2000

Omni-Directional Structure From Motion

Peng Chang
Carnegie Mellon University

Martial Hebert
Carnegie Mellon University

Published In
Proceedings of the 2000 IEEE Workshop on Omnidirectional Vision, 127-133.

Omni-directional Structure From Motion

Peng Chang Martial Hebert
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
[peng,hebert]@ri.cmu.edu

Abstract

In this paper, we describe our work on a robot navigation system using an omnidirectional camera as the primary sensor. An omnidirectional structure from motion algorithm is presented with its uncertainty analyzed. We evaluate the omnidirectional SFM on both synthetic data and on real image sequences. Comparison with the conventional camera is made and we show that in certain situations the omnidirectional SFM gives better results than the conventional one by taking advantage of its larger field of view.

1. Introduction

To extract the camera motion and scene structure from image sequences is known as "Structure from Motion (SFM)", which has been extensively studied by computer vision researchers. Although the underlying geometry is well studied, most of the research focuses on using conventional cameras. In this paper, we present an omnidirectional camera based SFM algorithm and applied it to our visual servoing navigation tasks. In Section 4, we compare the geometric property of the omni-directional camera and conventional camera with experiment results. We show that in some situations the omnidirectional SFM can achieve better results than the conventional one even though the omnidirectional image has a lower resolution.

Generally speaking, most existing SFM algorithms can be divided into two categories according to the criterion functions that they seek to optimize. The first one is to parameterize both the camera motion and scene structure, then estimate them by minimizing the criteria based on the distances of the image projections (as in [9] and [1]). The other trend is to develop the criteria based on epipolar constrains. See [12] and [11] for a complete review. The observation is that the image projections are related by the fundamental matrix (essential matrix, for calibrated cameras) which is only a function of the camera motion. Since the later tries to recover fewer parameters given the same data, presumably it is easier and that is the approach we adopt here.

The epipolar constrains can take either a differential form or a normal form. J. Gluckman and S. Nayar [5] addressed the problem of recovering the ego motion from an omnidirectional camera using the differential form epipolar constrain. As it is noted in [8],there is a fundamental tradeoff between the differential form and normal form. With the differential form, motion between two frames is assumed small so that the differential form epipolar constrain still holds, thus the correspondence problem is easier. On the other hand, with the normal form epipolar constrain, the motion can be arbitrarily large which poses a difficulty in finding the correspondence, but large motion also means large flow magnitude, which is very desirable for the optimization to be robust against the measurement noise in optic flow. Considering a robot moving at normal speed with a normal frame grabber, the differential form epipolar constrain usually does not hold in real situations. Therefore we developed our omnidirectional SFM based on the normal form epipolar constrain. To solve the correspondence problem, we can use a multiple target tracking process which finds correspondences over time.

2. Two-step omnidirectional SFM Algorithm

We present the algorithm by first introducing the epipolar constrain for an omnidirectional camera. In [10], the epipolar geometry is studied for panoramic cameras with hyperbolic mirrors. In this paper, we develop the epipolar constrain for catadioptric cameras which is simpler to compute due to its special geometric property.

If P is a point in the scene, and m_1, m_2 are its projections on two images, then the epipolar constrain holds:

$$m_1^T F m_2 = 0$$

and F is the fundamental matrix. If the camera is calibrated, then m_1 and m_2 are normalized image coordinates, and F

0-7695-0704-2/00 $10.00 (c) 2000 IEEE

is the essential matrix and can be decomposed as $[t]_\times R$ where $[t]_\times$ and R encode the translation and rotation of the camera motion.

Now let us look at the epipolar constrain for omnidirectional cameras as shown in Figure 1. Here we take a catadioptric omnicam with a parabola mirror as an example. See Nayar [7] for more details on this type of omnicam.

