Uncalibrated View Synthesis with Homography Interpolation

Pasqualina Fragneto
(*) STMicroelectronics
Via Olivetti 2, Agrate B.za, Italy
name.surname@st.com

Andrea Fusiello
Università degli Studi di Udine - DIEGM
Via delle Scienze, 208 - 33100 Udine, Italy
andrea.fusiello@uniud.it

Beatrice Rossi* Luca Magri* Matteo Ruffini*

Abstract

This paper presents a novel approach to uncalibrated view synthesis that overcomes the sensitivity to the epipole of existing methods. The approach follows a interpolate-then-derectify *scheme, as opposed to the previous* derectify-then-interpolate *strategy. Both approaches generate a trajectory in an uncalibrated framework that is related to a specific Euclidean counterpart, but our method yields a warping map that is more resilient to errors in the estimate of the epipole, as it is confirmed by synthetic experiments.*

1. Introduction

View synthesis (VS) is the rendering of virtual views of a scene starting from a few reference pictures of the scene itself. Uncalibrated view synthesis (UVS) is a particular case of view synthesis performed without the need of a calibrated camera system. The key advantage of UVS is the possibility to perform VS without any knowledge on the imaging device nor the need of any kind of user interaction including manual camera calibration. In the last years examples of applications of VS techniques have grown considerably. In free view point TV for instance, such techniques are used for the creation of videos where the user can look at the scene from a point of the scene itself, moving freely in the filmed scene. Deducing a set of information about the scene geometry without knowing any information more than some scene photos, VS is also used on many augmented reality systems such as digital archives or augmented panorama. Finally, VS is widely exploited in sports production allowing, one for all, camera views on the pitch. Observe that all these application fields can take a great benefit from fully automated procedures, and thus from UVS techniques.

Related work The VS is based on geometric relationships that are found between the positions of pixels representing the same point in the scene observed from different viewpoints [4]. 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 [13]. Where no knowledge on the imaging device can be assumed, uncalibrated point transfer techniques utilize image-to-image constraints such as the fundamental matrices [11], homographies [3], trifocal tensors [2], plane+parallax [10], or relative affine structure [15, 16] to re-project pixels from a small number of reference images to a given view. All these techniques requires some user interaction in order to specify a position for the virtual camera. In [5, 6] the authors focuses on a new automatic method for positioning virtual views based on the relative affine structure as an implicit geometry descriptor. Given two (or more) reference images, the technique yields a 1-parameter family of virtual camera poses along the geodesic between reference views and allows interpolation as well as extrapolation. In [8], this procedure is extended by allowing additional virtual camera poses from a 1-parameter family to positions in 3-D space along and orthogonally to the line of sight.

Contribution Starting from state-of-the-art UVS algorithms, we are going to describe an alternative UVS procedure to generate trajectories which correspond to a well defined rigid motion (similar to the one in [6]) and which are also naturally robust to the error introduced by epipolar geometry estimation. This last property holds since the epipole value, usually computed from image correspondences and numerically instable, is not involved in the generation of rectified virtual views. In our procedure epipolar geometry intervenes only implicitly during the rectification step. On the whole this results in a greater numerical stability with respect to the procedures in [6] and [8] in cases when epipole computation is affected by errors raising from inaccurate image correspondences.

The outline of the paper is as follows. In Sec. 2 we review the baseline UVS algorithm more in detail. Sec. 3

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.39

describes our alternative solution to performed UVS. Experiments are presented in Sec. 4. Finally some discussion is reported and the concluding remarks are drawn in Sec. 5.

2. Baseline approach

The approach described in [6] can be taken as our baseline. In this section we will briefly summarize it (see Fig. 1). Given two reference images I_1 and I_2, this pipeline consists of the following steps.

Keypoint matching. SIFT features matches are computed on the two reference images as in [12].

Robust epipolar geometry. RANSAC estimation of the fundamental matrix is performed in order to discard bad matches (outliers). The surviving correspondences (inliers) are used as input to the rectification and the epipole $\mathbf{e}_2$ of the second image is computed as the left zero vector of the fundamental matrix.

Rectification. Since camera calibration parameters are unknown, an uncalibrated rectification procedure based on sparse correspondences is employed, see [7] for details. This procedure provides an approximation of the rectifying homographies H_1 and H_2 which rectify respectively I_1 and I_2, and, as a by-product, the infinite plane homography

$$H_{\infty 12} = H_2^{-1} H_1 \tag{1}$$

between the reference images.

Stereo matching. Once the images are rectified, dense correspondences can be obtained using any stereo matching algorithm, for example [18, 17].

De-rectification. Dense correspondences are transferred back to the original reference images by applying the inverse of the rectifying homographies (de-rectification).

