PAPER

Multiple View Geometry for Curvilinear Motion Cameras

Cheng WAN[†a)] ***and*** **Jun SATO**[††b)]**,** ***Members***

SUMMARY This paper introduces a tensorial representation of multiple cameras with arbitrary curvilinear motions. It enables us to define a multilinear relationship among image points derived from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We show the new multilinear relationship is useful for generating images and reconstructing 3D non-rigid object motions viewed from cameras with arbitrary curvilinear motions. The method is tested in real image sequences.

key words: *multiple view geometry, curvilinear motion, spline curve, multifocal tensor, multiple cameras, camera calibration*

1. Introduction

The structure from motion problem (SFM) is to extract the 3D shape of the scene as well as the camera motion from a set of images taken by a camera undergoing unknown motion. The traditional methods in SFM provide us solutions if a moving camera observes a static scene or a set of static cameras observe a dynamic scene [1], [2]. In this paper, we consider SFM problem under dynamic environments, where both the set of cameras and the scene change non-rigidly. In particular, we consider multiple view geometry under non-rigid object motions viewed from multiple moving cameras.

Multiple view geometry can be used to describe the relationship among images taken from multiple cameras and to recover 3D geometry from images. From stationary configurations [2]–[5] to dynamic configurations [6]–[10], multiple view geometry has been extensively developed. However, previous multiple view geometry involving dynamic scenes are constrained from the translational motions of the cameras [9], [10] or a configuration of points in which each point can move independently along some restricted trajectory, i.e., straight line path and in some cases second-order [6]–[8]. In this research, we consider the case where non-rigid object motions are viewed from curvilinear motion cameras as shown in Fig. 1. The curvilinear motion means a curved trajectory without rotation.

A degree-n B-Spline curve is a set of piecewise smooth curves [11]. It is differentiable in high order at the joint of piecewise curves. Even if piecewise curves are low degree, B-Spline curve can represent a complex curve. Therefore, in this paper we use B-Spline curve to model the trajectory of the camera and adopt the widely used cubic B-Spline curve to describe the camera motion. We show that arbitrary non-rigid motions viewed from multiple cubic B-spline curve motion cameras could be represented by the multiple view geometry from 6D to 2D.

To do this, we assume affine cameras as a camera model. We analyze multiple view geometry under the projection from 6D to 2D, and show that we have multilinear relationships for up to 7 views. The four-view, five-view, six-view and seven-view geometries are studied extensively, and new quadrilinear, quintilinear, sextilinear and septilinear relationships from 6D space to 2D space are presented. The results from experiments show that the defined 6D to 2D multiple view geometry can be used to describe the relationship among images taken from non-rigid motions viewed from multiple curvilinear motion cameras, and it is also useful for view transfer among curvilinear motion cameras and 3D reconstruction.

2. Non-rigid Object Motions Viewed from Curvilinear Motion Cameras

Let us consider a single moving point in the 3D space. If

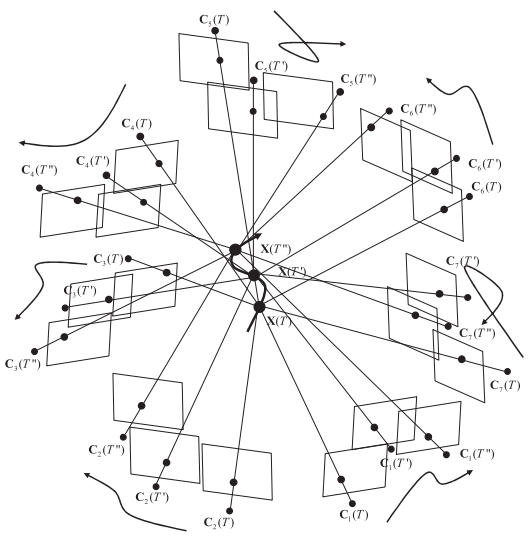

Fig. 1 A moving point in 3D space and its projections in seven curvilinear motion cameras. The multifocal tensor defined under projections from P^6 to P^2 can describe the relationship among these image projections.

Manuscript received October 4, 2010.
Manuscript revised January 29, 2011.

[†]The author is with Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China.
[††]The author is with Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.
a) E-mail: wanch@nuaa.edu.cn
b) E-mail: junsato@nitech.ac.jp

DOI: 10.1587/transinf.E94.D.1479

the multiple cameras are stationary or translational, we can compute the multifocal tensors with known methods [2], [9] to figure out multiple view geometry. However, if these cameras have independent curvilinear motions, the traditional multifocal tensors cannot be computed from the trajectory of the point. Nonetheless, we in this section show that if the camera motions are curvilinear as shown in Fig. 1, the multiple view geometry under extended projections can be computed from the trajectory of the point, and they can be used to, for example, generate point motions viewed from arbitrary curvilinear motion cameras.

Consider a usual affine camera which projects points in 3D to 2D images. The position of a point in the 3D space can be represented by homogeneous coordinate, $\mathbf{X}(T) = [X(T), Y(T), Z(T), 1]^\top$, where T denotes time. The trajectory of the point is projected to images, and can be observed as a set of points $[x(T), y(T), 1]^\top$. As we know, in the mathematical field of numerical analysis, B-spline curves are very useful for representing arbitrary 3D shapes with small number of control points. Hence, we make use of cubic B-spline curves [11] to describe the arbitrary 3D motions of cameras $\Delta\mathbf{V} = [\Delta X, \Delta Y, \Delta Z, \Delta W]^\top$ in homogeneous coordinates in this paper. The camera motion is relative to the camera initial position, and hence its fourth entry is equal to 0, and thus represented as $\Delta\mathbf{V} = [\Delta X, \Delta Y, \Delta Z, 0]^\top$. The ith segment of a B-spline curve is defined using four control points, $\mathbf{Q}_{i-1}$, $\mathbf{Q}_i$, $\mathbf{Q}_{i+1}$, $\mathbf{Q}_{i+2}$ and a parameter t as follows:

$$\Delta\mathbf{V}_i = [\mathbf{Q}_{i-1}, \mathbf{Q}_i, \mathbf{Q}_{i+1}, \mathbf{Q}_{i+2}]\,\mathbf{B}\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}, \quad (t \in [0,1], \quad i = 1, 2, 3, \cdots) \tag{1}$$