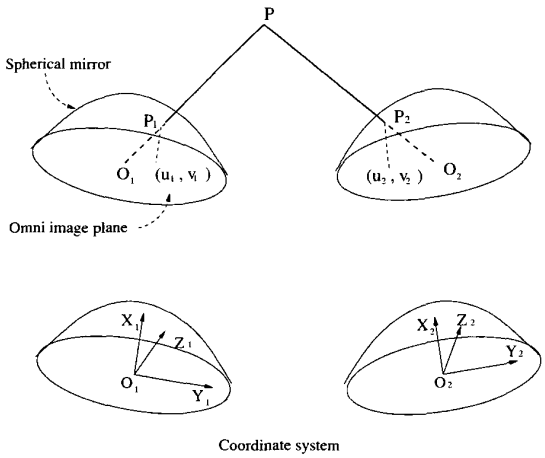


Figure 1. epipolar geometry of omnidirectional camera

P is a point in the scene. P_1 and P_2 are its projections on two omnidirectional mirrors. (u_1, v_1) and (u_2, v_2) are its corresponding image coordinates on the two omnidirectional images. We construct the local coordinate systems $X_1Y_1Z_1$ and $X_2Y_2Z_2$ as shown in Figure 1 then the local coordinates of P_1 and P_2 can be computed as in equation 1 according to [7]. For other types of omnicam with different mirror shapes, the appropriate projection equation should be used.

$$P_i = \begin{bmatrix} h^2 - u_i^2 - v_i^2 \\ u_i \\ v_i \end{bmatrix} \quad i = 1, 2 \tag{1}$$

Observing that P, P_1, P_2, O_1 and O_2 are coplanar, then

$$\overline{O_2O_1} \times \overline{O_2P_1} \cdot \overline{O_2P_2} = 0$$

which is equivalent to

$$O_1^2 \times P_1^2 \cdot P_2 = 0 \tag{2}$$

where O_1^2 and P_1^2 are the coordinates of points O_1 and P_1 in coordinate system $X_2Y_2Z_2$. Assuming the rigid motion between $X_1Y_1Z_1$ and $X_2Y_2Z_2$ can be described by rotation matrix R and translation vector t, then

$$O_1^2 = R \cdot O_1 + t = t$$

$$P_1^2 = R \cdot P_1 + t$$

Substitute into equation 2, we get

$$P_2^T E P_1 = 0 \tag{3}$$

where $E = [t]_\times R$ is the essential matrix encoding the motion parameters. Given multiple correspondence points, E can be estimated by minimizing the epipolar errors. In [12] Zhang gives a comprehensive review of estimating E for conventional cameras. Since we are more interested in the motion parameters than E itself, we choose to parameterize E with the underlying motion parameters $(t_x, t_y, t_z, q_0, ..., q_3)$, and then recover those parameters directly by minimizing the epipolar errors. (t_x, t_y, t_z) is the translation vector and $(q_0, ..., q_3)$ is the quaternion representing rotation matrix R. Since it naturally enforces the rank constrain of E, it achieves better result than simply estimating E.

We present the following two-step omnidirectional SFM algorithm. Suppose we have a set of point correspondences: $(m_i, m_i'), i = 1, ..., n$, where $n \geq 7$, where $m_i = [u_i, v_i]$ are the image coordinates in the omni image. We can can compute P_i and P_i' according to equation 1. Like what is usually done for a conventional camera, we construct the epipolar constrains for each correspondence pair and stack them together into one linear system.

$$Uf = 0$$

where

$$U = [\mathbf{u}_1, ..., \mathbf{u}_n]^T$$

and $\mathbf{u}_i$ and f are vectors constructed by stacking columns of matrices $\mathbf{P_i}$ and E, respectively.

$$\mathbf{P}_i = P_i P_i'^T$$

Step 1: estimate E with linear least square by solving

$$Uf = 0$$

where U is a $n \times 9$ matrix and f is 9×1 vector containing the 9 elements of E.