Relative affine structure computation. The *relative affine structure* is an implicit geometry descriptor of the scene and one of the key ingredients of this approach. Let $(\mathbf{m}_1^k, \mathbf{m}_2^k)$ with $k = 1, \ldots, m$ be the dense set of correspondences on the two reference images. They are related to each other by the following equation:

$$\mathbf{m}_2^k \simeq H_{\infty 12}\mathbf{m}_1^k + \mathbf{e}_2 {\gamma_1}^k, \tag{2}$$

where ${\gamma_1}^k$ is the relative affine structure of $\mathbf{m}_1^k$ and the sign $\simeq$ means equality up to a scale factor. The relative affine structure ${\gamma_1}^k$ is obtained by solving for γ in Eq. 2:

$${\gamma_1}^k = \frac{(\mathbf{m}_2^k \times \mathbf{e}_2)^T (H_{\infty 12}\mathbf{m}_1^k \times \mathbf{m}_2^k)}{\left\|\mathbf{m}_2^k \times \mathbf{e}_2\right\|^2}. \tag{3}$$

It is possible to demonstrate that the relative affine structure depends only on the source image I_1, i.e. does not depend on the second reference image. More details on relative affine structure can be found in [16].

Trajectory generation. This phase provides a 1-parameter family of uncalibrated rigid transformation matrices each of which describes a points transfer map from the source image to a virtual image.

According to Eq. 2, the points transfer map from the source image I_1 to the second one I_2 is:

$$\mathbf{m}_2^k \simeq \begin{bmatrix}\mathbb{I} & 0\end{bmatrix} D_{12} \begin{bmatrix}\mathbf{m}_1^k \\ {\gamma_1}^k\end{bmatrix}, \tag{4}$$

where D_{12} is the so-called *uncalibrated rigid transformation matrix* between I_1 and I_2 which can be expressed as a combination of the infinite plane homography $H_{\infty 12}$ and the epipole $\mathbf{e}_2$ of the second image (please note the resemblance to a rigid transformation matrix):

$$D_{12} = \begin{bmatrix} H_{\infty 12} & \mathbf{e}_2 \\ 0 & 1 \end{bmatrix}. \tag{5}$$

Since the relative affine structure does not depend on I_2, arbitrary new views I_t, for $t \in [0, 1]$, can be synthesized by substituting D_{12} with the matrix D_{1t} that represents the transformation mapping the source image I_1 to the virtual one:

$$\mathbf{m}_t^k \simeq \begin{bmatrix}\mathbb{I} & 0\end{bmatrix} D_{1t} \begin{bmatrix}\mathbf{m}_1^k \\ {\gamma_1}^k\end{bmatrix}. \tag{6}$$

This equation allows to transfer points from the source image to the synthetic image I_t. We now see how to compute D_{1t}.

As explained in [5], the set of uncalibrated rigid transformation matrices is a group isomorphic to the group of Euclidean rigid transformation $\mathrm{SE}(3, \mathbb{R})$. Let

$$G_{12} := \begin{bmatrix} R_{12} & \boldsymbol{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3, \mathbb{R}), \tag{7}$$

where R is a rotation matrix and $\boldsymbol{t}$ is a vector representing a translation. The isomorphism is given by the conjugacy:

$$D_{12} = \begin{bmatrix} KR_{12}K^{-1} & K\boldsymbol{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix} = \tilde{K} G_{12} \tilde{K}^{-1} \tag{8}$$

with K being the (upper triangular) matrix of the internal parameters of the camera, and $\tilde{K} = \left[\begin{smallmatrix} K & \mathbf{0} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$.

Since $\mathrm{SE}(3, \mathbb{R})$ is a Lie group, it is possible to continuously parameterize virtual camera positions. In particular, Alexa in [1] defines some operators that allow to interpolate, extrapolate and combine rigid transformation in $\mathrm{SE}(3, \mathbb{R})$, which, thanks to the isomorphism, can be mapped to the group of uncalibrated rigid transformation. These operators

