urban environments around a vehicle. The idea is to use two fisheye views to get 3D coordinates of the surrounding scene's points. The first contribution of this paper is the adaptation of an omnidirectional stereovision self-calibration algorithm to an equisolid fisheye projection model. The second contribution is the description of a new epipolar matching based on a scan-circle principle and a dynamic programming technique adapted for fisheye images. The method is validated using both synthetic images for which ground truth is available and real images of an urban scene.

1 INTRODUCTION

3D reconstruction of urban scenes has been highly investigated in the last few years. Different devices can be used in order to achieve this goal. It exists vision systems, based on one or two cameras.

In (Früh and Zakhor, 2004), the authors use a system combining one video camera and laser scanners. A laser scanner is used for geometrical reconstruction and the images' content is exploited for texture mapping. Other researchers address the 3D reconstruction problem from only image point of view. Some studies use multiple views acquired simultaneously by two cameras. 3D structure is computed by matching the images' content. Stereovision systems are often based on pinhole cameras, with a reduced field of view. In this case, the number of cameras have to be sufficient to cover all the area to be reconstructed. To reduce the number of needed cameras, some authors propose to use dioptric or catadioptric video sensors that offer a 360° × 180° field of view. Main steps for omnidirectional stereovision are the calibration, rectification (if any), and matching process.

Ragot (Ragot, 2009) uses a stereovision sensor based on two hyperbolic catadioptric cameras. Spherical model can be used for catadioptric and for dioptric omnidirectional sensors. He proposes to use a specific pattern made of a cylinder with lights as the pattern's features in order to strongly calibrate his sensor. He computes reconstruction with a volumetric method based on the photoconsistency of voxels. This method does not need a rectification. The difficulty with catadioptric sensors is to correctly place the mirror in the front of the lens. A fisheye lens is easier to work with. For this reason, our works are focusing on fisheye cameras.

Mičušìk in (Mičušìk and Pajdla, 2003) and (Mičušìk, 2004) is able to reconstruct some scenes in 3D with omnidirectional views. He proposes a unified self-calibration method for both catadioptric and fisheye omnidirectional system. He propose a new 9-points RANSAC based algorithm to compute the fundamental matrix. He takes into account only the equidistant projection model of the fisheye lens. He does not rectify images, and he shows the effectiveness of his calibration method by reconstructing a 3D scene with manually matched points between images in order to compute planar rectangles.

Abraham and Forstner (Abraham and Forstner, 2005) calibrate the fisheye stereo sensor with an half-box pattern. They propose an all-in-one epipolar rectification in order to compute virtual pinhole cameras, where epipolar lines are horizontal and corresponds line per line. Results are illustrated by sparse 3D point cloud.

Gehrig *et al.* (Gehrig et al., 2008) calibrate stereo fisheye sensor with a planar pattern and a method developed for standard cameras with strong distortions (Krüger et al., 2004). They use only the middle part of the image in order to get a 25° by 150° field of view, and propose a cylindrical stereo rectification of the images. 3D point clouds are computed using a classical matching method.

Li (Li, 2008) defines spherical disparity as an angle difference instead of a pixels' position difference between left and right images as it is used for pinhole cameras' images. He calibrates the sensor with an half-box pattern. Epipolar curves are transformed to horizontal lines, in order to use standard matching points algorithms.

All reconstruction methods require a calibration step that consists in estimating the transformation between the views.

The aim of the calibration is to determine a matrix of parameters linking the content of two images, acquired from a stereoscopic device or from a single moving camera. **The best model for omnidirectional images is the unit sphere model, and as we use complete circular images, we have no reason to apply a method developed for highly distorted pinhole cameras.** In the case of a stereoscope with a tiny baseline, we can use an half-box pattern to compute the fundamental matrix as described in (Li, 2008). For urban scene reconstruction with fisheye cameras, the baseline has to be large to compute an accurate 3D point cloud. In this configuration, half-box pattern based calibration process is not sufficient. In (Cannelle et al., 2012)), Cannelle proposes to place markers on well-known buildings and to compute the fundamental matrix exploiting the known geometrical and metric relation between corresponding extracted points. Some other studies propose to extract a set of unknown points in both views, then match them in order to finally estimate parameters. In (Mičušìk and Pajdla, 2003) and (Mičušìk, 2004), Mičušìk proposes an effective self-calibration solution. He proposes to adapt the 8-point algorithm presented by Hartley and Zisserman in (Hartley and Zisserman, 2004) to the case of a spherical cameras pair. His algorithm is based on a 9-point RANSAC in order to find the best matrix solution for fisheye lenses fitting the equidistant projection model.