where, the fourth entries of $\mathbf{Q}_{i-1}$, $\mathbf{Q}_i$, $\mathbf{Q}_{i+1}$, $\mathbf{Q}_{i+2}$ are equal to 0, since the forth entry of $\Delta\mathbf{V}_i$ is equal to 0. $\mathbf{B}$ denotes the following 4×4 matrix:

$$\mathbf{B} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}/6,$$

Assume each motion segment $\Delta\mathbf{V}_i$ spends time T_a. Then $t = T/T_a - i + 1$. Thus, the parameter vector can be written as:

$$\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \mathbf{C}_i \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \tag{2}$$

where

$$\mathbf{C}_i = \begin{bmatrix} 1/{T_a}^3 & -3(i-1)/{T_a}^2 & 3(i-1)^2/T_a & -(i-1)^3 \\ 0 & 1/{T_a}^2 & -2(i-1)/T_a & (i-1)^2 \\ 0 & 0 & 1/T_a & -i+1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let $\mathbf{G}_i = [\mathbf{Q}_{i-1}, \mathbf{Q}_i, \mathbf{Q}_{i+1}, \mathbf{Q}_{i+2}]$. Then, $\Delta\mathbf{V}_i$ can be rewritten as follows:

$$\Delta\mathbf{V}_i(T) = \mathbf{G}_i\mathbf{B}\mathbf{C}_i \begin{bmatrix} T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \tag{3}$$

Thus, point motions $\mathbf{X}(T)$ are projected to an affine camera with cubic B-spline curvilinear motion $\Delta\mathbf{V}_i(T)$ as follows:

$$\begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} = \mathbf{P}_a(\mathbf{X}(T) - \Delta\mathbf{V}_i(T)) \tag{4}$$

where $\mathbf{P}_a$ denotes a 3×4 affine camera matrix, whose third row is $[0, 0, 0, 1]$, and $(\mathbf{X}(T) - \Delta\mathbf{V}_i(T))$ is the position of a 3D point at time T relative to the camera. By substituting (3) into (4), we have the following equation:

$$\begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} = \mathbf{P}_a[\mathbf{I}, -\mathbf{G}_i\mathbf{B}\mathbf{C}_i] \begin{bmatrix} \mathbf{X}(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \tag{5}$$

Suppose $\mathbf{D}_i = \mathbf{P}_a[\mathbf{I}, -\mathbf{G}_i\mathbf{B}\mathbf{C}_i]$. Then

$$\begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} = \mathbf{D}_i \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ 1 \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \tag{6}$$

Let us consider the following 8×7 matrix $\mathbf{L}$:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Then, (6) can be described as follows:

$$\begin{bmatrix} x(T) \\ y(T) \\ 1 \end{bmatrix} = \mathbf{P}_i \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T^3 \\ T^2 \\ T \\ 1 \end{bmatrix} \tag{8}$$

where, $\mathbf{P}_i = \mathbf{D}_i\mathbf{L}$. $\mathbf{D}_i$ represents a 3×8 matrix, and $\mathbf{P}_i$ denotes a 3×7 extended affine camera matrix, whose third row is $[0, 0, 0, 0, 0, 0, 1]$. $\mathbf{P}_i$ depends on the choice of control points. We therefore find that, from (8), the projections of point motions to the image plane of a camera with a single segment B-spline curve motion can be described by the projection from 6D to 2D. In the next sections, the geometry of such projection will be given in more detail.

3. Projection from 6D to 2D

We first consider a projection from 6D space to 2D space. Let $\mathbf{W} = [W^1, W^2, W^3, W^4, W^5, W^6, W^7]^\top$ be the homogeneous coordinates of a 6D space point projected to a point in the 2D space, whose homogeneous coordinates are represented by $\mathbf{x} = [x^1, x^2, x^3]^\top$. Then, the extended affine projection from $\mathbf{W}$ to $\mathbf{x}$ can be described as follows:

$$\mathbf{x} \sim \mathbf{P}\mathbf{W} \tag{9}$$

where $(\sim)$ denotes equality up to a scale, and $\mathbf{P}$ denotes the following 3×7 matrix:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

From (10), we find that the extended affine camera, $\mathbf{P}$, has 14 DOF. In the next section, we consider the multiple view geometry of the extended affine cameras.

4. Multiple View Geometry from 6D to 2D

From (9), we have the following equation for N extended affine cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{x}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{x}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{W} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \tag{11}$$

where, the leftmost matrix, $\mathbf{M}$, in (11) is $3N \times (7 + N)$. By deriving a $(7 + N) \times (7 + N)$ minor $\mathbf{R}$ of $\mathbf{M}$, we have a multilinear relationship under the extended affine projection as follows:

$$\det \mathbf{R} = 0$$

We can choose any $7 + N$ rows from $\mathbf{M}$ to constitute $\mathbf{R}$, but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note, each camera has 3 rows in $\mathbf{M}$). Thus, $7 + N \geq 2N$ must hold for defining multilinear relationships for N view geometry in the 6D space. Thus, we find that, the multilinear relationship for 7 views is the maximal linear relationship in the 6D space.

4.1 Four-View Geometry

We next introduce multiple view geometry of four extended cameras. For four views, the sub square matrix $\mathbf{R}$ is 11×11. From $\det \mathbf{R} = 0$, we have the following quadrilinear relationship under extended camera projections:

$$x^i x'^j x''^k x'''^h \epsilon_{hvd} Q^v_{ijk} = 0_d \tag{12}$$

where ϵ_{hvd} (or its contravariant counterpart, ϵ^{hvd}) denotes a tensor, which represents a sign based on permutation from $\{h, v, d\}$ to $\{1, 2, 3\}$, and equals 0 when index is repeated. In this paper we use Einstein's summation convention for representing tensor equations. Q^v_{ijk} is the quadrifocal tensor for the extended cameras and has the following form:

$$Q^v_{ijk} = \epsilon_{ipq}\epsilon_{jrs}\epsilon_{ktu} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{c}^u \\ \mathbf{d}^v \end{bmatrix} \tag{13}$$

