Specifying Virtual Cameras in Uncalibrated View Synthesis

A. Fusiello *Member, IEEE,*

***Abstract*— This paper deals with the views synthesis problem and proposes an automatic method for specifying the virtual camera position and orientation in an uncalibrated setting, based on the interpolation and extrapolation of the motion among the reference views. Novel images can be rendered from virtual cameras moving on parametric trajectories. Synthetic and real experiments illustrate the approach.**

I. Introduction

View synthesis consists in rendering images of a scene as if they were taken from a virtual viewpoint different from all the viewpoints of the real views. It is an instance of *Image-Based Rendering* (IBR): While the traditional geometry-based rendering starts from a 3-D model, in IBR views are generated by re-sampling one or more example images, using appropriate warping functions (see [1] for a review). The advantage is that photographs of real scenes can be used as a basis to create very realistic images, and rendering time is decoupled from the complexity of the scene.

The warping functions are based on geometric relationships that are found between the positions of pixels representing the same point in the scene observed from different viewpoints [2]. For example, given the internal and external parameters of the camera, and the depth of a scene point (with respect to the camera), it is easy to obtain the position of the point in any synthetic view [3]. In the case of calibrated cameras, algorithms based on image interpolation yield satisfactory results [4], [5], [6]. Where no knowledge on the imaging device can be assumed, uncalibrated point transfer techniques utilize image-to-image constraints such as the fundamental matrices [7], trilinear tensors [8], plane+parallax [9], or homographies [10], to re-project pixels from a small number of reference images to a given view. Another way of linking corresponding points is the *relative affine structure*, a close relative of the plane+parallax, introduced by [11], [12].

Although uncalibrated point transfer algorithms are well understood, what prevent them to be applied in real-world applications is the lack of a “natural” way of specifying the *pose* (position and orientation) of the virtual camera in the familiar Euclidean frame, because this is not accessible. Everything is represented in a projective frame that is linked to the Euclidean one by an *unknown* projective transformation. All the uncalibrated view-synthesis algorithms requires to specify some projective elements, like epipoles, homographies, fundamental matrices or tensors. With the exception of few papers, previous work concentrates on the generation of the novel view, assuming that the problem of view specification is solved somehow. For example, in [5], [13], the user has to manually specify the position of four points in each frame of the synthetic sequence. The view specification problem is addressed only in [13], [14], where the internal parameters are assumed to be approximately known, thereby violating the assumption of uncalibrated camera.

A. Fusiello is with the Dipartimento di Informatica, Università degli Studi di Verona, I-37134 Verona, Italy. e-mail: andrea.fusiello@univr.it. phone: +39 045 8027088. fax +39 045 8027068.

A. Contribution

This paper tackles the problem of view specification with uncalibrated images. Our solution is a technique that enables a natural, easy-to-use and transparent way of dealing with the problem of posing the virtual camera in the *uncalibrated stratum*, as opposed to the familiar Euclidean stratum. The synthetic views are physically-valid, as in [5], but our method is not limited to in-between views, as it caters for interpolation *and* extrapolation within the same framework. We will consider here the case of two or three reference views, but the method can cope with an arbitrary number of reference views.

Our algorithm yields a parametric family of camera poses that describes a smooth trajectory in the Euclidean space as the parameters vary continuously. These trajectories interpolate between the poses of the reference cameras and extrapolate them by replicating the rigid motion among the reference views. To the best of our knowledge, this is the first solution which does not require manual input from a human operator and does not make any assumption on the internal parameters of the camera.

The method is based on an uncalibrated description of the rigid motion in terms of epipole and homography of the plane at infinity, which is isomorphic to the familiar Euclidean rigid motion. This enables to map operations transparently from the Euclidean to the uncalibrated stratum. In particular, we build upon the framework for linear combination of similarity transformations set forth by [15].

The idea of manipulating rigid motions in the uncalibrated stratum was also outlined in [16] for a robotic application, but it was limited to rotations only. A preliminary version of this work appeared in [17].

B. Synopsis

The rest of the paper is structured as follows. In Section II we review some background notions required to make the paper self-consistent. Section III summarizes points transfer using the relative affine structure, while Section IV reviews the framework for the linear combination of rigid motions.

The novel contribution is contained in Section V and VI. In the former, the structure of the uncalibrated rigid transformations is investigated and related to the Euclidean stratum. In the latter, the application to the specification of virtual poses is described. Albeit this paper does not aim at describing a complete system for view synthesis, some examples illustrating the approach are reported in Section VII. Finally, conclusions are drawn in Section VIII.

II. Background theory

In this section we review some background notions and provide a straightforward derivation of the relative affine structure framework.

A. Collineations and two-views geometry

We start by reviewing the geometry of two views. A more complete discussion can be found, for example, in [18].

If we take the first camera reference frame as the world reference frame, we can write the following two general camera matrices:

$$P = K[I|\mathbf{0}] = [K|\mathbf{0}] \tag{1}$$

$$P' = K'[R|\mathbf{t}] \tag{2}$$

where R is a 3×3 rotation matrix, $\mathbf{t}$ is a 3×1 translation vector, representing the rigid motion that brings the second camera reference frame onto the world reference frame, K is an upper-triangular matrix that contains the internal camera's parameters.

Definition 2.1: Two points $\mathbf{m}$ and $\mathbf{m}'$ that are the projection of the same 3-D point $\mathbf{M}$ onto the first and the second camera, respectively, are said to be *corresponding points*.

Proposition 2.2: Two corresponding points $\mathbf{m}$ and $\mathbf{m}'$ are related by

$$\zeta'\mathbf{m}' = \zeta K'RK^{-1}\mathbf{m} + K'\mathbf{t}. \tag{3}$$

where ζ and ζ' are the distance of $\mathbf{M}$ from the focal plane of the first and second camera respectively.