In this paper, we propose to adapt Mičušìk's automatic fundamental matrix estimation to lenses fitting the equisolid angle model (including utilized fisheye lenses). In addition, we present a graph-based matching algorithm based on spherical epipolar geometry in order to build a dense 3D point cloud of the scene. Our method does not require rectification. The matching process is applied by estimating and sampling epipolar curves. Firstly, the proposed method is evaluated on synthetic images in order to objectively measure the quality of calibration step and 3D point cloud estimation. Secondly, we apply our method on images acquired in a real urban scene.

The experiments are done using a stereoscope system based on Sigma fisheye lenses (Nikon F Mount, 180° and equisolid angle projection according to manufacturer's specifications). Acquired image disc is about 2350 pixels diameter. Unlike the most of studies in literature, the cameras of the system are looking at the zenith point in order to see the buildings all around the car. 3D structure is estimated by matching information from both images.
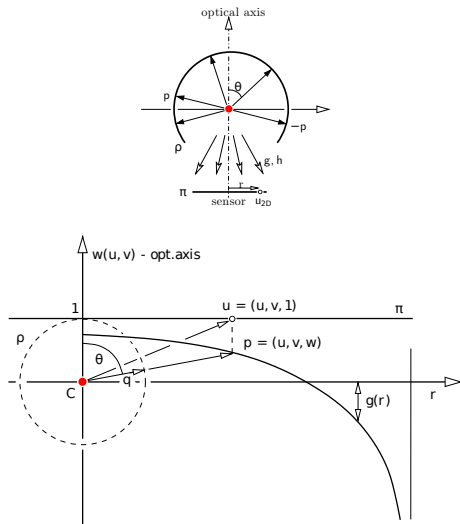
2 SPHERICAL MODEL

2.1 Generalities

Figure 1: Spherical representation of a Fisheye lens model (Mičušìk, 2004).

Fisheye projection can be efficiently described by projecting 3D points onto a unit sphere. The top part of the figure 1 represents the circular section of the sphere representing the lens. If we denote $\mathbf{u}_{2D} \binom{u}{v}$ the projection on the sensor of a real 3D point P then the coordinates u and v are linked to the angle θ of the incidence ray $\mathbf{CP}$ where C is the optical center of the lens. This relation is given by :

$$r = \sqrt{u^2 + v^2} = g(r)\tan(\theta). \qquad (1)$$

The second part of the figure 1 is a nice illustration of this mathematical relation proposed by Mičušìk. In this figure, $\mathbf{p}(u, v, w = g(r))$ (in fact $\mathbf{p}(r, w = g(r))$) is the intersection of a 3D optical ray $\mathbf{CP}$, which incidence angle is θ, with the curve described by the projection function $g(r)$.

As said by Hughes (Hughes et al., 2010), in reality, the lenses never exactly reach linear models, but they are designed to approach closely these theoretical models. He shows that linear models are not the most accurate, but may be good enough when appropriately used with a lens.

According to the used projection model, parameters are different. Complex models need more parameters. In our experiments, a simple linear model is exploited. It has only one parameter called a, and corresponds to one of the most common types of fisheye projection: the equisolid angle projection.

2.2 Equisolid Angle Model

Presented lens is the Sigma 4.5mm F2.8 EX HSM IF. According to manufacturer's specifications, it reaches equisolid angle projection.
In this paper, we propose to write the equisolid angle projection model as a linear model with one parameter a.
The equisolid angle projection is defined by the relation (Hughes et al., 2010):

$$r = 2f \sin\left(\frac{\theta}{2}\right), \qquad (2)$$

that can be written as $\theta = 2\arcsin(a.r)$ with $a = \frac{1}{2f}$ where f is the lens' focal length.
From this relation, the projection function (see eq. (1)) becomes:

$$g(r, a) = \frac{r}{\tan(2\arcsin(a.r))} \qquad (3)$$

3 CALIBRATION

3.1 9-Point Algorithm for the Equisolid Angle Projection

Fundamental matrix is computed with the Mičušìk's 9-point algorithm, initially proposed for the equidistant model in (Mičušìk, 2004). This algorithm takes into account a set of nine matched points in the left and right images to estimate an epipolar geometry. The final solution is obtained thanks to a RANSAC framework to guarantee the robustness of the results when outliers appear. RANSAC is a very popular technique to reach robust performance and information can be found in (Hartley, 1997)). We have applied the 9-point algorithm to the equisolid angle model. The degree 1 Taylor series $\tilde{g}(r, a)$ of $g(r, a)$ for a near a_0 is given by:

$$\tilde{g}(r, a) = g(r, a_0) + \frac{\partial g(r, a_0)}{\partial a}(a - a_0)$$

with

$$\frac{\partial g(r, a_0)}{\partial a} = \frac{-2r^2[\tan^2(2\arcsin(a_0 r)) + 1]}{\tan^2(2\arcsin(a_0 r))\sqrt{1 - a_0^2 r^2}}$$

For the equisolid angle model, a good initial value a_0 of the parameter a is $a_0 = \frac{1}{2f}$ where f is the lens focal length given in the manufacturer's specifications. From equation (2) and the knowledge of the maximal radius r_{max} in the image, we get $\frac{1}{2f} = \frac{\sin\left(\frac{\theta_{max}}{2}\right)}{r_{max}}$. An approximate value of θ_{max} is the half of the specified field of view given by the lens' manufacturer. For working in the unit sphere, we choose to scale $\mathbf{u}_{2D}$ distances in order to normalize r_{max}. In this case, we obtain $a_0 = \sin\left(\frac{\theta_{max}}{2}\right)$.

We add a refinement estimation step based on he Levenberg-Marquardt algorithm, as it is done for pinhole cameras in (Hartley, 1997). Initial values for the Levenberg-Marquardt algorithm are issued from the best estimation in the RANSAC. Accuracy can be improved by taking into account all the inliers instead of using an estimation based only of a set of 9 points.

3.2 Epipoles position estimation

This section deals with epipoles position estimation. The epipoles of the stereovision system are the intersection points of all epipolar circles defined on each sphere surface. Four epipoles exist: two for each sphere. As presented in figures 1 and 2, in spherical context, epipolar constraint is verified in the plane π. An epipolar circle is the projection of an epipolar line of π onto the unit sphere.
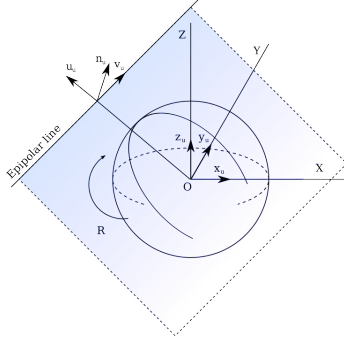

Figure 2: An epipolar circle is the projection of an epipolar line onto the unit sphere. An epipolar circle is inside the plane defined by the epipolar line and the sphere's center.

Let be x a point of the plane π_l of the left camera. The equation of its conjugate line in the plane π_r of the right camera is given by: $\mathbf{l}'(a\ b\ c)^T = \mathbf{Fx}$ where $\mathbf{F}$ is the fundamental matrix and $ax + by + c = 0$. Epipoles computation is done thanks to the relations $\mathbf{Fe} = 0$ and $\mathbf{F}^T \mathbf{e}' = 0$ where $\mathbf{F}$ is the fundamental matrix, $\mathbf{e}$ is an epipole from image 1 and $\mathbf{e}'$ an epipole from image 2.

3.3 Epipolar curves estimation

In the fisheye case, one needs arcs' equations onto the unit sphere, for projecting them into the image plane by taking into account the adapted projection function. In order to do this, the need is to project the computed lines on the unit sphere's surface, and to determine points' positions in the image plane by using the chosen spherical projection function.

To perform that, a simple way is to start from the well-known equation of an arc of a circle in the $(O\mathbf{xy})$ plane, as described in figure 2. This trick allows getting projections of infinite points of the epipolar line on the sphere. It is possible to write each original arc's point by $(\cos(\gamma)\ \sin(\gamma)\ 0)^T$, with $\gamma \in [0; 2\pi[$. The difficulty is to find the good 3D rotation that adjusts the arc with the line's projection on the sphere's surface. This rotation can be interpreted as the rotation that places the arc inside the plane defined by the epipolar line and the sphere's center.