where $\mathbf{a}^i$ denotes the ith row of $\mathbf{P}$, $\mathbf{b}^i$ denotes the ith row of $\mathbf{P}'$, $\mathbf{c}^i$ denotes the ith row of $\mathbf{P}''$ and $\mathbf{d}^i$ denotes the ith row of $\mathbf{P}'''$ respectively. The quadrifocal tensor Q^v_{ijk} is $3 \times 3 \times 3 \times 3$ and has 81 entries. Since all the third rows of the extended affine camera matrices are $[0, 0, 0, 0, 0, 0, 1]$, many zero entries arise in Q^v_{ijk}. As a result, Q^1_{133}, Q^2_{133}, Q^1_{233}, Q^2_{233}, Q^1_{313}, Q^2_{313}, Q^1_{323}, Q^2_{323}, Q^1_{331}, Q^2_{331}, Q^1_{332}, Q^2_{332}, Q^1_{333}, Q^2_{333}, Q^3_{333} are non-zero entries and thus we have only 14 free parameters in Q^v_{ijk} except a scale ambiguity. On the other hand, (12) provides us 3 linear equations on Q^v_{ijk}, but only 2 of them are linearly independent. Thus, at least 7 corresponding points are required to compute Q^v_{ijk} from images linearly.

Since corresponding points at each time induce linear constraints, for computing quadrifocal tensor, we reformulate (12) as follows:

$$\mathbf{E}(t)\mathbf{q} = \mathbf{0} \tag{14}$$

where $\mathbf{q} = [Q^1_{133}, Q^2_{133}, Q^1_{233}, Q^2_{233}, Q^1_{313}, Q^2_{313}, Q^1_{323}, Q^2_{323}, Q^1_{331}, Q^2_{331}, Q^1_{332}, Q^2_{332}, Q^1_{333}, Q^2_{333}, Q^3_{333}]^\top$, and $\mathbf{E}(t)$ is a 3×15 matrix whose elements are calculated from the corresponding points $\mathbf{x}(t)$, $\mathbf{x}'(t)$, $\mathbf{x}''(t)$ and $\mathbf{x}'''(t)$. Although (14) has 3 equations, only 2 of them are linearly independent. Then, if we have N corresponding points, $\mathbf{q}$ can be computed by solving the following linear equations.

$$\begin{aligned} &\mathbf{U}\mathbf{q} = \mathbf{0} \\ &\mathbf{U} = [\mathbf{E}(t_1)^\top, \cdots, \mathbf{E}(t_N)^\top]^\top \end{aligned} \tag{15}$$

where $N \geq 7$. The solution on $\mathbf{q}$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{U}^\top\mathbf{U}$.

Since two points x'''^d and x'''^h in the forth view can be used to represent a line l'''_v which goes through x'''^d and x'''^h as: $x'''^h x'''^d \epsilon_{hvd} = l'''_v$, (12) becomes

$$x^i x'^j x''^k x'''^h x'''^d \epsilon_{hvd} Q^v_{ijk} = x^i x'^j x''^k l'''_v Q^v_{ijk} = 0 \tag{16}$$

by multiplying x'''^d on both sides. Then, (16) shows the connection of the quadrifocal tensor with three points and one line. Furthermore, if multiplying x''^u, a point in the third view, to (16), we can derive:

$$\begin{aligned} x^i x'^j x''^k x''^u l'''_v Q^v_{ijk} &= \frac{1}{6} x^i x'^j x''^k x''^u \epsilon_{rku} \epsilon^{rku} l'''_v Q^v_{ijk} \\ &= \frac{1}{6} x^i x'^j l''_r l'''_v \epsilon^{rku} Q^v_{ijk} = 0^u \end{aligned} \tag{17}$$

Table 1 Quadrilinear relations between point and line coordinates in four views. The final column denotes the number of linearly independent equations.

correspondence	relation	eq.
four points	$x^i x'^j x''^k x'''^h \epsilon_{hvd} Q^v_{ijk} = 0_d$	2
three points, one line	$x^i x'^j x''^k l'''^v_v Q^v_{ijk} = 0$	1
two points, two lines	$x^i x'^j l''_r l'''_v \epsilon^{rku} Q^v_{ijk} = 0^u$	2
one point, three lines	$x^i l'_q l''_r l'''_v \epsilon^{qjt} \epsilon^{rku} Q^v_{ijk} = 0^{tu}$	4
four lines	$l_p l'_q l''_r l'''_v \epsilon^{pis} \epsilon^{qkt} \epsilon^{rku} Q^v_{ijk} = 0^{stu}$	8

where l''_r is a line in the third view going through x''^k and x'''^u. (17) is the correspondence on point-point-line-line. The other correspondences may be obtained in the same manner.

A complete set of the quadrilinear equations involving the quadrifocal tensor are given in Table 1. All of these equations are linear in the entries of the quadrifocal tensor Q^v_{ijk}.

As described in Sect. 2, this multiple view geometry can be applied to multiple affine cameras with curvilinear motions. Meanwhile, since the position of points in our research includes the information of time, we can derive the multiple view geometry from fewer time instants t if we observe more than one point. For example, in the case of four views, we need 7 time instants, if we observe a single point in the space. However, if we observe 2 point motions in 3D, we only need to observe them 4 time instants to figure out multiple view geometry.

4.2 Five-View, Six-View and Seven-View Geometry

Similarly, the five-view, six-view and seven-view geometry can also be derived for the extended cameras. The quintilinear relationship under extended projection is:

$$x^i x'^j x''^k x'''^h x''''^m \epsilon_{ktc} \epsilon_{hud} \epsilon_{mve} \mathcal{R}^{tuv}_{ij} = 0_{cde} \tag{18}$$

$\mathcal{R}^{tuv}_{ij}$ is the quintifocal tensor whose form is described as:

$$\mathcal{R}^{tuv}_{ij} = \epsilon_{ipq} \epsilon_{jrs} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{b}^s \\ \mathbf{c}^t \\ \mathbf{d}^u \\ \mathbf{e}^v \end{bmatrix} \tag{19}$$

where $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$, $\mathbf{d}^i$ and $\mathbf{e}^i$ denote the ith row of five camera matrices. The quintifocal tensor $\mathcal{R}^{tuv}_{ij}$ has 243 entries. Excluding 191 zero entries and a scale ambiguity, it has 51 free parameters. And 27 linear equations are given from (18) but only 8 of them are linearly independent. Therefore, minimum of 7 corresponding points are required to compute $\mathcal{R}^{tuv}_{ij}$ from images linearly. The quintilinear relationships involving the quintifocal tensor are summerized in Table 2.