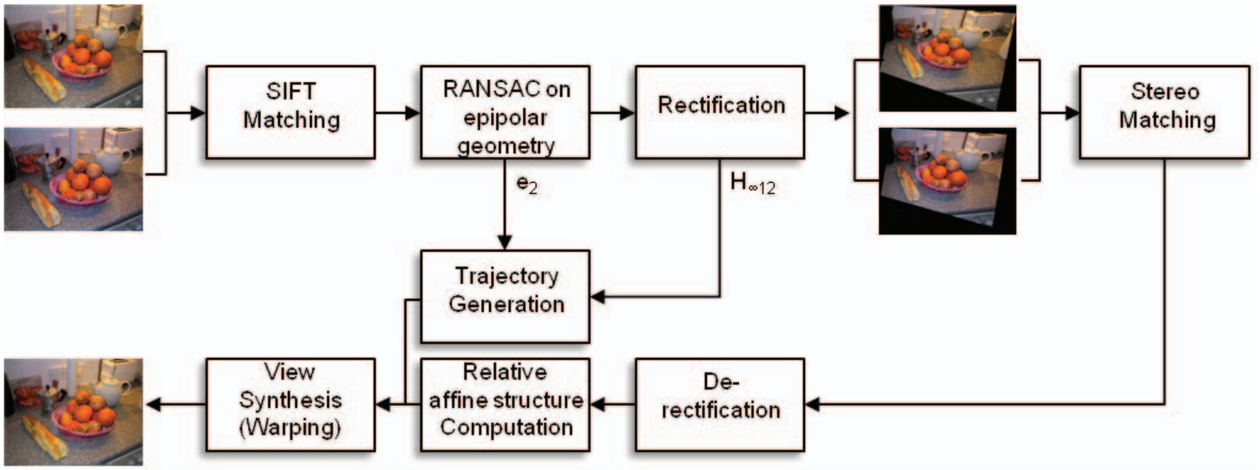


Figure 1: The baseline UVS approach.

are used to compute the uncalibrated rigid transformation D_{1t}. In particular, given the uncalibrated transformation matrix D_{12} between two reference cameras, D_{1t} is defined as its scalar multiple:

$$D_{1t} := D_{12}^t := \exp\left(t \log(D_{12})\right) \quad t \in [0,1]. \qquad (9)$$

Varying the value of t we obtain a 1-parameter family of virtual camera which moves along a geodesic path in $\mathrm{SE}(3,\mathbb{R})$ interpolating the position and orientation of the two reference cameras.

Warping. Warping is the last phase of the pipeline and implements the pixels transfer from the source image to the virtual image I_t. Once computed D_{1t}, the points transfer map is given by Eq. 6. To preserve the coherence of surfaces and their visibility, a pixel splatting with back-to-front rendering can be used as in [6]. Observe that, in order to better handle occlusions, it is possible to exploit information from both reference images as explained in [8].

3. The interpolate-then-derectify approach

Whereas the baseline approach follows a derectify-then-interpolate (DTI) scheme, we propose a novel interpolate-then-derectify (ITD) approach to UVS. More specifically, while the DTI pipeline interpolates the actual (uncalibrated) motion between reference images, in the ITD approach the interpolation is (trivially) performed between rectified images, and the complexity is moved onto the de-rectification step, which brings the interpolated rectified images back in the original (i.e. unrectified) domain, using the inverse rectifying homography. The problem is that this homography is known only for the reference images, while for all the intermediate synthetic images it should be *interpolated*.

The advantage, however, is that the ITD approach turns out to be intrinsically robust to the error committed in epipolar geometry estimation, since the generation of rectified virtual views involves only the theoretical value of the epipole in the rectified configuration, i.e. the direction of the x-axis.

In [14] a similar ITD approach was taken: the rectification and de-rectification were called *pre-warp* and *post-warp* respectively. However the framework was calibrated, so the post-warping was easily defined in a Euclidean reference frame, or manually specified via control points. Here we propose a more elegant solution which exploits the exponential map to directly interpolate the rectifying homographies relative to the reference images. We will see that the trajectory generated in this way corresponds to a well defined rigid motion although it is not a geodesic path in $\mathrm{SE}(3,\mathbb{R})$ as in [6].

The ITD pipeline outline is shown in Fig. 2. Please observe that only the second part of the chain is modified. The main difference with the baseline one is that UVS is now performed by taking as source image the *rectified* reference image, whereas the baseline approach (DTI) started from the original reference image. This produces rectified virtual images which can be de-rectified by using interpolated de-rectifying homographies.

Let I_1^R and I_2^R be the two reference images after being rectified: the steps that characterizes the ITD pipeline are described in the following.

Rectified trajectory generation. This phase provides a 1-parameter family of uncalibrated rigid transformation matrices in the rectified configuration. Each matrix describes a points transfer map from the source image I_1^R to a rectified virtual image I_t^R.

