MULTIPLE VIEW GEOMETRY FOR MIXED DIMENSIONAL CAMERAS

Kazuki Kozuka and Jun Sato
Department of Computer Science and Engineering, Nagoya Institute of Technology
Nagoya, 466-8555, Japan

Keywords: Multiple view geometry, stereo vision, structure from motion, non-homogeneous cameras.

Abstract: In this paper, we analyze the multiple view geometry under the case where various dimensional imaging sensors are used together. Although the multiple view geometry has been studied extensively and extended for more general situations, all the existing multiple view geometries assume that the scene is observed by the same dimensional imaging sensors, such as 2D cameras. In this paper, we show that there exist multilinear constraints on image coordinates, even if the dimensions of camera images are different each other. The new multilinear constraints can be used for describing the geometric relationships between 1D line sensors, 2D cameras, 3D range sensors etc., and for calibrating mixed sensor systems.

1 INTRODUCTION

The multiple view geometry is very important for describing the relationship between images taken from multiple cameras and for recovering 3D geometry from images (Hartley and Zisserman, 2000; Faugeras and Luong, 2001). The traditional multiple view geometry assumes the projection from the 3D space to 2D images (Faugeras and Keriven, 1995; Triggs, 1995; Heyden, 1998; Hartley and Zisserman, 2000). As a result, the traditional multiple view geometry is limited for describing the case, where enough number of corresponding points are visible from a static configuration of multiple cameras. Recently, some efforts for extending the multiple view geometry for more general point-camera configurations have been made (Hartley and Schaffalitzky, 2004; Shashua and Wolf, 2000; Sturm, 2005; Wexler and Shashua, 2000; Wolf and Shashua, 2001). Wolf et al. (Wolf and Shashua, 2001) studied the multiple view geometry on the projections from N dimensional space to 2D images and showed that it can be used for describing the relationship of multiple views obtained from moving cameras and moving points with constant speed. Hayakawa et al. (Hayakawa and Sato, 2006) showed that it is possible to define multilinear relationships for general non-rigid motions.

However, these existing research works assume that the scene is observed by homogeneous multiple cameras, that is the dimensions of images of multiple cameras are the same. Since it is sometimes important to combine different type of sensors, such as line sensors, 3D range sensors and cameras, the assumption of homogeneous cameras is the big disadvantage of the existing multiple view geometry. On the other hand, the traditional multiple view geometry has been extended gradually for non-homogeneous sensors. Sturm (Sturm, 2002) analyzed a method for mixing catadioptric cameras and perspective cameras, and showed that the multilinear relationship between standard perspective cameras and catadioptric cameras can be described by non-square tensors. Thirthala et al. (Thirthala and Pollefeys, 2005) analyzed the trilinear relationship between a perspective camera and two 1D radial cameras for describing the relationship between standarad cameras and catadioptric cameras. Although these results show some posibility of the use of non-homogeneous sensors, these are limited for specific camera combinations.

Thus, we in this paper analyze the multiple view geometry for general mixed dimensional cameras in various dimensional space, and show the multilinear relationships for various combinations of non-homogeneous cameras. We also show that these mul-

Kozuka K. and Sato J. (2008).
MULTIPLE VIEW GEOMETRY FOR MIXED DIMENSIONAL CAMERAS.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 5-12
DOI: 10.5220/0001072500050012

tilinear relationships can be used for calibrating multiple dimensional cameras and for transfering corresponding points to different dimensional cameras.

2 PROJECTION FROM P^k TO P^n

Let us consider a projective camera **P** which projects a point **X** in the kD projective space to a point **x** in the nD projective space.

$$\mathbf{x} \sim \mathbf{P}\mathbf{X} \tag{1}$$

where, $\sim$ denotes equality up to a scale. The camera matrix, **P**, of this camera is $(n+1)\times(k+1)$ and has $((k+1)\times(n+1)-1)$ DOF.

3 MULTIPLE VIEW GEOMETRY OF MIXED DIMENSIONAL CAMERAS

We next consider the properties of the multiple view geometry of mixed dimensional cameras, which represent geometric relationships of multiple cameras with various dimensions. Let us consider kD projective space, P^k, and a set of various dimensional cameras in the space. Consider k types of cameras C^i $(i=1,\cdots,k)$ which induce projections from P^k to P^i $(i=1,\cdots,k)$ respectively. For example, C^1 type cameras project a point in P^k to a point in P^1, and C^2 type cameras project a point in P^k to a point in P^2. Suppose there are n_i cameras of type C^i $(i=1,\cdots,k)$ in the kD space. Then, we have totally $N=\sum_{i=1}^{k} n_i$ cameras in the kD space. In this paper, a set of these cameras is represented by a k dimensional vector, **n**, as follows:

$$\mathbf{n} = [n_1, n_2, \cdots, n_k]^\top \tag{2}$$

Now, we consider DOF of N view geometry of mixed dimensional cameras, and specify the number of points required for computing the N view geometry of mixed dimensional cameras. Since camera matrices from P^k to P^i are $(k+1)\times(i+1)$, the DOF of N cameras is $N((k+1)(i+1)-1)$. The kD homography is represented by $(k+1)\times(k+1)$ matirx, and so it has $(k+1)^2-1$ DOF. Since these N cameras are in a single kD projective space, the total DOF of these N cameras is as follows:

$$L = \sum_{i=1}^{k} n_i((k+1)(i+1)-1)-(k+1)^2+1 \tag{3}$$

$$= (k+1)(\mathbf{n}^\top\mathbf{i}-k)+kN-k \tag{4}$$

where, $\mathbf{i}=[1,2,\cdots,k]^\top$. Thus, the N view geometry of mixed dimensional cameras has L DOF. The very special case of the multiple view geometry of mixed dimensional cameras is the traditional multiple view geometry of 2D cameras which induce projections from P^3 to P^2. In this case, $k=3$ and $\mathbf{n}=[0,2,0]^\top$.

Table 1: The minimum number of corresponding points required for computing the multiple view geometry of mixed dimensional cameras in the 3D space. Note, the multiple view geometries of $\mathbf{n}^\top=[3,0,0]$, $[2,0,0]$ and $[1,1,0]$ do not exist, since the image information is not enough for defining the multiple view geometry in these cases.

4 Views		3 Views		2 Views	
$\mathbf{n}^\top$	#	$\mathbf{n}^\top$	#	$\mathbf{n}^\top$	#
[4,0,0]	13	[2,1,0]	10	[0,2,0]	7
[3,1,0]	9	[1,2,0]	7	[1,0,1]	7
[2,2,0]	7	[0,3,0]	6	[0,1,1]	6
[1,3,0]	7	[2,0,1]	7	[0,0,2]	5
[0,4,0]	6	[1,1,1]	6		
[3,0,1]	7	[0,2,1]	6		
[2,1,1]	7	[1,0,2]	6		
[1,2,1]	6	[0,1,2]	6		
[0,3,1]	6	[0,0,3]	5		
[2,0,2]	6				
[1,1,2]	6				
[0,2,2]	6				
[1,0,3]	6				
[0,1,3]	6				
[0,0,4]	5				

Suppose M points in the kD space are projected to these N cameras. Then we have image information with $M\mathbf{n}^\top\mathbf{i}$ DOF from these cameras. Thus, the following inequality must hold for fixing all the geometry of N cameras and M points in the kD space.

$$M\mathbf{n}^\top\mathbf{i} \geq L + kM \tag{5}$$

By substituting (4) into (5), we find that the following condition must hold for computing the multiple view geometry of mixed dimensional cameras.

$$M \geq k+1+\frac{kN-k}{\mathbf{n}^\top\mathbf{i}-k} \tag{6}$$

(6) shows the minimum number of corresponding points required for computing the multiple view geometry of mixed dimensional cameras.

The complete table of the minimum number of corresponding points for mixed dimensional cameras in the 3D space is as shown in table 1.

In the following part of this paper, we show the detail of the multiple view geometry of some example combinations of different dimensional cameras.

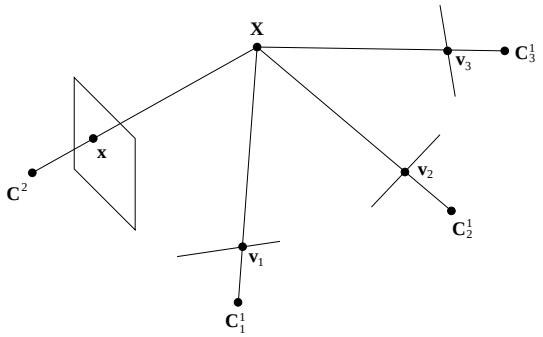

Figure 1: N view geometry of a 2D camera $\mathbf{C}^2$ and $N-1$ 1D cameras $\mathbf{C}_i^1$ $(i = 1, \cdots, N-1)$.

4 N VIEW GEOMETRY OF $\mathbf{n} = [N-1, 1, 0]^\top$

In this section, we consider the N view geometry of $\mathbf{n} = [N-1, 1, 0]^\top$, i.e. a single 2D camera and $(N-1)$ 1D cameras and no 3D camera in the 3D space. Let a 3D point $\mathbf{X} = [X^1, X^2, X^3, X^4]^\top$ be projected to $\mathbf{x} = [x^1, x^2, x^3]^\top$ in a 2D camera $\mathbf{C}^2$, and let $\mathbf{X}$ be projected to $\mathbf{v}_i = [v_i^1, v_i^2]^\top$ in 1D line cameras $\mathbf{C}_i^1$ $(i = 1, \cdots, N-1)$ as shown in Fig. 1. These projections can be described by using the camera matrices $\mathbf{P}^2$ of $\mathbf{C}^2$ and $\mathbf{P}_i^1$ of $\mathbf{C}_i^1$ as follows:

$$\mathbf{x} \sim \mathbf{P}^2\mathbf{X} \tag{7}$$

$$\mathbf{v}_i \sim \mathbf{P}_i^1\mathbf{X} \quad (i = 1, \cdots, N-1) \tag{8}$$

where, $\mathbf{P}^2$ is a 3×4 matrix and $\mathbf{P}_i^1$ are 2×4 matrices respectively. By reformulating (7) and (8), we have the following equation for $\mathbf{X}$:

$$\begin{bmatrix} \mathbf{P}^2 & \mathbf{x} & & & \\ \mathbf{P}_1^1 & & \mathbf{v}_1 & & \\ \vdots & & & \ddots & \\ \mathbf{P}_{N-1}^1 & & & & \mathbf{v}_{N-1} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \lambda_{\mathbf{x}} \\ \lambda_{\mathbf{v}_1} \\ \vdots \\ \lambda_{\mathbf{v}_{N-1}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{9}$$

where, λ denotes a scalar. Extracting the $(N+4) \times (N+4)$ sub-square matrix $\mathbf{M}$ of the left most matrix in (9), and computing the determinant of $\mathbf{M}$, we have the following multilinear constraints of mixed dimensional cameras:

$$\det \mathbf{M} = 0 \tag{10}$$

By expanding (10) in 3 views and 4 views, we have the following trilinear constraints and quadrilinear constraints respectively:

$$x^i v_1^j v_2^k \varepsilon_{jb} \varepsilon_{kc} \mathcal{T}_i^{bc} = 0 \tag{11}$$

$$x^i v_1^j v_2^k v_3^l \varepsilon_{iam} \varepsilon_{jb} \varepsilon_{kc} \varepsilon_{ld} \mathcal{Q}^{abcd} = 0_m \tag{12}$$

where the indices i, a and m take 1 through 3, and all the other indices take 1 or 2 in (11) and (12). The tensor ε_{ijk} takes 1 if $\{i, j, k\}$ is even permutation, and it takes -1 if $\{i, j, k\}$ is odd permutation. Similarly, the tensor ε_{ij} takes 1 if $\{i, j\}$ is even permutation, and it takes -1 if $\{i, j\}$ is odd permutation.

The $3 \times 2 \times 2$ tensor $\mathcal{T}_i^{bc}$ is the trifocal tensor, and the $3 \times 2 \times 2 \times 2$ tensor $\mathcal{Q}^{abcd}$ is the quadrifocal tensor of the mixed dimensional cameras. Note there is no 2 view geometry in this case, since the combination of a 2D camera and a 1D line camera does not have enough information for defining the 2 view constraints. Also, there is no linear constraint for more than 4 views.

By substituting $\mathbf{n} = [N-1, 1, 0]^\top$ and $k = 3$ into (4), we find that the N view geometry has $7N-11$ DOF in this case, and therefore the trifocal tensor $\mathcal{T}$ has 10 DOF, and the quadrifocal tensor $\mathcal{Q}$ has 17 DOF.

From (6), we find that the minimum number of corresponding points required for computing the trifocal tensor $\mathcal{T}$ is 10, while the minimum number of corresponding points for the quadrifocal tensor $\mathcal{Q}$ is 9.

We next consider linear estimation of $\mathcal{T}$ tensor and $\mathcal{Q}$ tensor. Since $\mathcal{T}$ tensor is $3 \times 2 \times 2$, it has 11 unknowns except a scale. On the other hand, a set of corresponding points in three views provides us a single constraint for $\mathcal{T}$ tensor from (11). Thus, we require 11 corresponding points for computing $\mathcal{T}$ tensor linearly.

Similarly, $\mathcal{Q}$ tensor is $3 \times 2 \times 2 \times 2$ and has 23 unknowns except a scale. Since (12) provides us 2 linearly independent constraints for $\mathcal{Q}$ tensor, we can compute $\mathcal{Q}$ tensor linearly from 12 corresponding points.

Once the trifocal tensor $\mathcal{T}$ is obtained, the epipolar line $l_i = v_1^j v_2^k \varepsilon_{jb} \varepsilon_{kc} \mathcal{T}_i^{bc}$ in the 2D image can be computed from $\mathbf{v}_1$ and $\mathbf{v}_2$ in 1D images. Similarly, once the quadrifocal tensor is obtained, the corresponding point $\mathbf{x}$ in the 2D image can be computed from the points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ in 1D images by using (12).

5 N VIEW GEOMETRY OF $\mathbf{n} = [0, N-1, 1]^\top$

We next consider the N view geometry in the case where a 3D range sensor and multiple 2D cameras exist in the 3D space, i.e. $k = 3$ and $\mathbf{n} = [0, N-1, 1]^\top$.