We next introduce the multiple view geometry of six extended cameras. The sextilinear constraint is expressed as follows:

Table 2 Quintilinear relations between point and line coordinates in five views. The final column denotes the number of linearly independent equations.

relation	# of eq.
$x^i x'^j x''^k x'''^h x''''^m \epsilon_{ktc} \epsilon_{hud} \epsilon_{mve} \mathcal{R}^{tuv}_{ij} = 0_{cde}$	8
$x^i x'^j x''^k x'''^h l''''^v_v \epsilon_{ktc} \epsilon_{hud} \mathcal{R}^{tuv}_{ij} = 0_{cd}$	4
$x^i x'^j x''^k l'''^u_u l''''^v_v \epsilon_{ktc} \mathcal{R}^{tuv}_{ij} = 0_c$	2
$x^i x'^j l''_t l'''_u l''''_v \mathcal{R}^{tuv}_{ij} = 0$	1
$x^i l'_q l''_t l'''_u l''''_v \epsilon^{qjn} \mathcal{R}^{tuv}_{ij} = 0^n$	2
$l_p l'_q l''_t l'''_u l''''_v \epsilon^{pim} \epsilon_{qjn} \mathcal{R}^{tuv}_{ij} = 0^{mn}$	4

Table 3 Sextilinear relations between point and line coordinates in six views. The final column denotes the number of linearly independent equations.

relation	eq.
$x^i x'^j x''^k x'''^h x''''^m x'''''^n \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \epsilon_{mue} \epsilon_{nvf} \mathcal{S}^{rstuv}_i = 0_{bcdef}$	32
$x^i x'^j x''^k x'''^h x''''^m l'''''^n_v \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \epsilon_{mue} \mathcal{S}^{rstuv}_i = 0_{bcde}$	16
$x^i x'^j x''^k x'''^h l''''^m_u l'''''^n_v \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \mathcal{S}^{rstuv}_i = 0_{bcd}$	8
$x^i x'^j x''^k l'''^t_t l''''^u_u l'''''^v_v \epsilon_{jrb} \epsilon_{ksc} \mathcal{S}^{rstuv}_i = 0_{bc}$	4
$x^i x'^j l''^s_s l'''^t_t l''''^u_u l'''''^v_v \epsilon_{jrb} \mathcal{S}^{rstuv}_i = 0_b$	2
$x^i l'^r_r l''^s_s l'''^t_t l''''^u_u l'''''^v_v \mathcal{S}^{rstuv}_i = 0$	1
$l_m l'^r_r l''^s_s l'''^t_t l''''^u_u l'''''^v_v \epsilon^{miw} \mathcal{S}^{rstuv}_i = 0_w$	2

$$x^i x'^j x''^k x'''^h x''''^m x'''''^n \epsilon_{jrb} \epsilon_{ksc} \epsilon_{htd} \epsilon_{mue} \epsilon_{nvf} \mathcal{S}^{rstuv}_i = 0_{bcdef} \tag{20}$$

where $\mathcal{S}^{rstuv}_i$ is the sextifocal tensor (six view tensor) whose form is represented as follows:

$$\mathcal{S}^{rstuv}_i = \epsilon_{ipq} \det \begin{bmatrix} \mathbf{a}^p \\ \mathbf{a}^q \\ \mathbf{b}^r \\ \mathbf{c}^s \\ \mathbf{d}^t \\ \mathbf{e}^u \\ \mathbf{f}^v \end{bmatrix} \tag{21}$$

where $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$, $\mathbf{d}^i$, $\mathbf{e}^i$ and $\mathbf{f}^i$ denote the ith row of six camera matrices. The sextifocal tensor $\mathcal{S}^{rstuv}_i$ has 729 entries. If the extended cameras are affine as shown in (9), we have only 175 free parameters in $\mathcal{S}^{rstuv}_i$ except zero entries and a scale. On the other hand, (20) shows one set of corresponding points provides us 243 linear equations on $\mathcal{S}^{rstuv}_i$, but only 32 of them are linearly independent. Furthermore, the constraints between multiple sets of points are not independent. As a result, at least 7 corresponding points are required to compute $\mathcal{S}^{rstuv}_i$ from images linearly. The sextilinear relationships are given in Table 3.

Finally, let us have a look at the multiple view geometry of seven extended cameras. The septilinear constraint is described as:

$$x^i x'^j x''^k x'''^h x''''^m x'''''^n x''''''^o \epsilon_{ipa} \epsilon_{jqb} \epsilon_{krc} \epsilon_{hsd} \epsilon_{mte} \epsilon_{nuf} \epsilon_{ovg} \mathcal{H}^{pqrstuv} = 0_{abcdefg} \tag{22}$$

where $\mathcal{H}^{pqrstuv}$ is the septifocal tensor (seven view tensor) whose form is represented as follows:

Table 4 Septilinear relations between point and line coordinates in seven views. The final column denotes the number of linearly independent equations.

relation	# of eq.
$x^i x'^j x''^k x'''^h x''''^m x'''''^n x''''''^o \epsilon_{ipa}\epsilon_{jqb}\epsilon_{krc}\epsilon_{hsd}$ $\epsilon_{mte}\epsilon_{nuf}\epsilon_{ovg}\mathcal{H}^{pqrstuv} = 0_{abcdefg}$	128
$x^i x'^j x''^k x'''^h x''''^m x'''''^n l_v''''''' \epsilon_{ipa}\epsilon_{jqb}\epsilon_{krc}$ $\epsilon_{hsd}\epsilon_{mte}\epsilon_{nuf}\mathcal{H}^{pqrstuv} = 0_{abcdef}$	64
$x^i x'^j x''^k x'''^h x''''^m l_u''''' l_v''''''' \epsilon_{ipa}\epsilon_{jqb}$ $\epsilon_{krc}\epsilon_{hsd}\epsilon_{mte}\mathcal{H}^{pqrstuv} = 0_{abcde}$	32
$x^i x'^j x''^k x'''^h l_t''''' l_u''''' l_v''''''' \epsilon_{ipa}\epsilon_{jqb}$ $\epsilon_{krc}\epsilon_{hsd}\mathcal{H}^{pqrstuv} = 0_{abcd}$	16
$x^i x'^j x''^k l_s''' l_t'''' l_u''''' l_v''''''' \epsilon_{ipa}\epsilon_{jqb}\epsilon_{krc}\mathcal{H}^{pqrstuv} = 0_{abc}$	8
$x^i x'^j l_r'' l_s''' l_t'''' l_u''''' l_v''''''' \epsilon_{ipa}\epsilon_{jqb}\mathcal{H}^{pqrstuv} = 0_{ab}$	4
$x^i l_q' l_r'' l_s''' l_t'''' l_u''''' l_v''''''' \epsilon_{ipa}\mathcal{H}^{pqrstuv} = 0_a$	2
$l_p' l_q'' l_r''' l_s'''' l_t''''' l_u''''''' l_v''''''''\mathcal{H}^{pqrstuv} = 0$	1