Proof: Insert (1) and (2) into the perspective projection equations $\zeta\mathbf{m} = P\mathbf{M}$ and $\zeta'\mathbf{m}' = P'\mathbf{M}$ and eliminate $\mathbf{M}$. ■

In very special cases views are related by a *collineation* (or *homography*), i.e, a non-singular linear transformation of the projective plane into itself, represented by a non-singular 3×3 matrix.

Proposition 2.3: If the scene is described by a 3-D plane Π with equation $\mathbf{n}^\mathsf{T}\mathbf{M} = d$, then two views of such a scene are related by:

$$\frac{\zeta'}{\zeta}\mathbf{m}' = H_\Pi \mathbf{m}. \tag{4}$$

where

$$H_\Pi \triangleq K'RK^{-1} + K'\mathbf{t}\frac{\mathbf{n}^\mathsf{T}}{d}K^{-1}. \tag{5}$$

is the collineation induced by plane Π.

Proof: Specialize (3) with $\mathbf{M} \in \Pi$. ■

If $d \to \infty$ in (5) we obtain the homography for the plane at infinity: $H_\infty \triangleq K'RK^{-1}$.

The most general homography matrix has eight degrees of freedom, being defined up to a scale factor. Therefore, four corresponding points in the two views define a homography.

Definition 2.4: The epipole in the second view, denoted by $\mathbf{e}'$, is the projection of the first camera's center onto the second camera. Similarly, the epipole in the first view $\mathbf{e}$ is defined as the projection of the second camera's center onto the first camera.

In our case,

$$\mathbf{e}' = K'[R|\mathbf{t}][0, 0, 0, 1]^\mathsf{T} = K'\mathbf{t}. \tag{6}$$

Hence, (3) can be re-written as:

$$\zeta'\mathbf{m}' = \zeta H_\infty \mathbf{m} + \mathbf{e}'. \tag{7}$$

B. Relative Affine structure

Equation (7) relates two corresponding points using the homography of the plane at infinity. An interesting formulation of the two-views geometry is obtained by taking a generic plane as the reference one instead of the infinity plane. This leads to the relative affine structure [12] or plane+parallax [9] formulation.

Proposition 2.5: Given a plane Π, with equation $\mathbf{n}^\mathsf{T}\mathbf{M} = d$, two corresponding points $\mathbf{m}$ and $\mathbf{m}'$ are related by

$$\frac{\zeta'}{\zeta}\mathbf{m}' = H_\Pi\mathbf{m} + \mathbf{e}'\left(\frac{a}{d\,\zeta}\right) \tag{8}$$

where $a \triangleq d - \mathbf{n}^\mathsf{T}\zeta K^{-1}\mathbf{m}$ is the orthogonal distance of the 3-D point $\mathbf{M}$ (of which $\mathbf{m}$ and $\mathbf{m}'$ are projections) to the plane Π, and ζ and ζ' are the distance of $\mathbf{M}$ from the focal plane of the first and second camera respectively.

Proof: Substitute $H_\infty = H_\Pi - K'\mathbf{t}\frac{\mathbf{n}^\mathsf{T}}{d}K^{-1}$ in (7). ■

If $\mathbf{M} \in \Pi$, then (8) reduces to (4). Otherwise, there is a residual displacement, called *parallax*, proportional to the *relative affine structure* $\gamma \triangleq \frac{a}{d\,\zeta}$ of $\mathbf{M}$, with respect to the plane Π [12].

This equation tells us that points are first transferred as if they were lying on the reference plane Π, and then their position is corrected by a displacement in the direction of the epipole, with magnitude proportional to the relative affine structure. It is worth noting that:

- The relative affine structure is independent of the choice of the second view.
- It is related to the disparity, being proportional to the inverse depth. Indeed, when the reference plane is the plane at infinity, the relative affine structure reduces to $\gamma = \frac{1}{\zeta}$.
- Points $\mathbf{m}'$, $H_\Pi\mathbf{m}$ and $\mathbf{e}'$ are collinear. The parallax field is a radial field centered on the epipole.

The latter property can be used to locate the epipole [19], given the homography between two views and two off-plane corresponding pairs $(\mathbf{m}_0; \mathbf{m}'_0)$ and $(\mathbf{m}_1; \mathbf{m}'_1)$. Following simple geometric considerations[1], the epipole is computed as

[1] In the projective plane, the line determined by two points is given by their cross product, as well as the point determined by two lines.

the intersection between the line containing $H_\Pi \mathbf{m}_0, \mathbf{m}'_0$ and the line containing $H_\Pi \mathbf{m}_1, \mathbf{m}'_1$:

$$\mathbf{e}' \simeq (H_\Pi \mathbf{m}_0 \times \mathbf{m}'_0) \times (H_\Pi \mathbf{m}_1 \times \mathbf{m}'_1) \tag{9}$$

where the symbol $\simeq$ means equality up to a scale factor.

III. Point transfer with relative affine structure

Since the relative affine structure is invariant on the choice of the second view, arbitrary "second views" can be synthesized, by giving a plane homography and an epipole, which specify the position and orientation of the virtual camera in a projective frame.

We will turn now to a notation for three (or more) views, and will use the subscripts $1,2,3$ to designate them. A transformation denoted by H_{ij} will always map quantities of view i to quantities of view j. An element present in view i will be denoted by $\mathbf{m}_i$. Likewise, an element present in view i in relation with view j will be denoted by $\mathbf{e}_{ij}$. Superscripts will be used to index elements in a set, like in $\mathbf{m}_i^k \quad k = 1, \dots, m$.

When working in the uncalibrated stratum, care must be taken to properly deal with projective scale factors. For example H_{12} and $\mathbf{e}_{21}$ computed from corresponding points are only defined up to a scale factor. In this context, we must rewrite (8) as:

$$\mathbf{m}_2^k \simeq H_{12}\mathbf{m}_1^k + \mathbf{e}_{21}\gamma_1^k. \tag{10}$$

and keep in mind that the sum makes sense only if the scale factors of H_{12} and $\mathbf{e}_{21}$ has been fixed. From now on we will assume, unless otherwise specified, that the epipole is represented by a unit vector. The scale of the homography is fixed by forcing a chosen point to have unit relative affine structure.