Let a 3D point $\mathbf{X} = [X^1, X^2, X^3, X^4]^\top$ be measured by a 3D range sensor $\mathbf{C}^3$, and let $\mathbf{y} = [y^1, y^2, y^3, y^4]^\top$ be the data measured by this sensor as shown in Fig. 2.

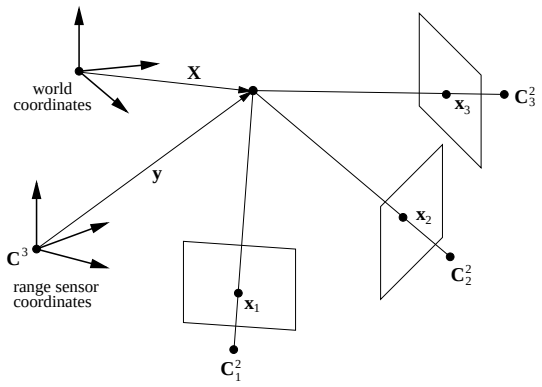

Figure 2: N view geometry of a 3D range sensor $\mathbf{C}^3$ and $N-1$ 2D cameras $\mathbf{C}_i^2$ $(i = 1, \cdots, N-1)$.

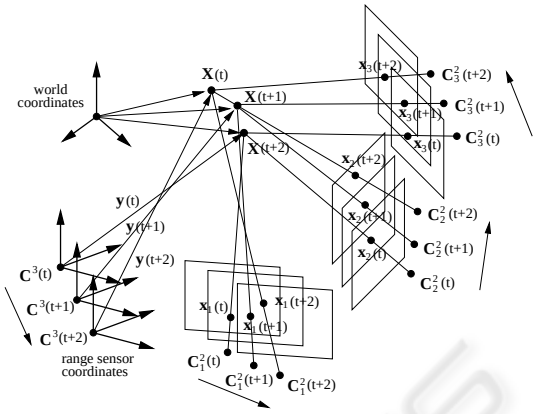

Figure 3: N view geometry of a moving 3D range sensor $\mathbf{C}^3$ and $N-1$ moving 2D cameras $\mathbf{C}_i^2$ $(i = 1, \cdots, N-1)$.

Suppose the point $\mathbf{X}$ is also observed by $(N-1)$ 2D cameras $\mathbf{C}_i^2$ as $\mathbf{x}_i = [x_i^1, x_i^2, x_i^3]^\top$ $(i = 1, \cdots, N-1)$.

Then, the projection in the 3D range sensor and the 2D cameras can be described as follows:

$$\mathbf{y} \sim \mathbf{P}^3\mathbf{X} \quad (13)$$

$$\mathbf{x}_i \sim \mathbf{P}_i^2\mathbf{X} \quad (i = 1, \cdots, N-1) \quad (14)$$

where, $\mathbf{P}^3$ is a 4×4 matrix with 15 DOF, and $\mathbf{P}_i^2$ is 3×4 projection matrices with 11 DOF.

From the similar analysis with section 4, we can derive the following bilinear, trilinear and quadrilinear constraints for the mixed dimensional cameras:

$$y^i x_1^j \varepsilon_{jbp} \mathcal{B}_i^b = 0_p \quad (15)$$

$$y^i x_1^j x_2^k \varepsilon_{iamn} \varepsilon_{jbp} \varepsilon_{kcq} \mathcal{T}^{ambc} = 0_{npq} \quad (16)$$

$$y^i x_1^j x_2^k x_3^l \varepsilon_{iamn} \varepsilon_{jbp} \varepsilon_{kcq} \varepsilon_{ldr} \mathcal{Q}^{abcd} = 0_{mnpqr} \quad (17)$$

where, indices i, a, m and n take 1 through 4, and all the other indices take 1 through 3 in (15), (16) and (17). The tensor ε_{ijkl} takes 1 if $\{i, j, k, l\}$ is even permutation, and it takes -1 if $\{i, j, k, l\}$ is odd permutation.

The 4×3 tensor $\mathcal{B}_i^b$, $4\times 4\times 3\times 3$ tensor $\mathcal{T}^{ambc}$ and $4\times 3\times 3\times 3$ tensor $\mathcal{Q}^{abcd}$ are the bifocal tensor, trifocal tensor and quadrifocal tensor of the mixed dimensional cameras.

By substituting $\mathbf{n} = [0, N-1, 1]^\top$ and $k = 3$ into (4) we find that the N view geometry has $11N-11$ DOF. Thus, $\mathcal{B}_i^b$ has 11 DOF, $\mathcal{T}^{ambc}$ has 22 DOF and $\mathcal{Q}^{abcd}$ has 33 DOF respectively. Also, we find from (6) that the following condition must hold for computing $\mathcal{B}_i^b$, $\mathcal{T}^{ambc}$ and $\mathcal{Q}^{abcd}$:

$$M \geq \frac{11}{2} \quad (18)$$

This means that the minimum number of corresponding points required for computing the multifocal tensors is 6 and is irrespective of the number of cameras.