Step 2: parameterize f as $f(t_x, ..., q_3)$. Use the estimated f in step 1 as a starting point, perform Levenberg-Marquard to solve the following equation:

$$Uf(\mathbf{t}, \mathbf{q}) = \mathbf{0} \tag{4}$$

In practice, to avoid the trivial solution, we actually minimize

$$Uf(\mathbf{t}/||\mathbf{t}||, \mathbf{q}/||\mathbf{q}||)$$

To verify the algorithm, we first test it with synthetic data. A $100m \times 40m \times 15m$ rectangle area is constructed.

We simulate the robot moves in a predefined 3D curve as shown in Figure 2. The circles represent 3D points (50 in total) randomly selected around this rectangle area and the solid curve is the trace of the robot which moves at a speed of 0.5m/s for 20 seconds. We simulate using an omnidirectional camera to estimate the motion of the robot every one second. We simulate the tracking uncertainty by adding a Gaussian noise to the computed optic flow. For the results shown in Figure 3, the Gaussian noise is chosen to be $N(0,1)$. Figure 3 shows the seven motion parameters estimated by the linear step and nonlinear step along the path. As we can observe, the linear step follows the true values, while the nonlinear step achieves more accurate results in presence of noise.

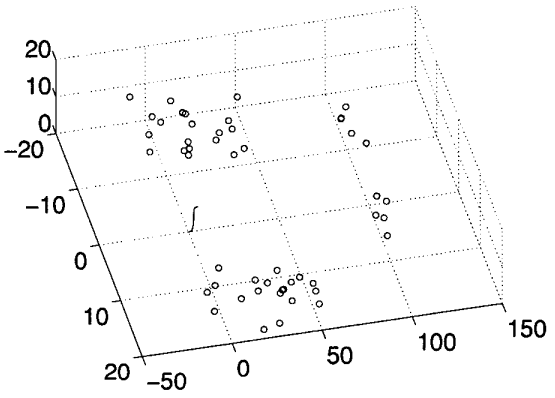


Figure 2. trace of a robot moving in the scene

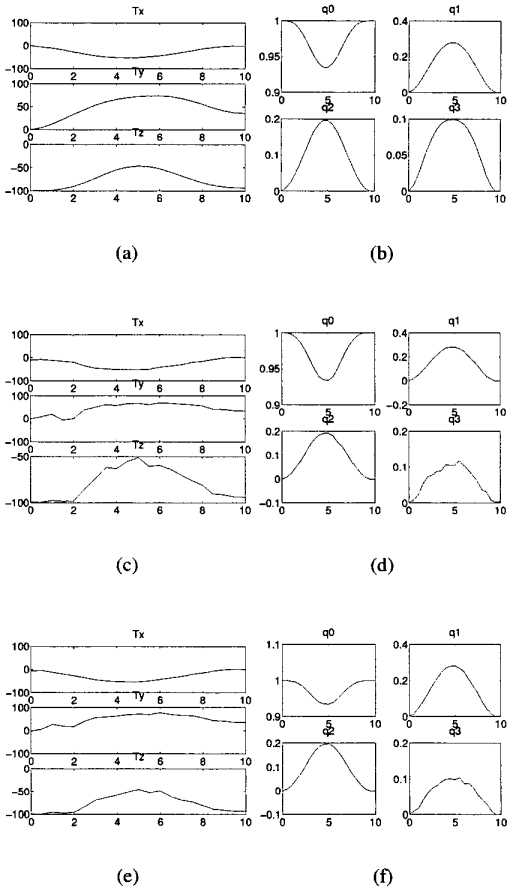


Figure 3. Results of Omnidirectional SFM (a)-(b) is the true motion parameters. (c)-(d) show the parameters estimated by linear step. (e)-(f) show the parameters estimated by nonlinear step.