Given a certain number of corresponding pairs $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k) \quad \forall k = 1, \dots, m$ their relative affine structure is obtained by solving for γ_1^k in (10):

$$\gamma_1^k = \frac{(\mathbf{m}_2^k \times \mathbf{e}_{21})^\mathsf{T}(H_{12}\mathbf{m}_1^k \times \mathbf{m}_2^k)}{||\mathbf{m}_2^k \times \mathbf{e}_{21}||^2}. \tag{11}$$

The view synthesis algorithm that we employ, inspired by [12], is the following.

GeneralViewSynthesis

1) given a set of corresponding pairs $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k) \quad k = 1, \dots, m$;
2) recover the epipole $\mathbf{e}_{21}$ and the homography H_{12};
3) choose a point $\mathbf{m}_1^0$ and scale H_{12} to satisfy $\mathbf{m}_2^0 \simeq H_{12}\mathbf{m}_1^0 + \mathbf{e}_{21}$;
4) compute the relative affine structure γ_1^k with (11);
5) obtain a new epipole $\mathbf{e}_{31}$ and a new homography H_{13} (properly scaled);
6) transfer points in the synthetic view with $\mathbf{m}_3^k \simeq H_{13}\mathbf{m}_1^k + \mathbf{e}_{31}\gamma_1^k$

The problem that makes this technique difficult to use in practice (and for this reason it has been overlooked for view synthesis) is point (5), namely the need to specify a new epipole $\mathbf{e}_{31}$ and a new (scaled) homography H_{13}.

In Section VI we will present a solution to this problem that provides an easy and transparent way of specifying the virtual camera's viewpoint in an projective frame. The solution builds upon the concept of *linear combination* of rigid motions in the Euclidean frame [15], which we shall recapitulate in the next Section.

IV. Linear combination of rigid motions

The linear combination of rigid motions has been defined by [15] as a tool for for dealing with geometric transformations in Computer Graphics. It is based on the concepts of *scalar multiple* and *commutative composition* of rigid motions.

Let $\mathrm{SE}(3,\mathbb{R})$ denote the *special Euclidean group*, and let $G \in \mathrm{SE}(3,\mathbb{R})$ be a 4×4 matrix that represents a rigid motion.

1) Scalar multiple: Thanks to the group structure of $\mathrm{SE}(3,\mathbb{R})$, integer multiples of a rigid motion are well defined as G^z for $z \in \mathbb{Z}$. However, $\mathrm{SE}(3,\mathbb{R})$ is also a differentiable manifold (it is a Lie group), hence we will be able to define *scalar multiples* G^t for $t \in \mathbb{R}$.

Let us consider, without loss of generality, the problem of interpolating between the element $G \in \mathrm{SE}(3,\mathbb{R})$ and the identity. On a differentiable manifold we can make sense of the interpolation between two elements as drawing the geodesic path between them [20] (the geodesic path is the shortest path between two points in a curved space).

The geodesic path in a neighborhood of I can be obtained as the projection onto $\mathrm{SE}(3,\mathbb{R})$ of a straight path in the tangent space; the mapping that projects from $\mathrm{SE}(3,\mathbb{R})$ to the tangent space at I is the matrix logarithm. The matrix logarithm is briefly introduced in the Appendix, where its existence for the matrices of interest in this work is discussed. Conversely, a straight path in the tangent space emanating from 0 is mapped onto a geodesic in $\mathrm{SE}(3,\mathbb{R})$ emanating from I by the exponential map. Hence, the geodesic path in $\mathrm{SE}(3,\mathbb{R})$ joining I and G is given by

$$G^t \triangleq \exp(t\log(G)), \quad t \in [0,1]. \tag{12}$$

This definition has some good properties:

- If G represent a rotation by an angle θ around axis $\mathbf{u}$, then G^t is a rotation of angle $t\theta$ around the same axis (this can be proved using Rodrigues' formula).
- In the case of a matrix G representing a pure translation by a vector $\mathbf{v}$, G^t represents a translation by $t\mathbf{v}$.
- The half transformation is well-defined, i.e, $G^{\frac{1}{2}}G^{\frac{1}{2}} = G$.

More in general:

Definition 4.1: Let $G \in \mathrm{SE}(3,\mathbb{R})$. The *scalar multiple* of G is defined as:

$$t \odot G \triangleq G^t \triangleq \exp(t\log(G)), \quad t \in \mathbb{R}. \tag{13}$$

The scalar multiple coincides with the integer multiple for $t \in \mathbb{Z}$, and produces a smooth interpolant (along the geodesic) for all the other values of t.

2) Commutative composition: Let us now consider the composition of two rigid motions G_1 and G_2. We would like to be able to define a *commutative composition* (or addition) such that G_1 followed by G_2 ends up in the same place as G_2 followed by G_1.

Definition 4.2: Let $G_1, G_2 \in \mathrm{SE}(3,\mathbb{R})$. The *commutative composition* of G_1, G_2 is defined as:

$$G_1 \oplus G_2 \triangleq e^{\log G_1 + \log G_2} \tag{14}$$

The operation is indeed commutative because the sum in the exponent is commutative. The commutative composition behaves much like the standard matrix product. If G_1 and G_2 commute, then $G_1 \oplus G_2 = e^{\log G_1} e^{\log G_2} = G_1 G_2 = G_2 G_1$. Since G_1 commutes with G_1^{-1}, then $G_1 \oplus G_1^{-1} = I$

The intuitive meaning of the commutative composition can be grasped thanks to the relationship:

$$\exp(G_1 + G_2) = \lim_{n\to\infty} \left(\exp(\tfrac{1}{n}G_1)\exp(\tfrac{1}{n}G_2)\right)^n \tag{15}$$

from which it is easily obtained that

$$G_1 \oplus G_2 = \lim_{n\to\infty} (G_1^{\frac{1}{n}} G_2^{\frac{1}{n}})^n. \tag{16}$$

In a sense, $G_1 \oplus G_2$ is like applying G_1 and G_2 simultaneously, as we are slicing infinitely many times G_1 and G_2 and then applying these slices alternately.