The camera frame (O, x, y, z) can be represented by the matrix $\mathbf{M}_{original}$. One needs the normal vector $\mathbf{n}$ of the plane defined by the epipolar line and the sphere's center O, and the direction vector $\mathbf{u}$ of the line, with an orientation such as it forms an acute angle with $\mathbf{y}$, where its cosine is positive: $cos(\mathbf{y}, \mathbf{u}) > 0$ (to do that, dot product can be easily used and computed from vectors' matrices). Vectors $\mathbf{n}$ and $\mathbf{u}$ must be normalized, in order to compute the cross product for getting $\mathbf{v} = \mathbf{n} \wedge \mathbf{u}$ and a direct orthonormal frame (u, v, n), that can be represented by $\mathbf{M}_{final}$. It is known that $\mathbf{M}_{final} = \mathbf{R}.\mathbf{M}_{original}$, where $\mathbf{R}$ is the rotation matrix from the original frame to the final frame. Then, one gets $\mathbf{R} = \mathbf{M}_{final}$, that can be used on the original arc of a circle in order to define the rotated arc corresponding to the epipolar curve $\mathbf{Arc}_{sphere} = \mathbf{R}.(\cos(\gamma)\ \sin(\gamma)\ 0)$. with γ varying to browse the lens' field of view.

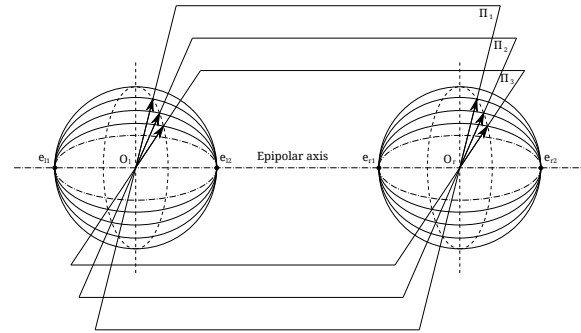## 4 EPIPOLAR CURVES COMPUTATION BY CONJUGATE CIRCLES SCANNING

Figure 3: The intersection of the epipolar plane Π_i with each unit sphere results in two epipolar circles.

The Figure 3 shows both cameras in 3D, the lenses represented by unit spheres. An epipolar plane Π_i contains the epipolar axis and intersects the unit spheres by forming great circles. These circles are the conjugate epipolar circles. They are projected as epipolar curves onto the images. A 3D point cloud is estimated thanks to a matching process applied along the conjugate epipolar circles as it is done along the epipolar lines for a pinhole stereovision system.

Two conjugate circles of a plane Π_i are defined by the unit radius vector resulting from the intersection of the three following geometrical objects: the plane Π_i, two parallel planes containing respectively the left and the right sphere center and the unit sphere. A simple solution is to choose two parallel planes perpendicular to the epipolar axis and to scan the unit radius vector inside both planes.
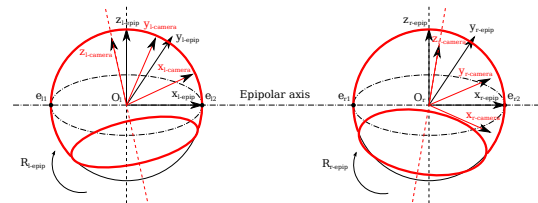

Figure 4: Illustration of epipolar and camera reference frames in both cameras.

This operation is done in the epipolar reference frame of each camera. Both epipolar reference frame are differing from a translation along the X_{epip} axis and can be estimated by applying rotations to left and right camera reference frame (cf. Figure 4). These rotations are calculated from the epipolar axes X_{l-epip} and X_{r-epip} and from Fundamental matrix by following these steps:

- Define the circle $\mathcal{C}_l$ of the unit left sphere, orthogonal to $X_{l\text{-epip}}$.

- Let be q_l the intersection between $\mathcal{C}_l$ and the arc of unit circle in the plane $(O_l xy)$ of the camera reference frame and centered in O_l.

- Compute the projection u_l of q_l onto the plan π_l (cf. Figure 1).

- Apply the mathematical relation $\mathbf{l_r}' = \mathbf{Fu_l}$ to calculate the epipolar line of the point u_l in the plane π_r (cf. Figure 1).

- Project $\mathbf{l_r}'$ on the right unit sphere.

- Define the circle $\mathcal{C}_r$ of the unit right sphere, orthogonal to $X_{\text{r-epip}}$.

- Let be $Z_{\text{r-epip}}$ defined by the point O_r and the intersection between the $\mathcal{C}_r$ and the projection of $\mathbf{l_r}'$.