We next consider the linear estimation of these multifocal tensors. The bifocal tensor $\mathcal{B}_i^b$ is 4×3 and has 11 unknowns except a scale. Since we have 2 linearly independent constraints for $\mathcal{B}_i^b$ from (15), 6 corresponding points are enough for computing $\mathcal{B}_i^b$ linearly.

6 N VIEW GEOMETRY OF $\mathbf{n} = [0, N-1, 1, 0]^\top$

We next extend the case described in section 5 and consider the N view geometry of a translational 3D range sensor and translational 2D cameras as shown in Fig. 3. If the 3D range sensor and the 2D cameras move independently, the multilinear constraints described by (15), (16) and (17) no longer hold. However, if we consider the multiple view geometry in the higher dimensional space, then we can derive the multilinear constraints which describe the relationship between a moving 3D range sensor and moving 2D cameras.

Suppose we have a moving point $\widetilde{\mathbf{X}} = [X, Y, Z]^\top$ in the 3D space, and suppose the point is projected to an image point $\widetilde{\mathbf{x}} = [x, y]^\top$ in a 2D camera. If the camera moves translationally with a constant speed, ΔX, ΔY and ΔZ in X, Y and Z directions, we can describe the relationship between the image point $\widetilde{\mathbf{x}}$ and the 3D point $\widetilde{\mathbf{X}}$ as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X - T\Delta X \\ Y - T\Delta Y \\ Z - T\Delta Z \\ 1 \end{bmatrix} \quad (19)$$

where, T denotes a time. If we consider a 4D space which consists of X, Y, Z and T, then the projection described by (19) can be rewritten as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} P_{11} & P_{12} & P_{13} & -P_{11}\Delta X - P_{12}\Delta Y - P_{13}\Delta Z & P_{14} \\ P_{21} & P_{22} & P_{23} & -P_{21}\Delta X - P_{22}\Delta Y - P_{23}\Delta Z & P_{24} \\ P_{31} & P_{32} & P_{33} & -P_{31}\Delta X - P_{32}\Delta Y - P_{33}\Delta Z & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \\ 1 \end{bmatrix} \quad (20)$$

Since the motion of the camera is unknown but is constant, the 3×5 projection matrix in (20) is unknown but is fixed. This means a moving camera in the 3D space can be considered as a static camera in the 4D space.

Similarly, a translational 3D range sensor can be considered as a static range sensor in the 4D space as follows:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \sim \begin{bmatrix} P_{11} & P_{12} & P_{13} & -P_{11}\Delta X - P_{12}\Delta Y - P_{13}\Delta Z & P_{14} \\ P_{21} & P_{22} & P_{23} & -P_{21}\Delta X - P_{22}\Delta Y - P_{23}\Delta Z & P_{24} \\ P_{31} & P_{32} & P_{33} & -P_{31}\Delta X - P_{32}\Delta Y - P_{33}\Delta Z & P_{34} \\ P_{41} & P_{42} & P_{43} & -P_{41}\Delta X - P_{42}\Delta Y - P_{43}\Delta Z & P_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \\ 1 \end{bmatrix} \quad (21)$$

This means the multiple view constraints between a moving range sensor and moving cameras can be defined in the 4D space. Thus, we consider the multiple view geometry in the case where $k = 4$ and $\mathbf{n} = [0, N-1, 1, 0]^\top$.

Suppose a 4D point, whose homogeneous coordinates are represented by $\mathbf{W} = [W^1, W^2, W^3, W^4, W^5]^\top$, is projected to an image point in a translational 2D camera, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Suppose the 4D point $\mathbf{W}$ is also measured by a translating 3D range sensor as $\mathbf{y} = [y^1, y^2, y^3, y^4]^\top$. Then, the projection in the moving 3D range sensor and the moving 2D cameras can be described as follows:

$$\mathbf{y} \sim \mathbf{P}^5\mathbf{W} \quad (22)$$

$$\mathbf{x}_i \sim \mathbf{P}_i^4\mathbf{W} \quad (i = 1, \cdots, N-1) \quad (23)$$

where, $\mathbf{P}^5$ is a 4×5 matrix with 19 DOF, and $\mathbf{P}_i^4$ are 3×5 matrices with 14 DOF.