3) Linear combination: Using the scalar multiple and the commutative composition we can do a weighted combination of two rigid motions:

Definition 4.3: Let $G_1, G_2 \in \mathrm{SE}(3,\mathbb{R})$. The *linear combination* of G_1, G_2 is defined as:

$$(u \odot G_1) \oplus (v \odot G_2) = e^{u \log G_1 + v \log G_2} \tag{17}$$

To be rigorous, this operation corresponds to a linear combination in the logarithm space. The linear combination of two independent motions spans a 2-manifold of $\mathrm{SE}(3,\mathbb{R})$.

V. The group of uncalibrated rigid motions

In this section we will first derive a description of a rigid motion that can be achieved when cameras are not calibrated (*uncalibrated motion*), resting on the knowledge of the epipole and the homography of the plane at infinity. Then we will draw its relationship with the Euclidean description of rigid motion, the special Euclidean group $\mathrm{SE}(3,\mathbb{R})$.

We aim to obtain an equation relating views 1-3 in terms of 1-2 and 2-3. To this end, let us consider (8), which expresses the epipolar geometry with reference to a plane, in the case of view pair 1-2:

$$\frac{\zeta_2}{\zeta_1}\mathbf{m}_2 = H_{12}\mathbf{m}_1 + \mathbf{e}_{21}\gamma_1 \tag{18}$$

and view pair 2-3:

$$\frac{\zeta_3}{\zeta_2}\mathbf{m}_3 = H_{23}\mathbf{m}_2 + \mathbf{e}_{32}\gamma_2. \tag{19}$$

By substituting the first into the second, we obtain:

$$\frac{\zeta_3}{\zeta_1}\mathbf{m}_3 = H_{23}H_{12}\mathbf{m}_1 + (H_{23}\mathbf{e}_{21} + \mathbf{e}_{32}\frac{d_1}{d_2})\gamma_1 \tag{20}$$

where d_1 and d_2 are the distances of the plane Π from the first and the second camera respectively. Comparison with (8) yields:

$$H_{13} = H_{23}H_{\Pi 12} \quad \text{and} \quad \mathbf{e}_{31} = H_{23}\mathbf{e}_{21} + \mathbf{e}_{32}\frac{d_1}{d_2} \tag{21}$$

The ratio d_1/d_2 in general is unknown, but if Π is the plane at infinity then $d_1/d_2 = 1$ (please note that this is approximately true for planes distant from the camera). Therefore, taking the plane at infinity as Π, (20) writes:

$$H_{\infty 13} = H_{\infty 23}H_{\infty 12} \quad \text{and} \quad \mathbf{e}_{31} = H_{\infty 23}\mathbf{e}_{21} + \mathbf{e}_{32} \tag{22}$$

Albeit, in general, homographies can be computed only up to a scale factor, in the case of the infinity plane homography, if internal parameters are assumed constant (as we do henceforth), the scale is fixed by the requirement that $\det(H_\infty) = 1$.

Definition 5.1: Let $\mathbf{e}_{ji}$ and $H_{\infty ij}$ be the epipole and the plane at infinity, respectively, linking two cameras i and j. The matrix

$$D_{ij} \triangleq \begin{bmatrix} H_{\infty ij} & \mathbf{e}_{ji} \\ \mathbf{0} & 1 \end{bmatrix} \tag{23}$$

is called the *uncalibrated rigid motion matrix*.
As opposed to a Euclidean rigid motion matrix, D_{ij} contains the homography of the plane at infinity in place of the rotation, and the epipole in lieu of the translation.

In matrix form (22) writes:

$$D_{13} = D_{23}D_{12} \tag{24}$$

Interestingly enough, uncalibrated rigid motion matrices D_{ij} follow the same multiplicative composition rule as the homogeneous rigid motion matrices G_{ij} of $\mathrm{SE}(3,\mathbb{R})$. This is something that [21] noted, but it has never been exploited, to the best of our knowledge. In a sense, D_{ij} is a homogeneous representation of the rigid motion at the uncalibrated stratum[2]. This observation leads to the following:

Proposition 5.2: The uncalibrated rigid motions form a group that is isomorphic to $\mathrm{SE}(3,\mathbb{R})$, under the assumption of constant internal parameters K.

Proof: Let

$$G_{ij} \triangleq \begin{bmatrix} R_{ij} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3,\mathbb{R}) \tag{25}$$

be a matrix that represent a rigid motion, where R is a rotation matrix and $\mathbf{t}$ is a vector representing a translation.

First, let us observe that the operator $\varphi_K : \varphi_K(G_{ij}) = D_{ij}$ that maps calibrated operations into the uncalibrated stratum, where the infinity plane homography substitutes the rotation and the epipole substitutes the translation, is a conjugacy map:

$$\varphi_K(G_{ij}) = D_{ij} = \begin{bmatrix} KR_{ij}K^{-1} & K\mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix} = \tilde{K}G_{ij}\tilde{K}^{-1} \tag{26}$$

with

$$\tilde{K} = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{27}$$

Then, it is easy to shown that φ_K is an homomorphism:

$$\begin{aligned}\varphi_K(G_{23})\varphi_K(G_{12}) = \tilde{K}G_{23}\tilde{K}^{-1}\tilde{K}G_{12}\tilde{K}^{-1} = \\ = \tilde{K}G_{23}G_{12}\tilde{K}^{-1} = \varphi_K(G_{23}G_{12})\end{aligned} \tag{28}$$

and, being φ_K invertible, it is an isomorphism. ■

Thanks to the fact that uncalibrated motions are isomorphic to $\mathrm{SE}(3,\mathbb{R})$, we can map the definitions of the previous section

[2]Technically, since we assume to know the plane at infinity, this correspond to the affine calibration stratum [21].

onto the uncalibrated motions. Every operation carried out in the uncalibrated stratum reflects itself consistently in the Euclidean stratum, even if the map φ_K is unknown.