- Compute $Y_{\text{l-epip}}$ and $Y_{r-\text{epip}}$ by applying the cross product to their respective epipolar X and Z axis.

Let be Π_i a set of epipolar plane obtained by moving along the circles $\mathcal{C}_l$ and $\mathcal{C}_r$. Each plan Π_i intersects both unit sphere to produce two conjugate circles that are finally mapped into two conjugate image curve. A 3D point cloud is computed by applying a matching process along each conjugate curves. In the the following section, we show some results of the calibration steps and of 2D point clouds obtained thanks to an algorithm similar from (Forstmann et al., 2004) and that we adapt to the typology of our conjugate curves. More information can be found in (Moreau et al., 2012).

5 EXPERIMENTAL RESULTS

5.1 Calibration results on synthetic images
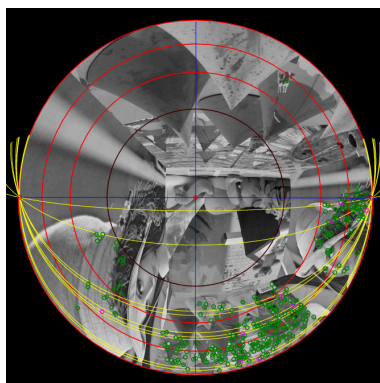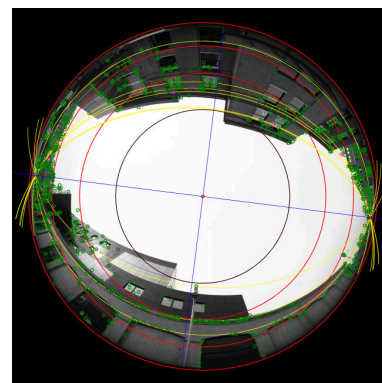

(a) Left calibration image
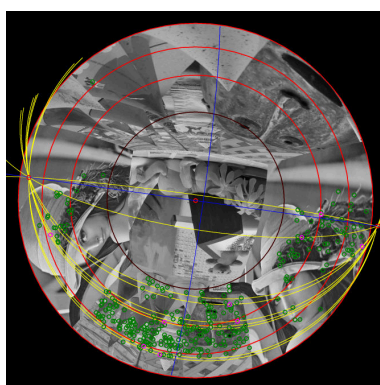
(b) Right calibration image

Figure 5: Illustration of the calibration result for a couple of synthetic fisheye images. Red large circle represents the limit field of view of the lens. The red point in the middle is its center. Opposite red points in left and right sides are the epipoles' projections on the image. Blue line passing through the image's center and both epipoles is the epipolar axis. Green and magenta points are fitting the model (inliers). 9 magenta points are those used to reach the better estimation with RANSAC. Yellow curves are epipolar curves. Perpendicular blue arc represents the scan-arc.

Synthetic images are produced through a lens that fits the equisolid angle model. For the calibration, we consider it has a 180° field of view, but in reality we generated images with a 181.8° field of view to get a 10% difference. Left and right cameras look at the Z direction, but right camera has some additional rotations along X (-8°), Y (-6°) and Z (-7°) axes to simulate the placement inaccuracy of a real case. We propose to measure the estimated field of view and parameter a, and the camera's rotation estimation. To do this, results are evaluated for 200 iterations. Figure 5 presents an example of calibration images.

The field of view and the parameter a are linked by the relation $a = \sin(\frac{\theta}{2})$. For $\theta = 180°$, $a_0 = a_{180} = 0.7071$. For $\theta = 181.5°$, $a_{181.5} = 0.7126$. The average value of the a values is $\hat{a} = 0.7130$. The difference with ground truth is 3.268×10^{-4}. The variance of the a estimations is 4.442×10^{-6}.

The average value of each rotation estimation are respectively $\hat{R}_X = -7.986°$, $\hat{R}_Y = -6.034°$ and $\hat{R}_Z = -7.0047°$. The difference between the averages values and the ground truths are respectively $e\hat{r}r_X = 0.0138°$, $e\hat{r}r_Y = 0.0343°$ and $e\hat{r}r_Z = 0.0047°$. The variance of each rotation estimations are respectively $var_X = 0.0700$, $var_Y = 0.0182$ and $var_Z = 0.0402$.