$$\mathcal{H}^{pqrstuv} = \det\begin{bmatrix}\mathbf{a}^p\\ \mathbf{b}^q\\ \mathbf{c}^r\\ \mathbf{d}^s\\ \mathbf{e}^t\\ \mathbf{f}^u\\ \mathbf{g}^v\end{bmatrix} \tag{23}$$

where $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$, $\mathbf{d}^i$, $\mathbf{e}^i$, $\mathbf{f}^i$ and $\mathbf{g}^i$ denote the ith row of seven camera matrices. The septifocal tensor $\mathcal{H}^{pqrstuv}$ has 2187 entries, including 576 non-zero entries. Then we have 575 free parameters in $\mathcal{H}^{pqrstuv}$ except a scale. On the other hand, (22) provides us 2187 linear equations on $\mathcal{H}^{pqrstuv}$, but only 128 of them are linearly independent. Excluding the dependences between the corresponding points, 7 sets of corresponding points are enough to compute $\mathcal{H}^{pqrstuv}$ from images linearly. The septilinear relationships are given in Table 4.

4.3 Minimum Number of Points for Computing Multifocal Tensors

We next consider the minimum number of points required for computing the multifocal tensors. The 6D affine transformation has 42 DOF, since it is represented by a 7×7 matrix whose last row is [0 0 0 0 0 0 1]. Even if a set of N cameras are transformed by a single 6D affine transformation, their relative geometry does not change in a 6D affine space. Thus, the geometric DOF of N extended affine cameras is $14N-42$. Meanwhile, if we are given M points in the 6D space, and let them be projected to N cameras defined in (9). Then, we derive $2MN$ measurements from images, while we have to compute $14N - 42 + 6M$ components for fixing all the geometry in the 6D space. Thus, the following condition must hold for computing the multifocal tensors from images: $2MN \geq 14N - 42 + 6M$. We find that minimum of 7 points are required to compute multifocal tensors in four, five, six and seven views.

5. Applications on Multiple View Geometry of Curvilinear Motion Cameras

5.1 View Transfer

The constraints between corresponding points and multifocal tensors have been derived (see (12), (18), (20), (22)), and multifocal tensors can be computed by 7 corresponding points in 4 to 7 views. Thus, if we have the point trajectories in $N-1$ images, the trajectories in the remaining image can be calculated from N view tensor. It realizes the view transfer from $N-1$ views to the other view.

5.2 3D Reconstruction

From (9), if image points and extended camera matrix are given, the coordinates of points in 3D can be obtained. Therefore, computing the extended camera matrix is very important.

Assuming that the first viewpoint is at the origin, the camera matrices may now be written as:

$$\mathbf{P}'_1 = [\mathbf{I} \mid \mathbf{0}]$$
$$\mathbf{P}'_n = [\mathbf{H}_{1n} \mid \mathbf{e}_{n1}]$$

where $\mathbf{H}_{1n}$ denotes the 3×3 homography from the first view to the nth view, and $\mathbf{e}_{n1}$ denotes a 3×4 matrix which represents the epipole. Here, the epipole $\mathbf{e}_{n1}$ is not the traditional epipole which represents the projection of a 3D viewpoint to the 2D image, but the projection from 6D to 2D which is a 3D space, and the four column vectors in $\mathbf{e}_{n1}$ are four basis points of this 3D space [8]. The third rows of $\mathbf{P}'_1$ and $\mathbf{P}'_n$ are [0 0 1 0 0 0 0], and the extended affine camera matrices, $\mathbf{P}'_1$, $\mathbf{P}'_n$, can be derived as follows:

$$\mathbf{P}_1 = \mathbf{P}'_1\mathbf{L}'$$
$$\mathbf{P}_n = \mathbf{P}'_n\mathbf{L}'$$

where, $\mathbf{L}'$ is the following 7×7 matrix:

$$\mathbf{L}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Let us consider four views for instance. In (16), $x^i x'^j x''^k Q^v_{ijk}$ can be considered as a point, p'''^v. Then p'''^v and l_v''' have the following relation:

$$p'''^v l_v''' = 0. \tag{24}$$

That is, p'''^v is a point on the line l_v''' in the fourth view. If x^i, x'^j and x''^k are corresponding points, then p'''^v is also a corresponding point x'''^v in the fourth view. Thus, (16) may be rewritten as:

$$x^i x'^j x''^k Q^v_{ijk} = x'''^v. \tag{25}$$

Then, the following equations can be derived:

$$x'''^v = H^v_{14i} x^i \tag{26}$$

$$H^v_{14i} = x'^j x''^k Q^v_{ijk} \tag{27}$$

H^v_{14i} denotes a homography from the first view to the fourth view. If we have two pairs of x'^j and x''^k, two H^v_{14i} can be obtained:

$$H^v_{14i} = x'^j_1 x''^k_1 Q^v_{ijk} \tag{28}$$

$$H'^v_{14i} = x'^j_2 x''^k_2 Q^v_{ijk} \tag{29}$$

Thus, we have the following constraints:

$$\mathbf{e}_{41} = \mathbf{H}_{14}\mathbf{e}_{14} \tag{30}$$

$$\mathbf{e}_{41} = \mathbf{H}'_{14}\mathbf{e}_{14} \tag{31}$$

If $\mathbf{H}_{14}$ and $\mathbf{H}'_{14}$ are independent, we can obtain:

$$(\mathbf{H}_{14} - \mathbf{H}'_{14})\mathbf{e}_{14} = \mathbf{0} \tag{32}$$

Since $\mathbf{H}_{14}$ and $\mathbf{H}'_{14}$ have been figured out, epipole $\mathbf{e}_{14}$ can also be derived. However, here we only can derive one column vector in $\mathbf{e}_{14}$. For obtaining the other three column vectors, we need other three homography pairs. Once $\mathbf{e}_{14}$ and $\mathbf{H}_{14}$ are known, $\mathbf{e}_{41}$ can be calculated from (30). Thus, the camera matrix $\mathbf{P}_4$ can be computed from $\mathbf{H}_{14}$ and $\mathbf{e}_{41}$. $\mathbf{P}_2$ and $\mathbf{P}_3$ can also be derived in the same manner. Then, using $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ and a set of corresponding points in these camera images, we can reconstruct $\mathbf{X}$ in (9), and hence the point in 3D space and time T.

6. Experiments

We next show the results of some experiments. We first show that the quadrifocal tensor for extended affine cameras can be computed from point trajectories viewed from arbitrary curvilinear motion cameras, and can be used for generating one view from the others and for recovering 3D motions. We next evaluate the stability of extracted quadrifocal tensors for extended affine cameras. We also discuss the approximate relationship between affine cameras and projective cameras. We finally show the results from real images taken from moving cameras.

6.1 Synthetic Image Experiment

6.1.1 View Transfer

We firstly show view transfer experiment by using synthetic images.

Figure 2 shows a 3D configuration of 4 moving cameras and a moving point. The black points show the viewpoints of four cameras, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ and $\mathbf{C}_4$, with B-spline motions which consist of two B-spline segments. The curvilinear motions of these four cameras are different and unknown. The black curve shows a locus of a moving point

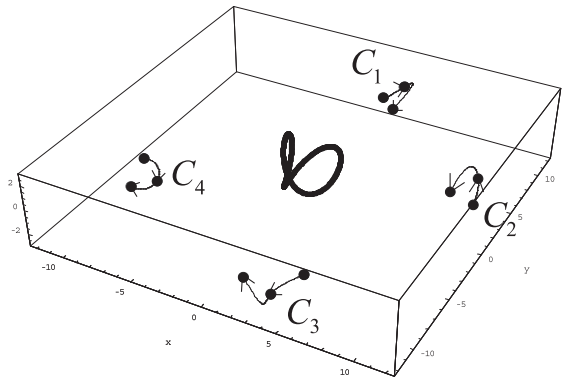

Fig. 2 Four curvilinear motion cameras and a moving point in the 3D space. Each camera motion consists of two B-spline segments.

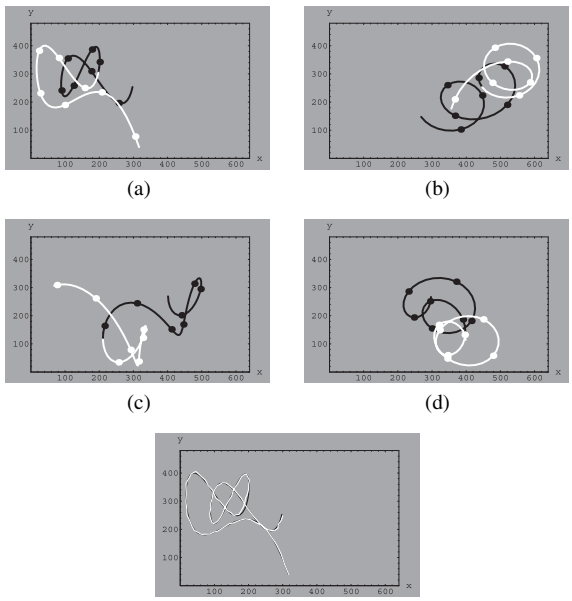

Fig. 3 View transfer. Figure (a), (b), (c) and (d) show four views of the motion in camera 1, 2, 3 and 4. The black curves represent the point trajectories when the cameras undergo the first curvilinear motions. The white curves correspond to the second camera motions. The 7 black points on each black loci and the 7 white points on each white loci in Fig. 3 (a), (b), (c) and (d) are used to compute the two quadrifocal tensors. The white curve in (e) shows point trajectories computed by the extended quadrifocal tensors in the image plane of camera 1. The black curve is the true value.

S. Figure 3 (a), (b), (c) and (d) show point trajectories of **S** viewed from $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ and $\mathbf{C}_4$ respectively. Note, the original locus of **S** is closed in the 3D space as shown in Fig. 2, but its loci in images are not closed as shown in Fig. 3 because of the camera motions. We added Gaussian image noises with the standard deviation of 1 pixel to all the points on the loci in images. The 7 black points on the black loci and the 7 white points on the white loci in Fig. 3 (a), (b), (c) and (d) are used to compute the two quadrifocal tensors on these four moving cameras with two B-spline motions. The quadrifocal tensors are used to recover the point trajectories in $\mathbf{C}_1$ from $\mathbf{C}_2$, $\mathbf{C}_3$ and $\mathbf{C}_4$. Figure 3 (e) shows the recovered

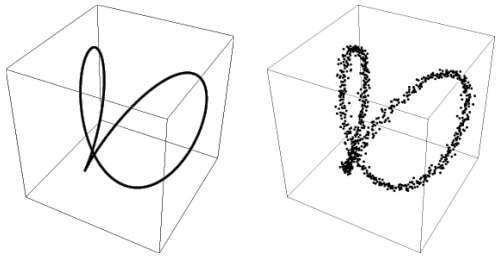

Fig. 4 3D reconstruction. Figure (a) shows the real 3D motion. Figure (b) shows the result of 3D reconstruction.

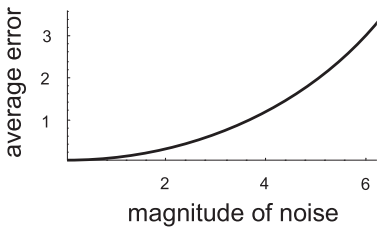

Fig. 5 The relationship between the noise level and the average error in 3D reconstruction.

result. The black curve shows the real trajectory, and the white curve shows the computed motion. The average error of the recovered point trajectories is 6.03 pixels.