3. Uncertainty analysis

Since the correspondence measurements are always corrupted with noise, we want to model how the uncertainty of the correspondences propagates into the estimated motion parameters. Following the implicit theorem as in [12], it is possible to derive an analytic solution to compute the uncertainty of the estimation. We summarize the results as in following.

Solving equation 4 with LM is equivalent to minimize the following criterion function:

$$C(\mathbf{x},\mathbf{p}) = \|U(\mathbf{x})f(\mathbf{t},\mathbf{p})\| \tag{5}$$

where $\mathbf{x}$ is the correspondence measurement and $\mathbf{p}$, the motion parameters $(\mathbf{t},\mathbf{q})$. We can compute the covariance matrix of $\mathbf{p}$ by equation 6. Further details can be found in [12].

$$\Lambda_{\mathbf{p}} = \mathbf{H}^{-1}\frac{\partial\Phi}{\partial\mathbf{x}}\Lambda_{\mathbf{x}}\left(\frac{\partial\Phi}{\partial\mathbf{x}}\right)^{T}\mathbf{H}^{-T} \tag{6}$$

where

$$\Phi = 2\left(\frac{\partial f}{\partial\mathbf{p}}\right)^{T}U(\mathbf{x})^{T}U(\mathbf{x})f(\mathbf{p})$$

and

$$\mathbf{H} = \frac{\partial\Phi}{\partial\mathbf{p}}$$

An approximation to the covariance matrix is also presented by Zhang [12]

$$\Lambda_{\mathbf{p}} = \frac{2S}{n-k}\mathbf{H}^{-T} \tag{7}$$

where S is the value of the criterion function C at the minimum, k is the number of parameters, i.e. 7 here, and

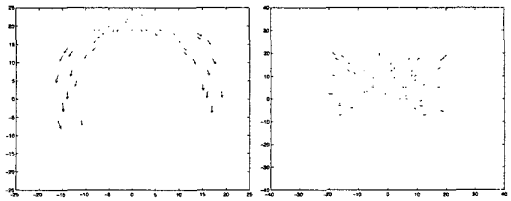
n is the number of correspondences. This approximation shows that the uncertainty of the estimated motion parameters ($\mathbf{p}$) depends on the condition of the Hessian $\mathbf{H}$.

The above analytic solutions provide ways to estimate the uncertainty of the estimated parameters, but they are only valid around the true minimum. The results would be meaningless if the optimization ends up at some wrong local minimums. To cope with this problem we use Bootstrap to estimate the uncertainty in our synthetic experiments. Bootstrap is a statistical tool for the computation of covariance matrices, bias and confidence regions for specified estimators. It is a more general alternative to the traditional covariance propagation technique. The book of Efron [3] contains an good introduction. Bootstrap does not require the analytic solution but is data driven and computationally more expensive. It is more appealing here since the covariance analysis is only valid around the true minimum while with bootstrap we statistically evaluate cases when the minimum is not properly found. Bootstrap is also simple to use, we assume Gaussian noise in the correspondence data and create a bootstrap set. We run the two-step SFM for many times (200) and compute the covariance matrix of the recovered motion parameters as its uncertainty measurement. We use bootstrap to generate the result in next sections.

4. Comparison with conventional camera

Obviously omnidirectional camera sacrifices image resolution in exchange for a wider field of view. One obvious advantage of the 360 degree horizontal field of view is that we can track feature points for longer distance with less constrain to the robot motion. Furthermore, one would wonder if this larger field of view would help solve the SFM problem too.

From equation 6 we can see that the actual uncertainty of the SFM results depends on both the configuration of the scene points and camera motion itself. Assuming **i.i.d.** noise in the flow measurement, better result would be expected when the flow magnitude is larger due to the increased SNR. If we perform SFM by looking at objects at short range, the conventional camera is no doubt preferred to the omnicam since it generates larger flow with its higher resolution. But when we look at objects far away, SFM becomes more difficult, since either the flow magnitude is very small or the bas-relief ambiguity arises due to the weak perspective effect. The omnidirectional camera bypasses this difficulty by looking at objects at its sides. The actual range beyond which an omnidirectional camera would outperform a conventional camera in SFM depends on the resolution of the images and actual environment. In the experimental setting as in Figure 2, we find the omnidirectional SFM is better when the objects in front are more than 50 meters away. Figure 4 shows two flow fields generated by a conventional and an omnidirectional camera respectively.