VI. SPECIFYING THE VIRTUAL CAMERA POSITION

Let us focus on the main contribution of this paper, namely an easy and transparent way of specifying the virtual camera's viewpoint in the uncalibrated stratum.

Using the infinity plane as the reference plane, the transfer equation that allows to render the virtual view I_3 becomes:

$$\mathbf{m}_3^k \simeq H_{\infty 13}\mathbf{m}_1^k + \mathbf{e}_{31}\gamma_1^k = [I|\mathbf{0}]D_{13}\begin{bmatrix}\mathbf{m}_1^k\\ \gamma_1^k\end{bmatrix} \tag{29}$$

The virtual viewpoint is specified through the uncalibrated rigid motion matrix D_{13}.

A. Motion along a 1-d manifold

Let $D = \varphi_K(G),\ G \in \mathrm{SE}(3,\mathbb{R})$; echoing the definition of the *scalar multiple* of a rigid motion, let us define

$$t \odot D \triangleq D^t = \exp(t\log(D)), \quad t \in \mathbb{R}. \tag{30}$$

It is easy to show, using Proposition 5.2, that $t\odot D = \varphi(t\odot G)$.

Starting from D_{12}, which is known, let us compute

$$D_{13}(t) = t \odot D_{12} \tag{31}$$

and plug it into (29). As t varies in $\mathbb{R}$ we obtain a 1-parameter family of uncalibrated motions.

Since $t \odot D_{12} = \varphi(t \odot G_{12})$, this correspond to placing the virtual camera at pose (position and orientation) $t \odot G_{12}$ with respect to the first camera, in the Euclidean space. Hence, we can make sense of this operation as posing the virtual camera at scalar multiples of G_{12}, even if G_{12} is unknown.

The 1-d manifold (a curve) in $\mathrm{SE}(3,\mathbb{R})$ described by $t\odot G_{12}$ contains the pose of the first camera ($t = 0$), and the pose of the second camera ($t = 1$). It interpolates between the two for $t \in [0, 1]$, and extrapolates for $t > 1$ or $t < 0$.

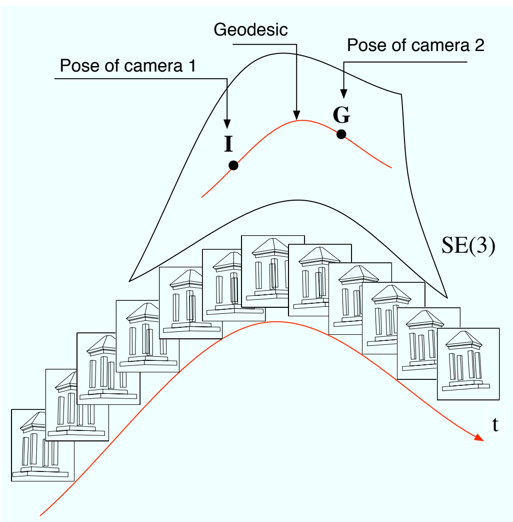

Fig. 1. As t varies in $\mathbb{R}$, the pose of the virtual camera describes a curve in $\mathrm{SE}(3,\mathbb{R})$. This allows to synthesize a sequence of images as taken by a smoothly moving camera.

The complete algorithm for view synthesis along a 1-d manifold with two reference views is summarized below.

TWOVIEWSYNTHESIS

1) Given a set of corresponding pairs $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k)\quad k = 1,\dots,m$;
2) Recover the epipole $\mathbf{e}_{21}$ and the infinity homography $H_{\infty 12}$;
3) Compute the relative affine structure γ_1^k with (11);
4) Let $D_{12} = \begin{bmatrix}H_{\infty 12} & \mathbf{e}_{21}\\ \mathbf{0} & 1\end{bmatrix}$;
5) For any given value of the parameter t:
   a) Compute the uncalibrated motion of the virtual camera $D_{13}(t) = t \odot D_{12}$;
   b) Transfer points to the 3rd view with (29).

B. Motion along a 2-d manifold

Assuming that three reference views are available, I_1, I_2, and I_3, we want to synthesize the fourth view, I_4. To this end we suppose that uncalibrated motions D_{12} and D_{13} are available.

Along the same line as in the previous section, we extend the definition of linear combination of rigid motions to the group of uncalibrated motions via the isomorphism φ:

$$D_{14}(u, v) = (u\odot D_{12})\oplus(v\odot D_{13}) = e^{u\log D_{12}+v\log D_{13}} \tag{32}$$

Thanks to Proposition 5.2, this has a geometric meaning in the Euclidean stratum, corresponding to posing the virtual camera at $(u \odot G_{12}) \oplus (v \odot G_{13})$. This is a 2-manifold (a surface) of $\mathrm{SE}(3,\mathbb{R})$ that contains the poses of the three reference cameras (for $(u, v){=}(0, 0), (u, v){=}(0, 1)$, and $(u, v){=}(1, 0)$, respectively).

A very special case is when the reference views are rectified [22]. Then $R_{12} = I = H_{\infty 12}$ and the epipole is at infinity. Given that no seed rotation is present, the virtual camera can only be translated i) along the line containing the centers of the cameras (in case of two cameras) or ii) in the plane containing the focal planes of all the cameras in the case of three cameras.

The complete algorithm for view synthesis along a 2-d manifold with three reference views is summarized below.