6.1.2 3D Reconstruction

We next show the results of 3D reconstruction using the 3D configurations shown in Fig. 2 to verify another application, 3D reconstruction. However, for convenience, we assumed camera $\mathbf{C}_1$ a static camera in this experiment. Figure 4 (a) shows the real 3D motion trajectory whose range is from -2 to 2 in X, Y, Z axes respectively. The corresponding points with Gaussian noise of standard deviation 1 pixel in the four images were used to figure out the coordinates of each point in 3D space. The reconstructed result is shown in Fig. 4 (b). The average error is 0.21. We can see the shape of the 3D motion is recovered properly. Figure 5 shows the relationship between the noise level and the average error.

6.2 Stability Evaluation

We next show the stability of extracted quadrifocal tensors under extended projections. For evaluating the extracted quadrifocal tensors, we computed reprojection errors derived from the quadrifocal tensors. The reprojection error is defined as: $\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$, where $d(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ denotes a distance between a true point $\mathbf{m}_i$ and a point $\hat{\mathbf{m}}_i$ recovered from the quadrifocal tensor. We increased the number of corresponding points used for computing quadrifocal tensors in four views from 7 to 20, and evaluated the reprojection errors. The camera motions are represented by single B-spline curve segments. Camera trajectories and 3D point motions are generated for 1000 times by changing the control points

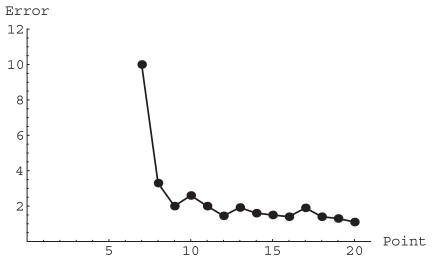

Fig. 6 The relationship between the number of corresponding points used for computing quadrifocal tensors and the reprojection errors. Camera trajectories and 3D point motions are generated for 1000 times by changing the control points of the B-spline curves.

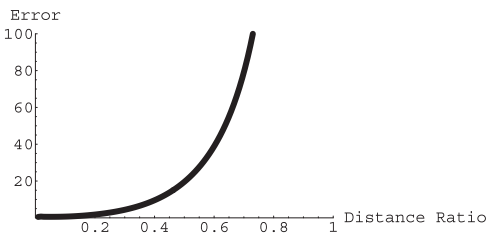

Fig. 7 The relationship between the distance ratio and the reprojection errors under the projective camera model.

of the B-spline curves. Each camera trajectory and 3D point are added Gaussian noise 100 times with the standard deviation of 1 pixel. Figure 6 shows the relationship between the number of corresponding points and the reprojection errors. As we can see, the stability is obviously improved by using a few more points than the minimum number of corresponding points.

6.3 Approximate Relationship between Affine Camera and Projective Camera

Affine camera is an ideal model whose optical center is at infinity. It does not exist in the real world. Therefore, we here desire to find some clue to the approximate relationship between affine camera and the most general camera model, projective camera.

We consider a ratio between the "radius" of the 3D motion (the average distance between the center and the boundary of the motion) and the distance between motion's center and projective camera, which we call *distance ratio*. The relationship between distance ratio and reprojection error (its definition is same to stability evaluation) is shown in Fig. 7. The image size is 640×480. As we can see, when distance ratio ≤ 0.4, reprojection error is less than 10.

6.4 Real Image Experiment

In the first experiment, we used four usual cameras, all of which have different one segment B-spline curve motions, and computed quadrifocal tensors among these 4 cameras by using two moving points in the 3D space. Figure 8 (a), (b), (c) and (d) show trajectories of two points viewed from

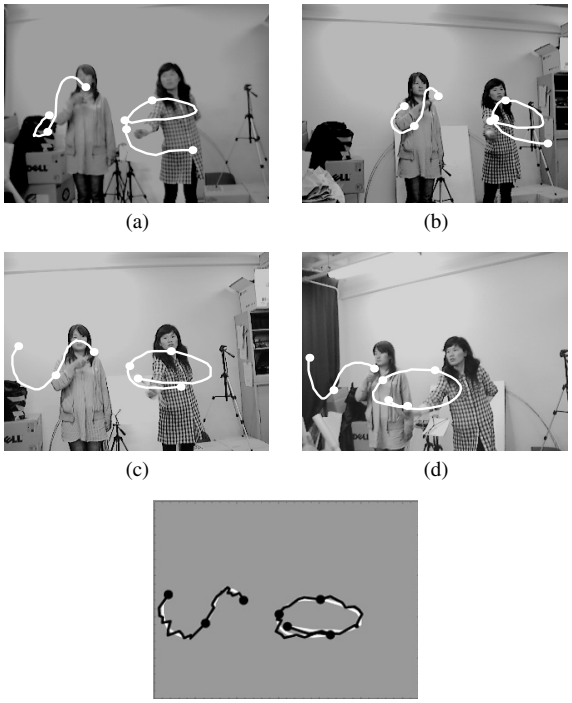

Fig. 8 Multiple point motion experiment. Figures (a), (b), (c) and (d) show four views of the motion in camera 1, 2, 3 and 4. The white curves show two different point trajectories in each view. The 7 white points on the two curves in each image are corresponding points used for computing the quadrifocal tensor. Note that all the cameras have one segment B-spline curve motion. The black curves in (e) show trajectories computed by the extended quadrifocal tensor in camera 3. The white curve shows the real point motion.