(a) (b)

Figure 4. Two flow fields generated by omnidirectional camera (a) and conventional camera (b). Both move five meters forward. The omnicam looks at points described in Figure 2 while the conventional camera looks at points fifty to sixty meters away.

5. Experiment results

In this section, we present some results on the uncertainty of the motion estimation with respect to flow magnitude and noise level in flow data. The basic observation is that motion estimation generally improves when flow magnitude increases and that the motion estimation degrades when noise increases in the flow magnitude. We also compare results from the omnidirectional SFM with the conventional SFM for a specific situation to verify our discussion in section 4.

Experiment 1: We simulate the robot with an omnidirectional camera moving 10 meters forward in the rectangle area which is described in section 2. We assume we can keep tracking all the points along the path and we perform omnidirectional SFM every half meter. The flow data is corrupted with a Gaussian noise, whose variance is set to 1 in the following experiment. We compare the recovered parameters with the ground truth. The error metric for (t_x, t_y, t_z) is chosen to be the angle between the true translation direction and recovered translation direction. The error metric for $(q_0, ..., q_3)$ is the angle between the true quaternion and the recovered one. Absolute mean error and standard deviation are computed for both translation and quaternion after running the algorithm for 200 times. Figure 5 shows that motion parameters and reconstructed structure improve along the path, which is what we expect to see. In figure 5(c) 5(d), the linear method becomes less accurate on rotation estimation along the path.

We also show that the reconstruction improves along the path in figure 6. All the reconstructions are up to a scale factor. The reconstructed points are inaccurate when the

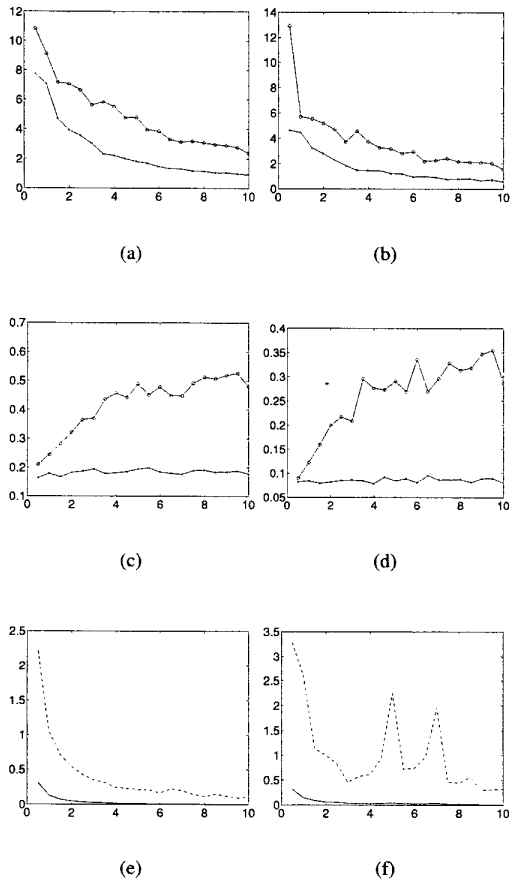


Figure 5. "o" represents linear step and "x" represents LM optimization. X axis is the distance along the path (a) absolute mean error of translation direction in degree. (b) standard deviation of translation direction in degree. (c) absolute mean error of quaternion direction in degree. (d) standard deviation of quaternion direction in degree. (e) mean reconstruction error (meter), the dash line is the maximum value and the solid line is the mean value (f) standard deviation of reconstruction error (meter)