THREEVIEWSYNTHESIS

1) Given two sets of corresponding pairs in 3 views $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k)$ and $(\mathbf{m}_1^k\,;\,\mathbf{m}_3^k)\quad k = 1,\dots,m$;
2) Recover the epipoles $\mathbf{e}_{21}, \mathbf{e}_{31}$ and the infinity homographies $H_{\infty 12}, H_{\infty 13}$;
3) Compute the relative affine structure γ_1^k with (11);
4) Choose a point $\mathbf{m}_1^0$ and scale $\mathbf{e}_{31}$ to satisfy $\mathbf{m}_3^0 \simeq H_{\infty 13}\mathbf{m}_1^0 + \gamma_1^0\mathbf{e}_{31}$;
5) Let $D_{12} = \begin{bmatrix}H_{\infty 12} & \mathbf{e}_{21}\\ \mathbf{0} & 1\end{bmatrix}$ and $D_{13} = \begin{bmatrix}H_{\infty 13} & \mathbf{e}_{31}\\ \mathbf{0} & 1\end{bmatrix}$;
6) For any given value of the parameters u, v:
   a) Compute the uncalibrated motion of the virtual camera $D_{14}(t) = (u \odot D_{12}) \oplus (v \odot D_{13})$;
   b) Transfer points in the 4th (synthetic) view with

$$\mathbf{m}_4^k \simeq [I|\mathbf{0}]D_{14}\begin{bmatrix}\mathbf{m}_1^k\\ \gamma_1^k\end{bmatrix}\qquad k = 1,\dots,m. \tag{33}$$

The linear combination of uncalibrated motions can be applied to an arbitrary number of motions, depending on the number of reference views available. In general, n independent motions will span a n-dimensional manifold of $\mathrm{SE}(3,\mathbb{R})$, which has dimension six. As the dimension of the parameter space grows, however, our view specification method loses part of its ease of use.

VII. EXAMPLES

Some synthetic examples are reported here to illustrate the kind of trajectories that can be achieved with the proposed method. The reference views have been obtained by projecting a (simplified) 3-d model of the "Tribuna" (Piazza delle Erbe, Verona) with known camera matrices. Epipoles and infinity plane homographies have been extracted from the camera matrices. Then, the two synthesis algorithms have been applied respectively to produce the sequences shown here.

In particular, the sequence depicted in Fig. 2 has been obtained with the TWOVIEWSYNTHESIS algorithm. The motion of the virtual camera is described by a single parameter t varying in $[-1.2, 2.2]$, so as to illustate both interpolation and extrapolation.

Fig. 2. Montage of a virtual sequence obtained starting from two reference views (shown in red or lighter line).

The two sequences reproduced in Figures 3 and 4 have been produced with the THREEVIEWSYNTHESIS) algorithm. In this case, the parameter space (u, v) is two-dimensional, and two specific trajectories (shown in Figure 5) have been chosen in this space to generate the two different sequence.

The same sequences have been generated using the ground-truth 3-d structure and camera matrices, and the results are identical, as expected. Figure 6 shows the trajectories of the virtual camera in the Euclidean space for the three sequences reported here.

Albeit the focus of this paper is on the mathematical framework for the generation of virtual trajectories, we implemented a rough-and-ready view synthesis algorithm, intended for illustrative purposes, that works with real images. Figure 7 shows the real reference images and the some frames of the synthetic sequence. The full movie together with other examples is available on the Internet.[3]

A view synthesis algorithm needs the following pieces of data to render new images:

[3]http://profs.sci.univr.it/~fusiello/demo/synth/

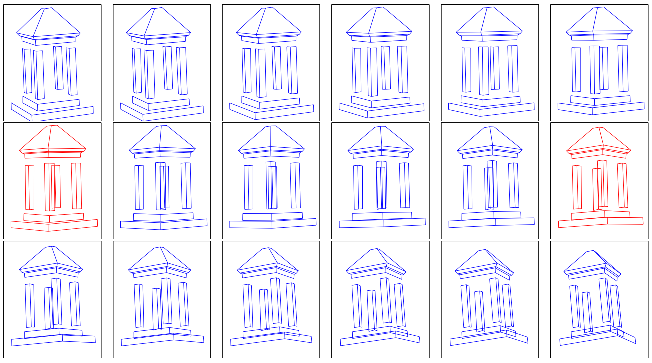

Fig. 3. Montage of a virtual sequence obtained starting from three reference views. The trajectory of the virtual camera is a line in the 2-d parameters space, which contains two reference views, shown in red (or lighter line).

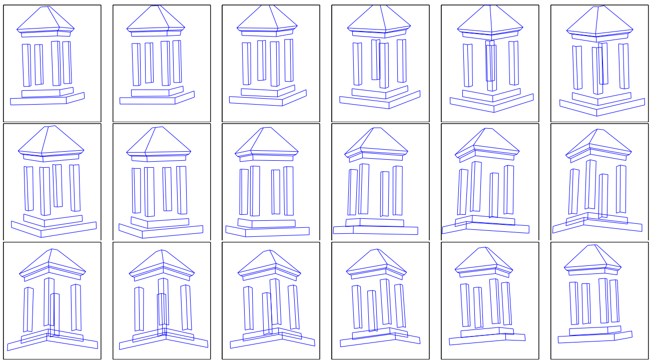

Fig. 4. Montage of a virtual sequence obtained starting from three reference views. The trajectory of the virtual camera is a circle in the 2-d parameters space, hence the last view coincide with the first.

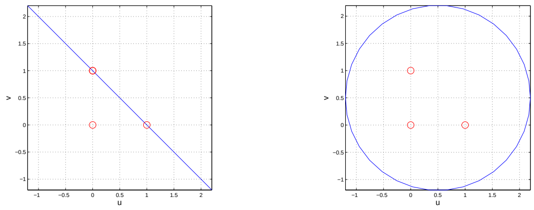

Fig. 5. Trajectories in the (u, v) space $[-1.2, 2.2] \times [-1.2, 2, 2]$. The circles represent the reference views.

1) The plane at infinity between pairs of reference images.
2) The epipoles between pairs of reference images.
3) The relative affine structure of points in one reference image.