Similarly to Eq. (4), points transfer map between I_1^R and

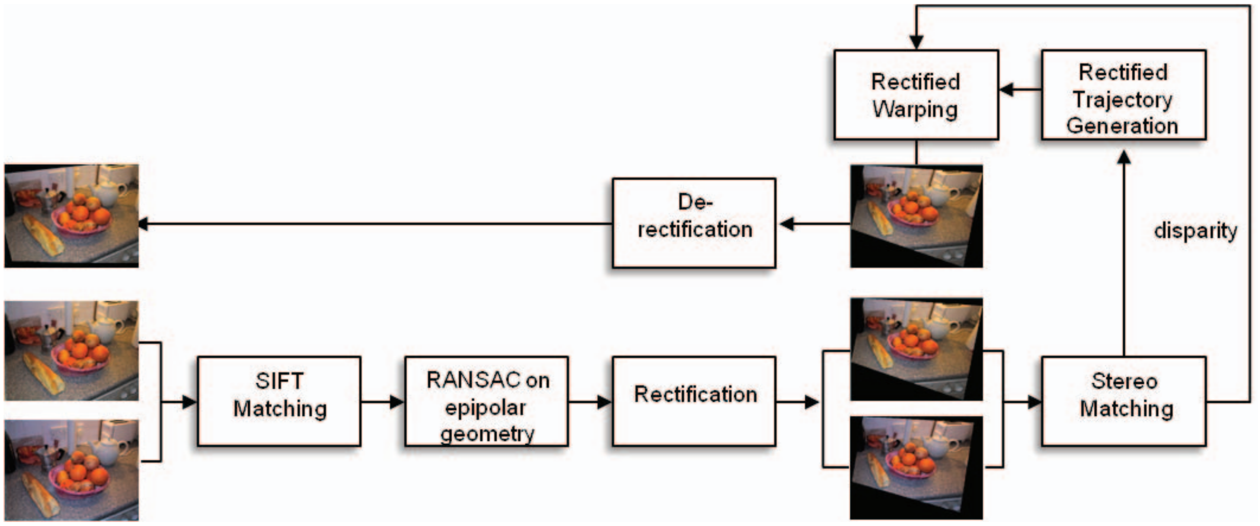


Figure 2: The proposed ITD approach.

I_2^R has the following form:

$$\mathbf{m}_2^{kR} \simeq \begin{bmatrix}\mathbb{I} & 0\end{bmatrix} D_{12}^R \begin{bmatrix}\mathbf{m}_1^{kR}\\ d_1^k\end{bmatrix}, \tag{10}$$

where $(\mathbf{m}_1^{kR}, \mathbf{m}_2^{kR})$ with $k = 1, \dots, m$ is the dense set of correspondences on the two rectified reference images, d_1^k is the disparity value of $\mathbf{m}_1^{kR}$ and

$$D_{12}^R = \begin{bmatrix} H_{\infty 12}^R & \mathbf{e}_2^R \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

is the uncalibrated transformation matrix between I_1^R and I_2^R. Please observe how D_{12}^R has a very simple form. This holds since in the rectified configuration the infinity plane homography is trivial and the epipole has the direction of the x-axis, i.e. $H_{\infty 12}^R = \mathbb{I}$ and $\mathbf{e}_2^R = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$. Furthermore, observe that in this case the relative affine structure and the disparity computed by the stereo matching coincide (up to a scale factor).

The 1-parameter family of virtual rectified cameras which moves along a geodesic path in $\mathrm{SE}(3, \mathbb{R})$ passing through the position and orientation of I_1^R and I_2^R has thus the following very simple form:

$$D_{1t}^R := \exp\left(t \log(D_{12}^R)\right) = \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$t \in [0, 1].$$

Warping. In the ITD pipeline, the warping phase is performed in the rectified configuration. Once computed D_{1t}^R, the points transfer map to use has the very simple form:

$$\mathbf{m}_t^{kR} \simeq \mathbf{m}_1^{kR} + \begin{bmatrix} t d_1^k \\ 0 \\ 0 \end{bmatrix}. \tag{13}$$

Interpolated de-rectification. To obtain a virtual image on a path interpolating the position and orientation of the two original reference cameras, we must de-rectify the virtual rectified image I_t^R. The de-rectifying homography, though, is known only for the reference images, whilst for intermediate virtual images it must be interpolated.

Let H_1 and H_2 the homographies that rectify I_1 and I_2 respectively. As pointed in App. A, the geodesic interpolation between the two is given by:

$$H_t := H_1 (H_1^{-1} H_2)^t = H_1 \exp(t \log(H_1^{-1} H_2)) \tag{14}$$

with $t \in [0, 1]$.

To de-rectify I_t^R we can simply use H_t^{-1}. Please note in Eq. (14) that $H_1^{-1} H_2 = H_{\infty 12}^{-1}$, and that $H_{\infty 12}$ is conjugated (or similar) to a rotation matrix, hence its eigenvalues are $\{1, e^{\pm i\theta}\}$. This guarantees that $H_{\infty 12}$ has a *real* logarithm, as discussed in App. A (with reference to that analysis, $H_{\infty 12}$ falls in case b).)