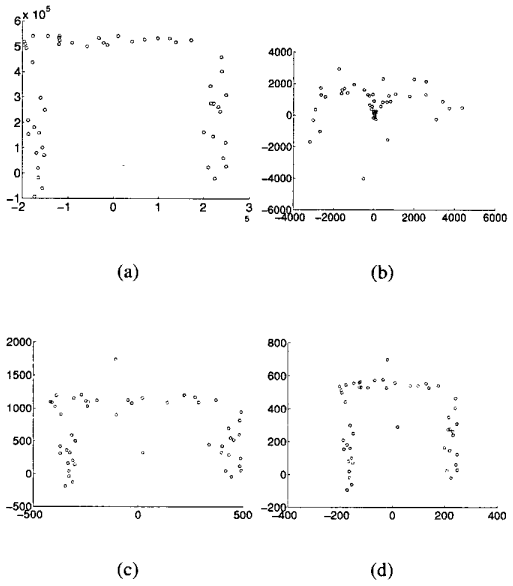


Figure 6. ground projection of the scene points (a) true scene (b) (c) (d) recovered scene points at 0.5, 5, 10 meter along the path

translational distance is small, as shown in figure 6(b) and 6(c).

Now We compare the performance of the omnidirectional SFM with the conventional SFM using the experimental setting in Figure 2. Note that the objects in front are 100 meters away and we only move the robot 0.25 meter forward. When the translational distance is small, it is very difficult for SFM with the conventional camera, and it is less difficult for the omnidirectional camera which can look at its sides. Figure 7 shows that in this case the rotation (quaternion) estimate is comparable for both cameras while the translation estimated by the conventional camera is far off (50 degrees with the N(0,1) noise).

Experiment 3: We also evaluate our algorithm on real images taken with a cadaoptric omnidirectional camera in the outdoor environment. Figure 8 shows three frames of omnidirectional images taken at different positions along a path. The robot was moving forward between two buildings. The figure 8(a) is taken at the start point. Figure 8(b) and 8(c) are taken at 0.5 meters and 2.5 meters respectively. The circles on the images are feature points manually selected and tracked. The dark points are feature points that are very far away. Figure 8(d) and 8(e) show the flow fields. We verified that the translation and rotation are properly recovered. In figure 9 we show the reconstructed scene points at 0.5 meters and 2.5 meters respectively.

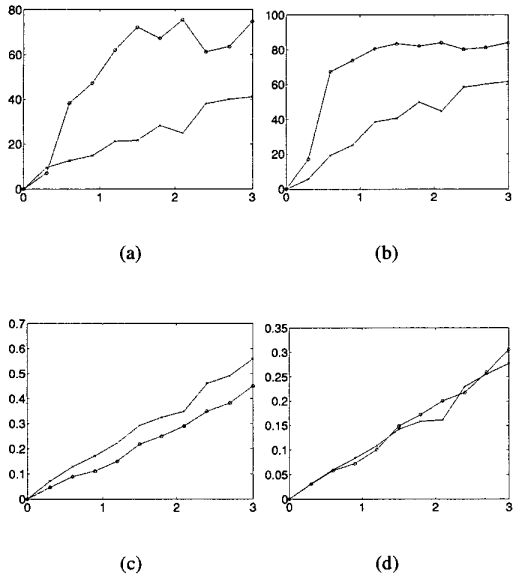


Figure 7. Comparison between SFMs with an omnicam and a conventional camera when objects are far away. "x" indicates the omnicam and "o" indicates the conventional camera. X axis is the variance of the noise and Y is the actual error in degree. (a) absolute mean error of translation (b) standard deviation of translation (c) absolute mean error of quaternion (d) standard deviation of quaternion estimation