The more critical item is the infinity homography, which, however, can be recovered using a variety of heuristics [23]: for example it can be approximated by the dominant homography, or by the homography of the upper part of the image, or by extracting vanishing points. In our case we obtained the infinity homography (or an approximation of it) as part of an uncalibrated rectification algorithm [24]. Given an homography and at least two off-plane points, the epipoles can be obtained with (9). The rectified images are then matched with a stereo matching algorithm [25], and the corresponding points are used to compute the relative affine structure γ. Small artifacts in Figure 7 are due to wrong matches. A more

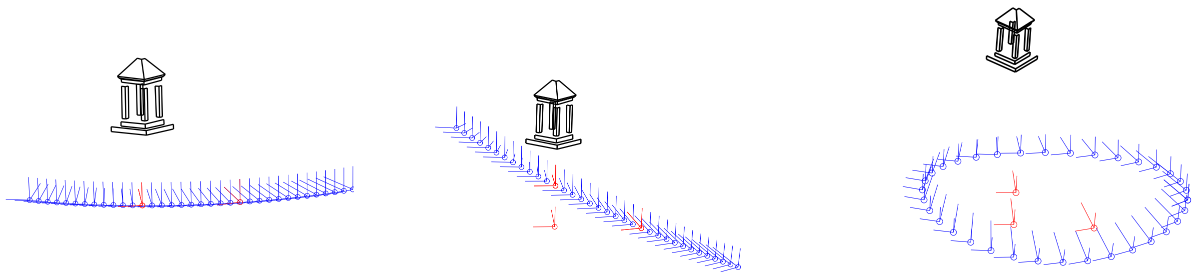

Fig. 6. Position and orientation of the virtual camera in the Euclidean space for the sequences of Fig. 2 (left) Fig. 3 (middle) and Fig. 4 (right).

sophisticated stereo matching with some post-processing of the disparity map (as [26], [27]) would be required to achieve high-quality synthetic images.

Fig. 7. Top row: reference views. Bottom rows: montage of the virtual sequence obtained starting from the reference views.

The algorithm then must specify how the rendering of the new image is performed. In our case, the pixels of the reference image are mapped to the destination image in increasing γ order, which guarantees that points closer to the camera overwrites farther points. Remaining holes in the destination image are filled by interpolation. While this is satisfactory for small holes, larger ones (usually due to occlusions) give rise to blurred areas, as visible in Figure 7. This problem could be tackled by i) reducing the size of holes by using the information coming from all the reference views [28], [27], or by ii) filling the remaining holes with inpainting [29].

VIII. Conclusion

We presented a mathematical framework for posing the virtual camera in the uncalibrated rendering of synthetic views. The method builds upon the linear combination scheme that had been previously developed [15] within the group of rigid motions, $\mathrm{SE}(3,\mathbb{R})$. We extended it to the uncalibrated motion, by observing that an isomorphism exists between $\mathrm{SE}(3,\mathbb{R})$ and the group of uncalibrated motion. This allowed to reflect into the Euclidean stratum all the operations carried out in the uncalibrated stratum, even if the former is not accessible. In particular, we defined parametric trajectories for the virtual camera based on the linear combination of the uncalibrated rigid motions among the reference cameras.

A drawback of this framework is that the description of the uncalibrated motion is less general than it could be, as it requires the homography of the infinity plane. It is not clear whether the theory could be modified to work with a more general description, like a finite plane homography. This issue is left for future investigation.

Acknowledgments: Giandomenico Orlandi contributed to this work with inspiring discussions. Luca Irsara produced the sequence shown in Figure 7.

Appendix

We shall discuss here the existence of the *real* matrix logarithm for elements of $\mathrm{SE}(3,\mathbb{R})$ and $\varphi_K(\mathrm{SE}(3,\mathbb{R}))$ which is necessary for the definitions above to make sense.

Definition 1.1: The linear group $\mathrm{GL}(n,\mathbb{R})$ is the group of the invertible $n \times n$ real matrices.

$$\mathrm{GL}(n,\mathbb{R}) \triangleq \{A \in \mathrm{M}(n,\mathbb{R}) : \det(A) \neq 0\} \tag{34}$$

where $\mathrm{M}(n,\mathbb{R})$ denote the space of all $n \times n$ real matrices.

Given a $A \in \mathrm{GL}(n,\mathbb{R})$, any solution of the matrix equation $e^X = A$, where e^X denotes the exponential of the matrix X, is called *logarithm* of A. In general, A may have an infinite number of real and complex logarithms. However, if A has no eigenvalues on the closed negative real axis then A has a real logarithm [30]. Among all real logarithm there is a unique one whose eigenvalues has imaginary part lying in $]-\pi,\pi[$. This unique logarithm is called the *principal logarithm* of A. It will be denoted by $\log A$. Operationally, the logarithm is defined by the following series

$$\log A \triangleq - \sum_{k=1}^{\infty} \frac{(I-A)^k}{k} \tag{35}$$

which converges in the ball $\|A - I\| < 1$.

The logarithm map projects a neighborhood of I in $\mathrm{GL}(n, \mathbb{R})$ into a neighborhood of 0 in $\mathrm{M}(n, \mathbb{R})$, which can be identified with the tangent space to $\mathrm{GL}(n, \mathbb{R})$ at I.

In our case, both $\mathrm{SE}(3, \mathbb{R})$ and $\varphi_K(\mathrm{SE}(3, \mathbb{R}))$ are subgroups of $\mathrm{GL}(4, \mathbb{R})$. Moreover, they satisfy the spectral condition for existence of a real logarithm. Let $\sigma(A)$ denote the spectrum of A. If $G \triangleq \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3, \mathbb{R})$ then it is easy to see that $\sigma(G) = \{1\} \cup \sigma(R)$, and $\sigma(R) = \{1, e^{\pm i\theta}\}$, being a rotation matrix. As $\varphi_K(G)$ is similar to G, then $\sigma(\varphi_K(G)) = \sigma(G)$.

In [31] a closed-form form for the logarithm in $\mathrm{SE}(3, \mathbb{R})$ is given, which can also be adapted to compute the logarithm in $\varphi_K(\mathrm{SE}(3, \mathbb{R}))$.

REFERENCES

[1] C. Zhang and T. Chen, “A survey on image-based rendering - representation, sampling and compression,” *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, 2004.