(a) Left calibration image.
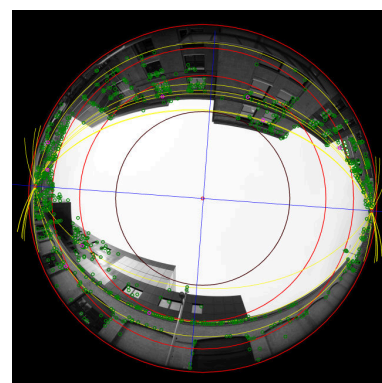
(b) Right calibration image.

Figure 6: Illustration of the calibration for a couple of real images taken with the Sigma lens.

The figure 6 presents an example of calibration images for a real scene. The results are obtained for 200 iterations of the calibration process for a couple of real images (cf. figure 6). The variance of a estimation is 1.138×10^{-5}. The average of the estimated a value is $\hat{a} = 0.7309$ and initial a_0 value is 0.7071. The difference between $\hat{a}$ and a_0 is 0.0238. According to the $\hat{a}$ value and the equisolid angle projection model, we can conclude that the real sigma's lens field of view is about 188°.
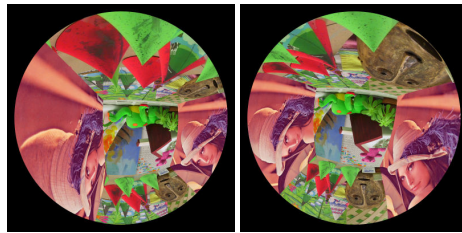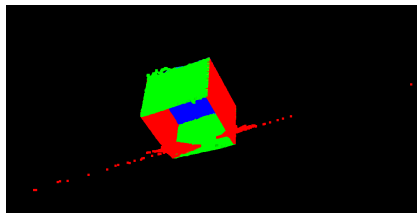
(a) Textured fisheye views.
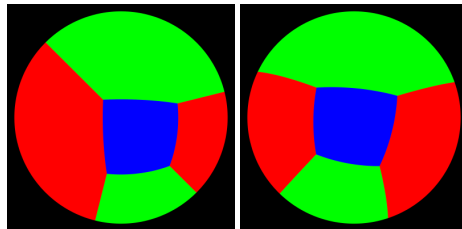
(b) Colored fisheye views.
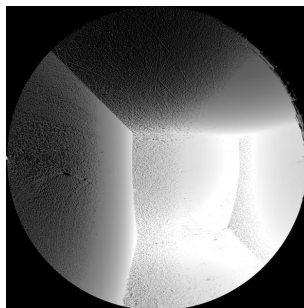
(c) Distances map.

(d) Point cloud.

Figure 7: Fisheye images inside the cube for evaluation. 3D reconstruction of the cube. Distances in distances map are shown for a limit of 5 units (white). Dark points in distance maps are non-matched points. They are numerous because of the large difference of the area taken by the faces from an image to the other one. The cube's point cloud has not been filtered.

5.2 Reconstruction Evaluation

We propose to evaluate the reconstruction with a couple of computed images of a textured cube. Images size is of 2000×2000 in order to approach real images' dimensions. The images contain a cube of 5×5 faces, an edge at the origin and the opposite one at the position (5,5,5). Faces are textured in order to be able to find feature points for the calibration and to apply the matching point process. Three different textures are used, the same on each opposite face. A second couple of synthetic images is used, with the same cube and textures replaced by colors red, green and blue. Red faces are located across X axis, green faces across Y axis and blue faces across Z axis. By using these color information in the computed point cloud, we can know for each point to which face it belongs. We can compute a distance error for each point to its face.

Left camera is at the position (1,1,1), oriented in the direction of the Z axis. Right camera is at the position (1,3,1), oriented the same way but with an additional rotation in order to simulate the position error we would get in a real case. For this test, we do not add a random translation in order to compute an exact metric reconstruction of the cube and to measure efficiently error distances. These cameras respect the equisolid angle model and have a field of view near but not equal to 180° in order to simulate real cameras with not very accurate manufacturer's specifications.

Textured fisheye images, distances map and 3D reconstruction are shown in figure 7. Calibration images are given in figure 5. Cameras' location near one of the corners adds a difficulty to the matching process, especially for the face drawn in the top of fisheye images that is very close to the sensors. Cloud point is made of 2.134.143 points. Point's error is defined as the distance between the point and its parent face. Average error of cube's points is of 0.016 units and standard deviation is of 0.290 units. Standard deviation is not so good because of the set of bad points mostly located far along the epipolar axis.