Let us now consider the following 4×4 matrices:

$$\tilde{H}_1 = \begin{bmatrix} H_1 & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{15}$$

$$\tilde{H}_t = \begin{bmatrix} H_t & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{16}$$

With reference to the following diagram:

$$\begin{tikzcd}
I_1 \arrow[r, "D_{1t}"] \arrow[d, "\tilde{H}_1"'] & I_t \\
I_1^R \arrow[r, "D_{1t}^R"'] & I_t^R \arrow[u, "\tilde{H}_t^{-1}"']
\end{tikzcd} \tag{17}$$

it is easy to see that:

$$\begin{aligned}
D_{1t} = \tilde{H}_t^{-1} D_{1t}^R \tilde{H}_1 &= \begin{bmatrix} H_t^{-1} H_1 & H_t^{-1} \begin{bmatrix} t \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} (H_2^{-1} H_1)^t & (H_2^{-1} H_1)^t H_1^{-1} \begin{bmatrix} t \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} H_{\infty 12}^t & t H_{\infty 12}^t \mathbf{e}_{12} \\ \mathbf{0} & 1 \end{bmatrix}.
\end{aligned} \tag{18}$$

So D_{1t} is conjugated (cf. Eq. 8) with the rigid motion:

$$\begin{aligned}
\begin{bmatrix} R_{12}^t & -t R_{12}^t R_{12}^\mathsf{T} \mathbf{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix} &= \begin{bmatrix} R_{12}^t & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbb{I} & -t R_{12}^\mathsf{T} \mathbf{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} R_{12} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^t \begin{bmatrix} \mathbb{I} & -R_{12}^\mathsf{T} \mathbf{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix}^t
\end{aligned} \tag{19}$$

because $\mathbf{e}_1 = -K R_{12}^\mathsf{T} \mathbf{t}_{12}$ (whereas $\mathbf{e}_2 = K \mathbf{t}_{12}$).

The geodesic path in $\mathrm{SE}(3, \mathbb{R})$ would have been:

$$\begin{bmatrix} R_{12} & \mathbf{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix}^t \tag{20}$$

but since $(AB)^t \neq A^t B^t$ Eq. 19 is not the geodesic. As a matter of fact, at time t our trajectory corresponds to the rigid motion obtained by composing a rotation R_{12}^t and a translation of $t\mathbf{C}$, where, $\mathbf{C}$ is the center of the second reference camera.

4. Results

In this section we show and discuss some results obtained with a MATLAB® implementation of the two pipelines. Experiments show that the ITD is more robust than the DTI pipeline with respect to the error introduced by automatic sparse correspondence extraction and epipolar geometry estimation.

In the first experiment, we synthetically generated 1000 reference images pairs by projecting a cloud of 50 3D-points with randomly chosen camera matrices lying on a sphere enclosing the points. We run the two pipelines for increasing values of t and considered the final point positions as a reference ground truth (different for DTI and ITD). We repeated the experiment by progressively adding Gaussian random noise to the reference images (noise has 0 mean and variance respectively of 0.1, 0.4 and 0.7) and evaluating for each pipeline the absolute value of the difference between the final point positions and the reference positions. This simulates the typical precision on keypoints location in real cases.

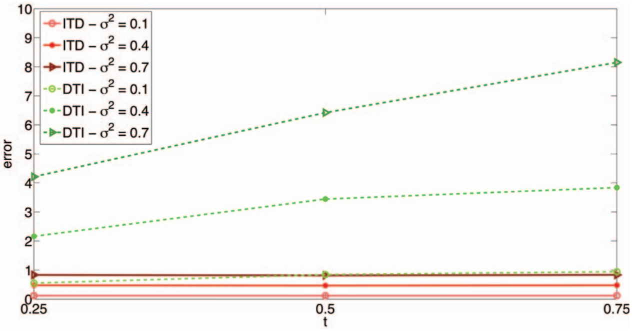


Figure 3: **Average difference**. On the x-axis the values for the t parameter, on the y-axis the average absolute difference (in pixels) between the point positions obtained with ITD/DTI and the reference position. Green dashed lines correspond to DTI, whereas red solid lines correspond to ITD. Different markers correspond to different values of the variance.