[2] O. D. Faugeras and L. Robert, “What can two images tell us about a third one?” in *Proceedings of the European Conference on Computer Vision*, Stockholm, 1994, pp. 485–492.

[3] L. McMillan and G. Bishop, “Head-tracked stereo display using image warping,” in *Stereoscopic Displays and Virtual Reality Systems II*, ser. SPIE Proceedings, no. 2409, San Jose, CA, 1995, pp. 21–30.

[4] L. McMillan and G. Bishop, “Plenoptic modeling: An image-based rendering system,” in *SIGGRAPH 95 Conference Proceedings*, August 1995, pp. 39–46.

[5] S. M. Seitz and C. R. Dyer, “View morphing: Synthesizing 3D metamorphoses using image transforms,” in *SIGGRAPH 96 Conference Proceedings*, August 1996, pp. 21–30.

[6] D. Scharstein, “Stereo vision for view synthesis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, S. Francisco, June 1996, pp. 852–858.

[7] S. Laveau and O. Faugeras, “3-D scene representation as a collection of images,” in *Proceedings of the International Conference on Pattern Recognition*, vol. 1, 1994, pp. 689–691.

[8] S. Avidan and A. Shashua, “Novel view synthesis in tensor space,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1034–1040.

[9] M. Irani and P. Anandan, “Parallax geometry of pairs of points for 3D scene analysis,” in *Proceedings of the European Conference on Computer Vision*, 1996, pp. 17–30.

[10] B. S. Boufama, “The use of homographies for view synthesis,” in *Proceedings of the International Conference on Pattern Recognition*, 2000, pp. 563–566.

[11] A. Shashua and N. Navab, “Relative affine structure: Theory and application to 3-D reconstruction from perspective views,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 483–489.

[12] ——, “Relative affine structure: Canonical model for 3D from 2D geometry and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 873–883, September 1996.

[13] S. Avidan and A. Shashua, “Novel view synthesis by cascading trilinear tensors,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 4, pp. 293–306, Oct-Dec 1998.

[14] K. Connor and I. Reid, “Novel view specification and synthesis,” in *British Machine Vision Conference*, 2002, pp. 243–252.

[15] M. Alexa, “Linear combination of transformations,” in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. ACM Press, 2002, pp. 380–387.

[16] A. Ruf and R. Horaud, “Projective rotations applied to a pan-tilt stereo head,” in *IEEE Conference on Computer Vision and Pattern Recognition*. Fort Collins, Colorado: IEEE Computer Society Press, June 1999, pp. 144–150.

[17] A. Colombari and A. Fusiello and V. Murino, “Uncalibrated interpolation of rigid displacements for view synthesis,” in *Proceedings of the IEEE International Conference on Image processing (ICIP 2005)*, Genova, September 11-14 2005.

[18] S. Ivekovic, A. Fusiello, and E. Trucco, “Fundamentals of multiple view geometry,” in *3D Videocommunication. Algorithms, concepts and real-time systems in human centered communication*, O. Schreer, P. Kauff, and T. Sikora, Eds. John Wiley & Sons, 2005, ch. 6, iSBN: 0-470-02271-X.

[19] P. H. S. Torr and A. Zisserman, “Robust computation and parametrization of multiple view relations,” in *ICCV*, 1998, pp. 727–732.

[20] J. Jost, *Riemannian Geometry and Geometric Analysis*. Berlin: Springer-Verlag,, 2002.

[21] Q.-T. Luong and T. Viéville, “Canonical representations for the geometries of multiple projective views,” *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 193–229, 1996.

[22] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” *Machine Vision and Applications*, vol. 12, no. 1, pp. 16–22, 2000.

[23] T. Viéville, C. Zeller, and L. Robert, “Using collineations to compute motion and structure in an uncalibrated image sequence,” *International Journal of Computer Vision*, vol. 20, no. 3, pp. 213–242, 1996.

[24] L. Irsara and A. Fusiello, “Quasi-euclidean uncalibrated epipolar rectification,” Dipartimento di Informatica - Università di Verona, Research Report RR 43/2006, 2006, submitted to CVPR 2007.

[25] A. Fusiello, U. Castellani, and V. Murino, “Relaxing symmetric multiple windows stereo using markov random fields,” in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Computer Science, M. Figueiredo, J. Zerubia, and A. Jain, Eds. Springer, 2001, no. 2134, pp. 91–104, proceedings of the Third International Workshop, EMMCVPR 2001.

[26] N. Atzpadin, P. Kauff, and O. Schreer, “Stereo analysis by hybrid recursive matching for real-time immersive video conferencing,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 3, pp. 321–334, 2004.

[27] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 600–608, 2004.

[28] J. Shade, S. Gortler, L. He, and R. Szeliski, “Layered depth images,” in *SIGGRAPH 98 Conference Proceedings*, 1998.

[29] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[30] R. Horn and C. Johnson, *Topics in Matrix Analysis*. London: Cambridge University Press, 1991.

[31] J. Cardoso and F. S. Leite, “On computing the logarithm in the special euclidean group of motions in $\mathbb{R}^n$,,” 1999, preprint, Departamento de Matematica, 99-01, Universidade de Coimbra.

Andrea Fusiello received his *Laurea* degree in Computer Science from the Università di Udine, Italy in 1994 and his PhD in Information Engineering from the Università di Trieste, Italy in 1999. He worked with the Computer Vision Group at IRST (Trento, Italy) in 1993-94, and with the Machine Vision Laboratory at the Università di Udine from 1996 to 1998. He had been a Visiting Research Fellow in the Department of Computing and Electrical Engineering of Heriot-Watt University (UK) in 1999. Since year 2000 he is with the Dipartimento di Informatica, Università di Verona, Italy. Currently he holds the position of Associate Professor and teaches Computer Vision and Computer Graphics. He has published more than 30 papers on several topics of Computer Vision, including image rectification, tracking, autocalibration, stereopsis, 2-D and 3-D image registration. His present research is focused on 3-D model acquisition from images and image-based rendering. Further information can be found at http://www.sci.univr.it/~fusiello.