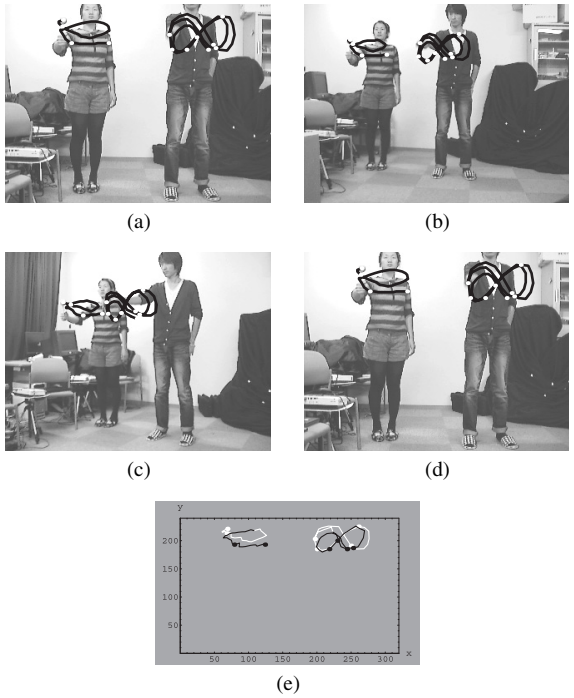

Fig. 9 Multiple point motion experiment. Figures (a), (b), (c) and (d) show four views of the motions in camera 1, 2, 3 and 4. The cameras undergo two segment B-spline curve motions. The black curves show two different point trajectories in each view. The white points on the two curves in each image are corresponding points used for computing the quadrifocal tensor. The curves in (e) show trajectories computed by the extended quadrifocal tensors in camera 1. The 7 white points and 7 black points are corresponding points to be used to figure out the quadrifocal tensor respectively.

camera 1, 2, 3 and 4. Here, distance ratio is about 0.25. Such configuration could be considered approximating with affine camera models as addressed. The white curves represent two different point trajectories. The 7 white points on the two curves in each image are corresponding points used for computing the quadrifocal tensor. The black curves in (e) show trajectories computed from the extended quadrifocal tensor in camera 3. The average error of the recovered trajectories is 7.5 pixels. The error is caused by the following reasons according to our analysis: (1) camera motion error. It is difficult for controlling four cameras to do rigorous spline curve motions; (2) approximate error. We used projective cameras to approximate affine cameras; (3) selection of corresponding points. Since coplanar corresponding points may arise degeneration, correct results are derived from non-coplanar corresponding points.

We next show the result from the case when cameras undergo two segment B-spline curve motions. We still observed two moving points and computed quadrifocal tensors among four cameras. Figure 9 (a), (b), (c) and (d) show trajectories of two points viewed from camera 1, 2, 3 and 4. Here, distance ratio is about 0.20. Since the camera trajectories are two segment B-spline curves, we calculated two quadrifocal tensors to realize view transfer from camera 2, 3 and 4 to camera 1. The result is shown as Fig. 9 (e). White curves are derived from the first quadrifocal tensor, and black curves are computed from the second quadrifocal tensor. The 7 white points and 7 black points are corresponding points to be used to figure out the quadrifocal tensor respectively. The average error is 6.2 pixels.

From the experiments, we can see that the quadrifocal tensor defined under extended projection can be derived from multiple point motions viewed from the 4 cameras with curvilinear motions, and they are practical for generating images of multiple point motions viewed from curvilinear motion camera.

7. Conclusion

In this paper, we showed that a multilinear relationship under the projection from 6D to 2D can represent the geometric relationship of multiple curvilinear motion cameras whose motions are represented by cubic B-spline curves. The multifocal tensors defined under 6D to 2D multilinear relationships can be computed from non-rigid object motions viewed from multiple cameras with arbitrary curvilinear motions. We also showed that the multilinear relationships are very useful for generating images and reconstructing 3D non-rigid motions viewed from cameras with arbitrary curvilinear motions. The method was implemented

and tested by using real image sequences. The stability of extracted quadrifocal tensors was also evaluated. The recovered extended affine camera matrices include camera motion information. However, the decomposition of the camera matrices is our future work.

References

[1] C. Tomasi, "Shape and motion from image streams under orthography: A factorization method," Int. J. Comput. Vis., vol.9, pp.137–154, 1992.
[2] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.
[3] O. Faugeras and Q. Luong, The Geometry of Multiple Images, MIT Press, 2001.
[4] P. Sturm, "Multi-view geometry for general camera models," Proc. Conference on Computer Vision and Pattern Recognition, pp.206–212, 2005.
[5] R.I. Hartley and F. Schaffalitzky, "Reconstruction from projections using Grassmann tensors," Int. J. Comput. Vis., vol.83, no.3, pp.274–293, 2009.
[6] Y. Wexler and A. Shashua, "On the synthesis of dynamic scenes from reference views," Proc. Conference on Computer Vision and Pattern Recognition, pp.576–581, 2000.
[7] S. Avidan and A. Shashua, "Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, pp.348–357, 2000.
[8] L. Wolf and A. Shashua, "On projection matrices $p^k \rightarrow p^2$, $k = 3, \cdots, 6$, and their applications in computer vision," Proc. 8th International Conference on Computer Vision, pp.412–419, 2001.
[9] C. Wan and J. Sato, "Multiple view geometry under projective projection in space-time," IEICE Trans. Inf. & Syst., vol.E91-D, no.9, pp.2353–2359, Sept. 2008.
[10] C. Wan and J. Sato, "Computing multiple view geometry in space-time from mutual projections of multiple cameras," Proc. 19th International Conference on Pattern Recognition, pp.1–4, 2008.
[11] C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978.
[12] R. Vidal and D. Abretske, "Nonrigid shape and motion from multiple perspective views," European Conference on Computer Vision, pp.205–218, 2006.
[13] H.S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3d reconstruction of a moving point from a series of 2d projections," European Conference on Computer Vision, pp.158–171, 2010.

Cheng Wan received the B.Eng. degree from PLA University of science and technology, China in 2001, and the Ph.D. degree in information engineering from Nagoya Institute of Technology, Japan in 2011. She is presently a teacher at the College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Her research interests include computer vision and visual geometry. She was awarded the Special Session Prize of Meeting on Image Recognition and Understanding, Japan in 2007.

Jun Sato received the B.Eng. degree from Nagoya Institute of Technology, Japan in 1984, and the Ph.D. degree in information engineering from the University of Cambridge, England in 1997. From 1996 to 1998 he was a research associate at the Department of Engineering, University of Cambridge. During 1997–1998, he was an invited researcher at the ATR Human Information Processing Research Laboratories, Kyoto. He joined the Department of Electrical and Computer Engineering, Nagoya Institute of Technology in 1998 as an associate professor. He is currently a professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology. His research interests include computer vision, geometric invariants, visual navigation, visual interface and mixed reality. He was awarded the Best Science Paper Prize of BMVC twice in 1994 and 1997, the Best Paper Prize of SSII'99 and the Special Session Prize of MIRU'07, etc. He is a member of the IEEE and IPSJ.