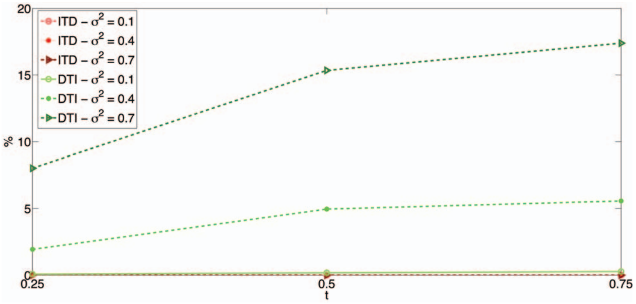
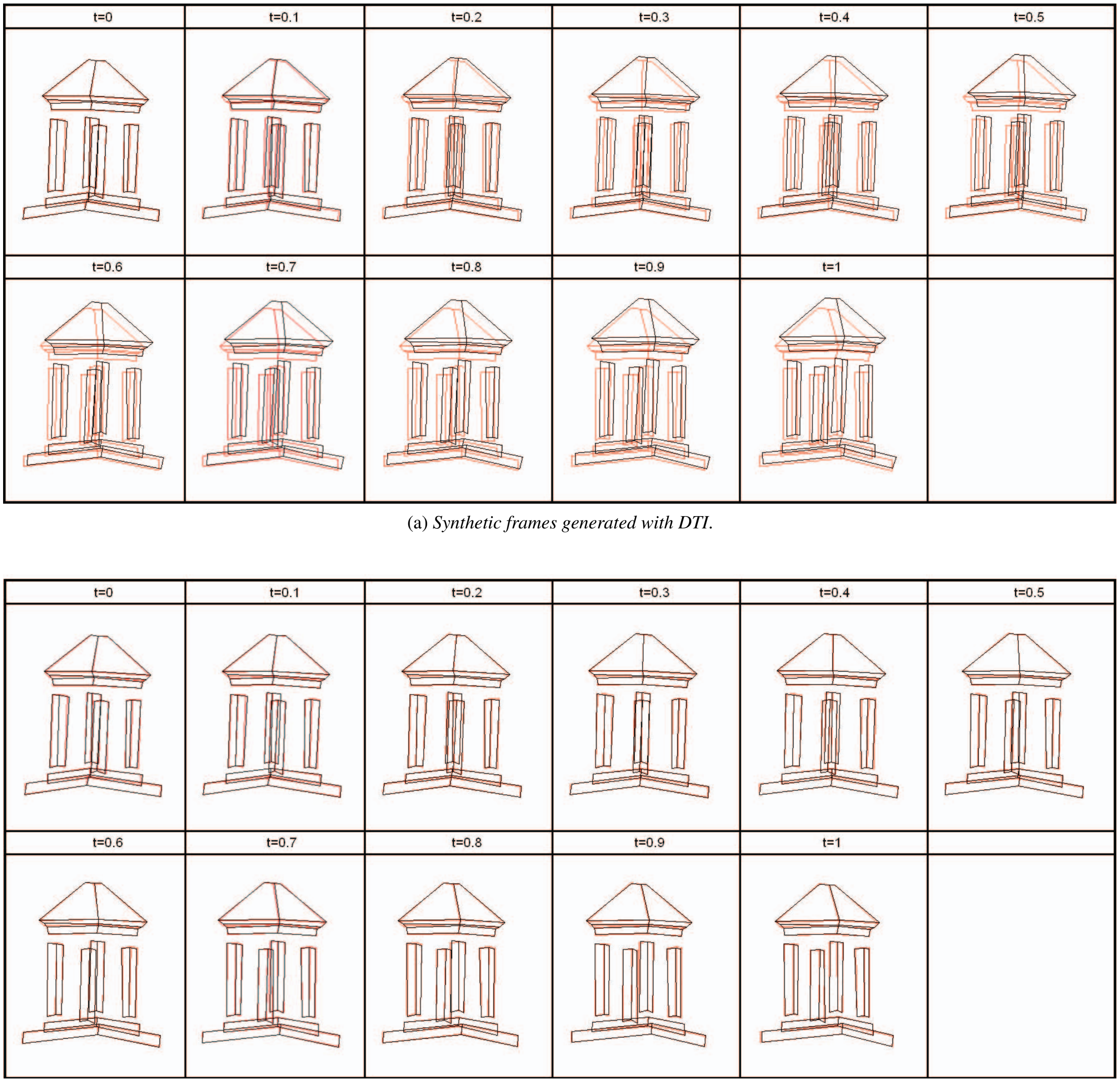


Figure 4: **Percentage of Corrupted Points**. On the x-axis the values for the t parameter, on the y-axis the percentage of points that are mapped by ITD/DTI more that 5 pixels away reference position. Green dashed lines correspond to DTI, whereas red solid lines correspond to ITD. Different markers correspond to different values of the variance. The red lines corresponding to ITD are not visible as they are horizontal and set to 0%.