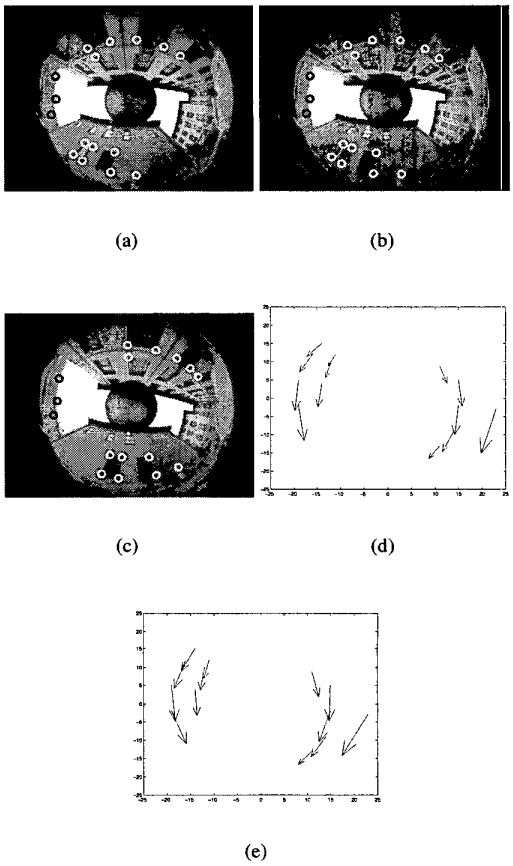


Figure 8. (a) first frame (b) frame grabbed after driving 0.5 meters (c) frame at 2.5 meters (d) omnidirectional flow at 0.5 meters (e) omnidirectional flow at 2.5 meters

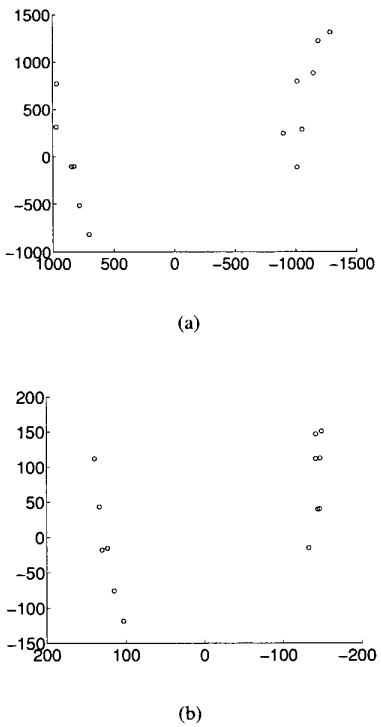


Figure 9. Reconstructed scene points (up to a scale) (a) reconstruction after driving 0.5 meters (b) reconstruction after driving 2.5 meters

6. Conclusions

In this paper, we describe a SFM algorithm based on the omnidirectional camera. Our experiments show that there are certainly some advantages of the omnidirectional camera that can be explored in a robot navigation task.

References

[1] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transaction on Pattern Recognition and Machine Intellegence*, 17(6), 1995.

[2] P. Chang and M. Hebert. Omnidirectional visual servoing for human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[3] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[4] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[5] J. Gluckman and S. Nayar. Ego-motion anad omnidirectional cameras. In *International Conference on Computer Vision*, 1998.

[6] S. Kang and R. Szeliski. 3-d scene data recovery using omnidirectional multibaseline stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–370, 1996.

[7] S. Nayar. Catadioptric omnidirectional camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, 1997.

[8] J. Oliensis. A critique of structure from motion algorithms. Technical report, NEC research institute, 1998.

[9] R. Szeliski and S. Kang. Recovering 3d shape and motion from from image streams using monlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.

[10] T. P. T. Svoboda and V. Hlavac. Epipolar geometry for panoramic cameras. In *European Conference on Computer Vision*, 1998.

[11] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal on Computer Vision*, 24(3):271–300, 1997.

[12] Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *International Journal on Computer Vision*, 27(2):161–195, 1998.