By defining the matrix $\mathbf{M}$ similar to the one in (9) , and expanding $\det\mathbf{M} = 0$, we have the bilinear, trilinear, quadrilinear and quintilinear constraints (i.e. 5 view constraints) for the moving cameras and range sensors as follows:

$$y^i x_1^j \mathcal{B}_{ij} = 0 \quad (24)$$

$$y^i x_1^j x_2^k \varepsilon_{jbr} \varepsilon_{kcs} \mathcal{T}_i^{bc} = 0_{rs} \quad (25)$$

$$y^i x_1^j x_2^k x_3^l \varepsilon_{iapq} \varepsilon_{jbr} \varepsilon_{kcs} \varepsilon_{ldt} \mathcal{Q}^{apbcd} = 0_{qrst} \quad (26)$$

$$y^i x_1^j x_2^k x_3^l x_4^m \varepsilon_{iapq} \varepsilon_{jbr} \varepsilon_{kcs} \varepsilon_{ldt} \varepsilon_{meu} \mathcal{R}^{abcde} = 0_{pqrstu} \quad (27)$$

where, indices i, a, p, q take 1 through 4, and all the other indices take 1 through 3. $\mathcal{B}_{ij}$, $\mathcal{T}_i^{bc}$, $\mathcal{Q}^{apbcd}$ and $\mathcal{R}^{abcde}$ are the bilinear, trilinear quadrilinear and quintilinear tensors respectively.

By substituting $k = 4$ and $\mathbf{n} = [0, N-1, 1, 0]^\top$ into (4), we find that the N view geometry has $14N - 19$ DOF, and thus $\mathcal{B}_{ij}$, $\mathcal{T}_i^{bc}$, $\mathcal{Q}^{apbcd}$ and $\mathcal{R}^{abcde}$ have 9 DOF, 23 DOF, 37 DOF and 51 DOF respectively.

Also, by substituting $k = 4$ and $\mathbf{n} = [0, N-1, 1, 0]^\top$ into (6), we find that the minimum number of corresponding points required for computing $\mathcal{B}_{ij}$, $\mathcal{T}_i^{bc}$, $\mathcal{Q}^{apbcd}$ and $\mathcal{R}^{abcde}$ is 9, 8, 8 and 8 respectively.

7 EXPERIMENTS

In this section, we show the results of synthetic image experiments.

7.1 2D Camera and 1D Line Cameras

We first show the results from the combination of a single 2D camera and multiple 1D line cameras. Fig. 4 shows a 2D camera and three 1D line cameras used in this experiment. Fig. 5 (a) shows the image viewed from the 2D camera, and (b), (c) and (d) show images viewed from three 1D cameras respectively. The images of $\mathbf{C}^2$, $\mathbf{C}_1^1$ and $\mathbf{C}_2^1$ are used for computing the trifocal tensor, $\mathcal{T}_i^{bc}$, of these three cameras. Then, $\mathcal{T}_i^{bc}$ tensor was used for computing the epipolar lines in $\mathbf{C}^2$ image from the image points in 1D images of $\mathbf{C}_1^1$ and $\mathbf{C}_2^1$. The extracted epipolar lines are shown in Fig. 6. As shown in this figure, the extracted epipolar lines go through the corresponding points in the 2D image as we expected.

We next computed the quadrifocal tensor, $\mathcal{Q}^{abcd}$, for a 2D camera and three 1D cameras from images shown in Fig. 5 (a), (b), (c) and (d). The extracted $\mathcal{Q}^{abcd}$ tensor was used for computing a corresponding point in the 2D image from a set of points in the three 1D images. Fig. 7 shows the points in the 2D image, which are computed from $\mathcal{Q}^{abcd}$ tensor. As shown in this figure, the set of points in 1D cameras are transfered into the point in the 2D camera properly by using the proposed multiple view geometry.

7.2 Moving 3D Range Sensor and 2D Cameras

We next show the results from a moving 3D range sensor and moving 2D cameras. As we have seen in section 6, the multiple view geometry of moving range sensors and moving 2D cameras can be described by considering the projections from 4D space to 3D space and 2D space. We translated the 3D range sensor during the 3D measurement, and obtained the 3D data. $\mathbf{C}^3$ and $\mathbf{C}^{3\prime}$ in Fig. 8 show the position of the 3D range sensor before and after the translational motion. $\mathbf{C}_1^2$ and $\mathbf{C}_2^2$ show 2 fixed cameras. Fig. 9 (a)

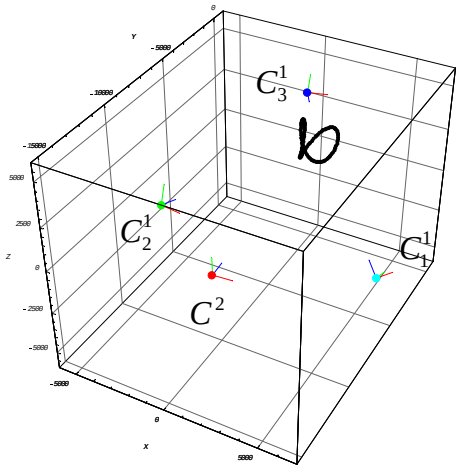

Figure 4: 3D scene. The black curve shows a locus of a moving point **X**. The moving point **X** is observed by a 2D camera, $\mathbf{C}^2$, and three 1D cameras, $\mathbf{C}_1^1$, $\mathbf{C}_2^1$ and $\mathbf{C}_3^1$. The three lines of each camera show the orientation of the camera.