(a) Fisheye views.
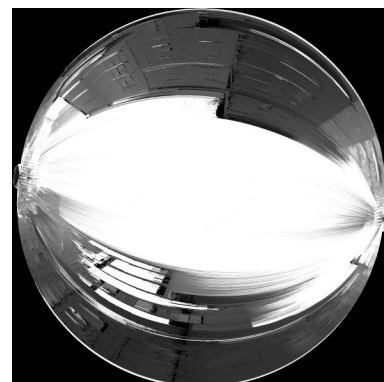
(b) Distances map.
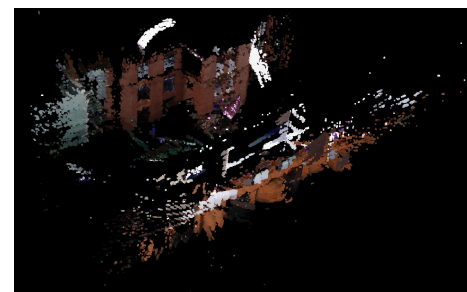
(c) Point cloud.

Figure 8: 3D reconstruction of a real scene. Matching process took 183 seconds. Distances in distances map are shown for a limit of 75 meters (white). Point cloud has been filtered to reduce noise.

Presented real scene is the same as for the calibration in figure 6. In figure 8, distances map shows near points in dark gray and far ones in light gray. The top part of the images is well recon-

structed. The bottom part presents holes and bad matched points. This is due to the lack of textures on bottom building. Some transition areas between buildings and sky show disparity propagation in the sky. This is an effect of the continuity constraint in the dynamic programming graph.

6 CONCLUSIONS

We have shown that it is possible to automatically calibrate fisheye lenses with regards to different fisheye projection models by using linear models. We propose to exploit fisheye epipolar geometry for matching by scanning corresponding epipolar arcs. Fundamental matrix estimation shows a high stability, that means automatic estimation gives good results. Reconstruction process, i.e. calibration and 3D point cloud computation, is validated by the cube's test. In future works, we plan to improve the matching process by taking into account textureless areas in the images. In addition, we will mesh and merge models in order to reconstruct whole streets.

ACKNOWLEDGMENTS

REFERENCES

Abraham, S. and Forstner, W., 2005. Fish-eye-stereo calibration and epipolar rectification. ISPRS Journal of Photogrammetry & Remote Sensing 59, pp. 278–288.

Cannelle, B., Paparoditis, N., Pierrot-Deseilligny, M. and Papelard, J.-P., 2012. Off-line vs. on-line calibration of a panoramic-based mobile mapping system. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. I-3, Melbourne, Australia.

Forstmann, S., Kanou, Y., Ohya, J., Thuering, S. and Schmitt, A., 2004. Real-time stereo by using dynamic programming. In: Computer Vision and Pattern Recognition Workshop.

Früh, C. and Zakhor, A., 2004. An automated method for large-scale, ground-based city model acquisition. International Journal of Computer Vision 60(1), pp. 5–24.

Gehrig, S. K., Rabe, C. and Krger, L., 2008. 6d vision goes fisheye for intersection assistance. In: Canadian Conference on Computer and Robot Vision, pp. 34–41.

Hartley, R., 1997. In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(6), pp. 580 –593.

Hartley, R. and Zisserman, A., 2004. Multiple View Geometry in Computer Vision. second edn, Cambridge University Press.

Hughes, C., Denny, P., Jones, E. and Glavin, M., 2010. Accuracy of fish-eye lens models. Applied optics 49, pp. 3338–3347.

Krüger, L., Wöhler, C., Würz-wessel, E. and Stein, F., 2004. Infactory calibration of multiocular camera systems. In: SPIE Photonics Europe (Optical Metrology in Production Engineering, pp. 126–137.

Li, S., 2008. Binocular spherical stereo. IEEE Transactions On Iintelligent Transportation Systems 9, pp. 589–600.

Mičušìk, B., 2004. Two-View Geometry of Omnidirectional Cameras. PhD thesis, Czech Technical University in Prague.

Mičušìk, B. and Pajdla, T., 2003. Estimation of omnidirectional camera model from epipolar geometry. In: Conference on Computer Vision and Pattern Recognition.

Moreau, J., Ambellouis, S. and Ruichek, Y., 2012. 3d reconstruction of urban environments based on fisheye stereovision. In: 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), pp. 36–41.

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