Results are shown in Fig. 3 and 4. It can be seen that as the noise variance increases, ITD keeps on outperforming the DTI pipeline, resulting in a more robust behavior. Remarkably, no point is mapped more that 5 pixels away from its reference position by ITD, whereas DTI maps about 15% of points outside the 5-pixels radius. This is due to the independence of ITD from the lack of precision in the epipo-



(a) *Synthetic frames generated with DTI.*

(b) *Synthetic frames generated with ITD.*

Figure 5: **Synthetic frames generated with the two pipelines**. This figure is best viewed in colour. In red the ground truth, in black the virtual views computed after perturbing the epipolar geometry. Clearly ITD (bottom) is closer to the ground truth than DTI (top).

lar geometry computation, despite the non-linear refinement with minimization of the Sampson error.

In the second experiment, whose results are shown in Fig. 5, we generated two synthetic sequences using respectively ITD and DTI pipelines with point correspondences corrupted with Gaussian random noise with zero mean and variance $\sigma^2 = 0.7$). In this particular instance, the epipolar geometry computation is particularly instable, for an error of order 10^3 is committed over $\mathbf{e}_2$. As the parameter t increases the DTI output (in black) departs visibly from the ground truth (in red), whereas the ITD output is indistinguishable from the noise-free one.

5. Conclusions

We described a modified view synthesis pipeline that renders a synthetic sequence of virtual images from uncalibrated reference images. We demonstrated that our method generates trajectories which correspond to a well defined rigid motion and which are also naturally robust to the error introduced by epipolar geometry estimation. This alternative approach can thus be particularly useful when reference images present poor precision in keypoints correspondences, or when epipolar geometry estimation is degenerate. In these situations the gain of view synthesis quality is demonstrated with some experiments. The visual quality of the output now depends almost exclusively on the stereo matching procedure. Future work will aim at improving this stage.

Acknowledgments

A.F. is grateful to Giandomenico Orlandi for hints and discussions on homography interpolation.

A. On interpolating homographies

In this appendix we will set and discuss the problem of interpolating homographies of the projective plane. This is similar to the problem of interpolating rigid motion that arise in the field o robotics and computer graphics, but unfortunately it does not have an equivalent closed form solution, to the best of our knowledge.

Before, let us state some known results in linear groups theory.

Definition A.1. *The linear group* $\mathrm{GL}(n,\mathbb{R})$ *is the group of the invertible* $n \times n$ *real matrices.*

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

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

The group of homographies (or projectivity) of $\mathbb{P}^2$ is isomorphic to $\mathrm{GL}(3,\mathbb{R})/\sim$, where $\sim$ represents the equivalence relation "up to a scale factor".

Definition A.2. *For each matrix* $A \in \mathrm{M}(n,\mathbb{R})$ *let us define:*

$$\exp(A) := I + \sum_{k=1}^{\infty} \frac{A^k}{k!}$$

Proposition A.3. *There exist an open neighborhood* $\mathcal{U} \ni$ ***0*** *in* $\mathrm{M}(n,\mathbb{R})$ *which is mapped homeomorphically by the map* $X \to \exp(X)$ *onto an open neighborhood* $\mathcal{V} \ni \mathbb{I}$ *in* $\mathrm{GL}(n,\mathbb{R})$.

The surjectivity of exp restricted to $\mathcal{U}$ implies the existence of the inverse map $\log : \mathcal{V} \to \mathcal{U}$ such that $\exp(\log(X)) = X$. Indeed it turns out that the series

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

is convergent for all matrices in the ball $\|A - I\|_F < 1$, and $\exp(\log(A)) = A$.

The logarithm map projects a neighborhood V of $\mathbb{I}$ in $\mathrm{GL}(n,\mathbb{R})$ into a neighborhood U of **0** in $\mathrm{M}(n,\mathbb{R})$, which can be identified with the tangent space to $\mathrm{GL}(n,\mathbb{R})$ at $\mathbb{I}$. This picture is reminiscent of Riemann's projection of the sphere onto a plane. The exponential map projects this tangent space back into $\mathrm{GL}(n,\mathbb{R})$. A straight path in the tangent space emanating from **0** is mapped onto a geodesic in $\mathrm{GL}(n,\mathbb{R})$ emanating from $\mathbb{I}$. This is formalized by the following:

Proposition A.4. *Given a matrix* $A \in \mathrm{GL}(n,\mathbb{R})$*, the matrix* $A^t \in \mathrm{GL}(n,\mathbb{R})$

$$A^t := \exp(t \log(A)),$$

where $t \in [0,1]$*, describes a geodesic path in* $\mathrm{GL}(n,\mathbb{R})$ *from* $\mathbb{I}$ *to* A

Every matrix in $\mathrm{GL}(n,\mathbb{R})$ has a logarithm, albeit not necessarily real. For obvious reasons, though, we are interested in real matrices, hence the following proposition is particularly useful [9]:

Theorem A.5. *Given* $A \in \mathrm{GL}(n,\mathbb{R})$*, a real logarithm exists iff each of its Jordan blocks associated to a negative eigenvalue occours an even number of times.*

Corollary A.6. *A sufficient condition for* $A \in \mathrm{GL}(n,\mathbb{R})$ *to have a real logarithm is that* A *has no eigenvalues on the closed negative real axis.*

When exists, the real logarithm is not unique. However, among all real logarithms 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 $\log A$.