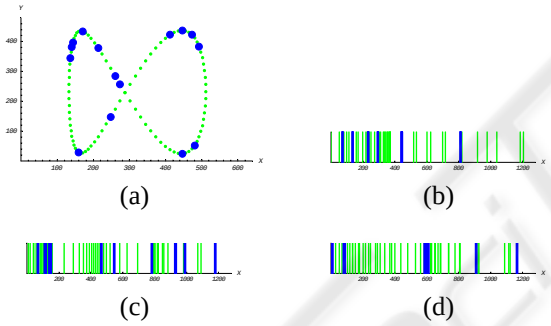

Figure 5: The images of a 2D camera and three 1D cameras. (a) shows the image of $\mathbf{C}^2$. (b), (c) and (d) show the image of $\mathbf{C}_1^1$, $\mathbf{C}_2^1$ and $\mathbf{C}_3^1$ respectively. The green points and lines in these images show the projection of a moving point. The blue points and lines show corresponding basis points for computing the multifocal tensors.

shows the 3D data measured by the translational 3D range sensor. As we can see in this figure, the measured 3D data is distorted because of the motion of the range sensor. The laser light of the range sensor reflected at the surface of the object was observed by 2 cameras. The green points in Fig. 9 (b) and (c) show the laser light observed in these 2 cameras. The blue points in these data show basis corresponding points used for computing the trifocal tensor. The extracted trifocal tensor was used for transfering the 3D range data into the image of camera $\mathbf{C}_1^2$. The green points in Fig. 10 show the original image points and the red points show the 3D data transfered by using the trifocal tensor. As shown in this figure, the 3D data was transfered properly by using the multiple view geometry in 4D space, even if the range sensor moved during the measurement.

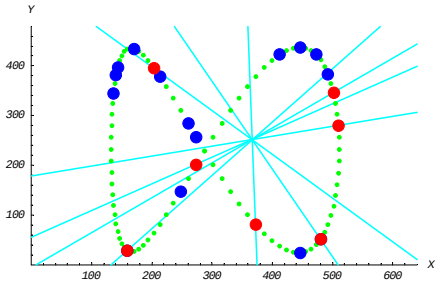

Figure 6: The epipolar lines computed from the trifocal tensor and the points in 1D line cameras. The blue points show the basis points used for computing the trifocal tensor, and the red points show some other corresponding points. Since the epipolar lines go through the corresponding points, we find that the computed trifocal tensor is accurate.

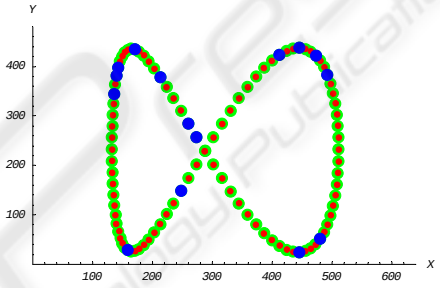

Figure 7: The points in the 1D camera images are transfered into the 2D camera image by using the extracted quadrifocal tensor. The green points show the original loci in the image, and the red points show the result of the point transfer.

We next computed the 3D range data from two 2D camera images and the extracted trifocal tensor. Fig. 11 shows the points in the 3D image, which are computed from the trifocal tensor. As shown in this figure, the set of points in 2D camera images are transfered into the points in the 3D range data properly by using the proposed multiple view geometry. These properties can be used for mapping the image textures to the range image only by using point transfer.

8 CONCLUSIONS

In this paper, we showed that there exist multilinear constraints on image coordinates, even if the dimensions of camera images are different from each other. We first analyzed the multiple view geometry of general mixed dimensional cameras. We next showed

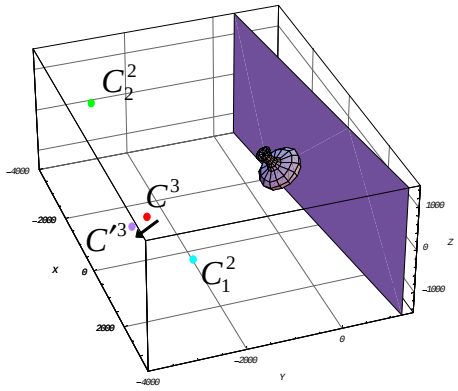

Figure 8: 3D scene. A 3D object (vase) is observed by a moving 3D range sensor, $\mathbf{C}^3$, and two 2D cameras, $\mathbf{C}_1^2$, $\mathbf{C}_2^2$. $\mathbf{C}^{3'}$ shows the position of the range sensor after a translational motion.

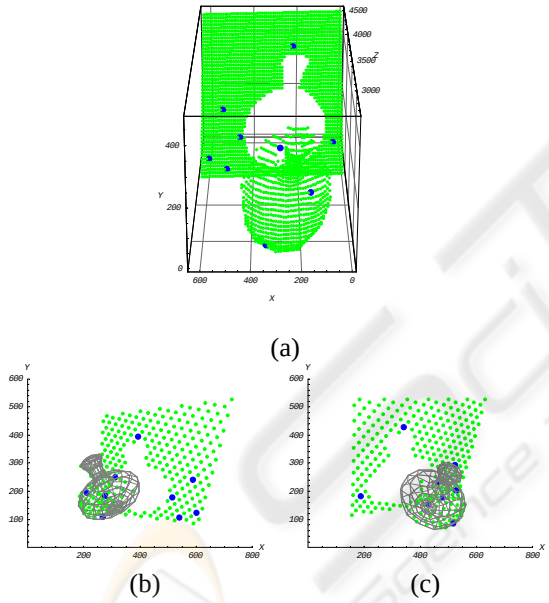

Figure 9: Range data and camera images. The green points in (a) show the range data obtained from the range sensor, and the green points in (b) and (c) show the laser points observed in 2 cameras. The blue points in these images show basis points used for computing the trifocal tensor.

the multilinear constraints of some example cases of mixed dimensional cameras. The new multilinear constraints can be used for describing the geometric relationships between 1D line sensors, 2D cameras, 3D range sensors, etc. and thus they are useful for calibrating sensor systems in which different types of cameras and sensors are used together. The power of

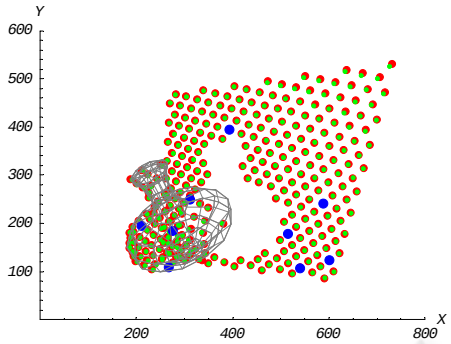

Figure 10: The range data transfered into the camera image. The green points show the original laser points observed in the image, and the red points show the points transfered from the 3D range data by using the trifocal tensor. The blue points show the basis points used for computing the trifocal tensor.

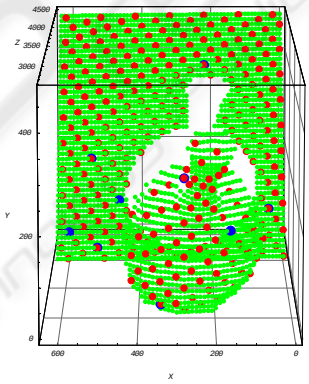

Figure 11: The image data transfered into the range data. The green points show the original laser points, and the red points show the points transfered from the 2D camera images by using the trifocal tensor. The blue points show the basis points used for computing the trifocal tensor.

the new multiple view geometry was shown by using synthetic images in some mixed sensor systems.

REFERENCES

Faugeras, O. and Keriven, R. (1995). Scale-space and affine curvature. In *Proc. Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 17–24.

Faugeras, O. and Luong, Q. (2001). *The Geometry of Multiple Images*. MIT Press.

Hartley, R. and Schaffalitzky, F. (2004). Reconstruction from projections using grassman tensors. In *Proc. 8th European Conference on Computer Vision*, volume 1, pages 363–375.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hayakawa, K. and Sato, J. (2006). Multiple View Geometry in the Space-Time. In *Proc. 7th Asian Conference on Computer Vision*, volume 2, pages 437–446.

Heyden, A. (1998). A common framework for multiple view tensors. In *Proc. 5th European Conference on Computer Vision*, volume 1, pages 3–19.

Shashua, A. and Wolf, L. (2000). Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *Proc. 6th European Conference on Computer Vision*, volume 1, pages 507–521.

Sturm, P. (2002). Mixing catadioptric and perspective cameras. In *Proc. Workshop on Omnidirectional Vision*.

Sturm, P. (2005). Multi-view geometry for general camera models. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 206–212.

Thirthala, S. and Pollefeys, M. (2005). Trifocal tensor for heterogeneous cameras. In *Proc. Workshop on Omnidirectional Vision*.

Triggs, B. (1995). Matching constraints and the joint image. In *Proc. 5th International Conference on Computer Vision*, pages 338–343.

Wexler, L. and Shashua, A. (2000). On the synthesis of dynamic scenes from reference views. In *Proc. Conference on Computer Vision and Pattern Recognition*.

Wolf, L. and Shashua, A. (2001). On projection matrices $P^k \rightarrow P^2, k = 3, \cdots, 6$, and their applications in computer vision. In *Proc. 8th International Conference on Computer Vision*, volume 1, pages 412–419.