We can now state the problem as follows:

Problem A.7. *Given two homographies of* $\mathbb{P}^2$ H_1 *and* H_2 *find a "natural" interpolating homography* $H(t)$ *where* $t \in [0,1]$.

As homography matrices are defined up to a scale factor, we can assume $\det(H_1) > 0$ and $\det(H_2) > 0$[1]. The problem can therefore be re-formulated – with no generality loss – as follows:

Problem A.8. *Given a homography of* $\mathbb{P}^2$ *whose matrix H is real and with* $\det(H) > 0$*, find an interpolating path* $H(t) \in \mathrm{GL}(3,\mathbb{R})$*, from* $\mathbb{I}$ *to* H*, where* $t \in [0,1]$.

[1] As these matrices are 3x3 one can change the sign of the determinant by multiplying by -1

The original problem can be reduced to the latter by taking $H = H_1^{-1} H_2$. Given such an interpolating path $H(t)$, the solution to the original problem is $H_1 H(t)$.

Based on Proposition A.4, the natural solution is to take

$$H(t) := H^t = \exp(t \log H). \tag{21}$$

As a matter of fact, the formula above solves our problem as long as $\log H$ is real, hence we shall study the spectral properties of H in order to apply Theorem A.5.

The eigenvalues of any matrix in $\mathrm{GL}(3, \mathbb{R})$ are either all real, or one real and two conjugate complex. In our case, as $\det(H) > 0$, the eigenvalues of H are either a1) all real and positive, a2) two negatives and one positive, or b) one real positive and two conjugate complex.

In cases a1) and b) the real logarithm of H exists thanks to Corollary A.6. Let us now concentrate on case a2). Let

$$H = V \begin{bmatrix} A & 0 \\ 0 & \lambda_3 \end{bmatrix} V^{-1}$$

be the eigendecomposition of H in this case. It turns out that either a2.1) A is diagonal $A = \left[\begin{smallmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{smallmatrix}\right]$ if H can be diagonalized, or a2.2) A is a 2×2 Jordan block $A = \left[\begin{smallmatrix} -\lambda & 1 \\ 0 & -\lambda \end{smallmatrix}\right]$ if H cannot be diagonalized. Unfortunately, according to Theorem A.5, A does not have a real logarithm in these two cases, unless $\lambda_1 = \lambda_2$ in case a2.1.

Hence, a solution to the problem of interpolating *general* homographies is still missing. However, as far as infinite plane homographies are concerned, the favorable case b applies.

References

[1] M. Alexa. Linear combination of transformations. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 380–387. ACM Press, 2002. 2

[2] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):293–306, Oct-Dec 1998. 1

[3] B. S. Boufama. The use of homographies for view synthesis. In *Proceedings of the International Conference on Pattern Recognition*, pages 563–566, 2000. 1

[4] 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*, pages 485–492, Stockholm, 1994. 1

[5] A. Fusiello. Specifying virtual cameras in uncalibrated view synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(5):604–611, May 2007. 1, 2

[6] A. Fusiello and L. Irsara. An uncalibrated view-synthesis pipeline. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 609–614, 10-14 September 2007. 1, 2, 3

[7] A. Fusiello and L. Irsara. Quasi-euclidean epipolar rectification of uncalibrated images. *Machine Vision and Applications*, 22(4):663 – 670, 2011. 2

[8] F. Gigengack and X. Jiang. Improved uncalibrated view synthesis by extended positioning of virtual cameras and image quality optimization. In *Asian Conference on Computer Vision*, pages 438–447, 2009. 1, 3

[9] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, London, 1991. 7

[10] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Proceedings of the European Conference on Computer Vision*, pages 17–30, 1996. 1

[11] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and foundamental matrices. Technical Report 2205, INRIA, Institut National de Recherche en Informatique et an Automatique, February 1994. 1

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[13] L. McMillan and G. Bishop. Head-tracked stereo display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II*, number 2409 in SPIE Proceedings, pages 21–30, San Jose, CA, 1995. 1

[14] S. Seitz and C. Dyer. View morphing. In *SIGGRAPH 96*, pages 21–30, 1996. 3

[15] 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*, pages 483–489, 1994. 1

[16] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, September 1996. 1, 2

[17] B. M. Smith, L. Zhang, and H. Jin. Stereo matching with nonparametric smoothness priors in feature space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 485–492, 2009. 2

[18] Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2