The 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2008)
M. Ashley, S. Hermon, A. Proenca, and K. Rodriguez-Echavarria (Editors)

Reality Filtering: A Visual Time Machine in Augmented Reality

Michael Zoellner[1], Alain Pagani[2], Yulian Pastarmov[2], Harald Wuest[1] and Didier Stricker[2]

[1]Fraunhofer IGD, Germany
[2]DFKI, Germany

Abstract

We present Reality Filtering, an application that makes it possible to visualize original content like drawings or paintings of buildings and frescos seamlessly superimposed on reality by using filtered augmented reality. This enables simple and inexpensive applications in the cultural heritage and architecture area. The main idea is that the video stream showing the reality is filtered on the fly to acquire the same presentation style as the virtual objects. This allows for a better integration of original historic content and creates the impression of a virtual time journey. The registration of the virtual objects in the video images is provided by a robust 6DOF tracking framework based on two technologies that work in tandem: an initialization step based on Randomized Trees and a frame-to-frame tracking phase based on KLT. For the initialization, we present the novel concept of temporally distributed computational load *(TDCL), which is able to automatically detect and register multiple objects while maintaining a constant video frame rate of 20 frames / sec. For mid- to long-range augmentation a pure 2-dimensional tracking with 3DOF is applicable and leads to significant performance gain. The entire application runs in real time on Ultra Mobile PCs.*

Categories and Subject Descriptors (according to ACM CCS): J.5 [Computer Applications]: Arts and humanities

1. Introduction

In the last years we observed three major problems of Augmented Reality in cultural heritage projects: first, the virtual reconstructions of missing masterpieces or buildings often suffer from poor visual quality, doubtful scientific accuracy and are at the same time costly to produce. Second, robust markerless tracking solutions are often missing, which leads to the use of non aesthetic markers. Third, no high-performance and portable computing platforms are commonly available to run demanding software solutions. Based on our experience in cultural heritage projects, we show in this paper a solution to the two first problems, and suggest a platform to address the third one.

Most interactive cultural heritage projects suffer from a lack of 3D content for interactive visualizations. The reason is not only the expense of the creation of high quality 3D models or the usage of laser scanning technologies. Scientific accuracy is another serious matter. Untrue photorealistic interpretations of historic objects can lead to a fact in the viewers mind. Furthermore exaggerated and wrong lighted 3D models are ruining every Augmented Reality scene. Thus we have developed a way for the intelligent use of existing historical content like drawings and paintings to enable more cultural heritage sites to use Augmented Reality to enrich the experience of their sites. Instead of rendering the virtual overlays in the way the reality looks like we are rendering the video stream of the reality to fit the source material: Black and white drawings, oil paintings and so on.

Another difficulty of using Augmented Reality in cultural heritage projects is that the tracking technology has to be robust enough to handle large environments with difficult lighting conditions, and must be at the same time discreet enough to not disturb the user. At best, the tracking technology should be invisible to the user, excluding the usage of standard markers. In this application, we opted for a mixture of two recent markerless tracking technologies, which proves to be robust and fast enough for our aimed platform. The usage of 2D textures instead of 3D models also needs less computing power and thus enables the visualization on

© The Eurographics Association 2008.

Figure 1: *Virtual ancient temple on filtered real background*

ultra mobile PCs like the Sony UX and mobile phones in the near future.

The remainder of the paper is organized as follows. Section 2 is a global overview of our system. Sections 3 and 4 present the markerless tracking technologies we are using respectively for indoor and outdoor scenarios. We present our new concept of Reality Filters in section 5, and section 6 shows the obtained results in a test case. We finally conclude in section 7 and suggest future work.

2. System overview

In this section, we give a brief overview of our system. Typical users of our system are visitors of archeological or historical sites (indoor or outdoor), museum visitors or visitors of architectural sites. The equipment consists of a hand-held Ultra Mobile PC (UMPC), on which our specific software is installed. The UMPC serves as a *reality-window*: a small, integrated video camera placed on the back of the device records live pictures that are shown in full screen on the 5-inches display. Thus, the user sees the reality through the device (see Figure 1). For a given application site (*e. g.* a museum, an open-air archeological or site or an architectural ensemble), a number of objects of interest termed *spots* are selected by the application designer. These spots can be for example sculptures or paintings in a museum, building facades, archeological ruins, or even gardens like in our applicative example (see section 6). When looking through the UMPC, these spots will be *augmented* with additional information in form of original historical content like drawings or paintings of buildings and frescos in a seamless integration based on our new concept of Reality Filtering.

Given a few reference images of the spots, our system learns in a preliminary phase to recognize the spots. In the online phase (on site), the system is first in a *initialization* rountine, which analyses the images from the camera and finds out if the current view is from a given spot or not. When a spot has been recognized, the system enters a *tracking* state, where the position of the object of interest is precisely recovered and tracked, allowing for a seamless geometric integration of the additional information. These two phases (initialization + tracking) take advantage of recent advance in computer vision in a novel multiple object detection and tracking framework, and are described in more detail in section 3.

We identified two main scenarios naturally arising in cultural heritage sites: the first one involves 3D objects located relatively near to the user (1 to 10 meters). This is the case generally for indoor scenarios with small to medium sized objects (paintings, sculptures, walls of a room). The second one involves distant objects like buildings or landscapes, and appears generally outdoor. In this case, the size of the object is small in comparison with the distance from the user, and the objects can be approximated as a panoramic view. In our system, we developed specific pose estimation algorithms for each of these scenarios: in the indoor case, a full 6 degrees of freedom (DoF) camera position is computed, while in the outdoor case, a simpler, 3-DoF motion model is adopted.

The seamless photometric and stylistic integration of the original drawings is achieved using our new concept of Reality Filtering, where the style of the real images are adapted to the original content. The style of presentation for the reality can be chosen among a list of filters, or new styles can be easily added to the system. The choice of the style is made at application integration time and is fixed for a given spot. Some of our filters are described in section 5.

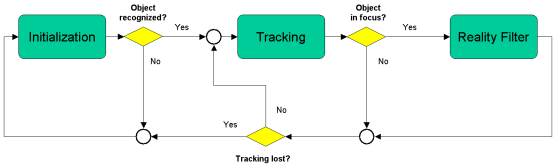

Figure 2: *Block diagram of our multiple scene augmentation algorithm*

Figure 2 shows a block diagram of the logics of our system. The system first enters an initialization state, where an object / scene detector searches for known scenes as long as no such images are discovered. When an object has been discovered, the position and orientation of the object is computed. Then, the systems enters the tracking phase, where the position of the object is updated in a frame-to-frame basis. At the same time, the system looks if the object of interest is in the focus (a predefined zone in the field of view, roughly in the middle of the image). If so, the real images are processed by the Reality Filter and the original content is shown on the screen.

© The Eurographics Association 2008.

3. Tracking system for indoor scenarios

In this section, we describe the tracking system we use when the motion of the user (or of the camera) is general in 6 degrees of freedom. This is mostly the case in indoor scenarios, where the dimensions of the object have the same order of magnitude as the distance between the user and the objects. As mentioned in the system overview, our system can deal with multiple objects of interest, or spots, in the same application. These objects can be paintings, scultpures, walls, etc. For each spot, one or more reference images are required. These images are collected under varying viewing angles or lightning conditions. Let us call $\{I_n^{ref}\}, n \in [0...N]$ the set of N reference images. The reference images are calibrated manually by using known points in the scene or markers. As a result, for each reference image I_n^{ref}, a corresponding camera pose $[R_n, t_n]$ is known.

3.1. Real time multiple objects detection and pose estimation

In a first step, the system tries to identify objects in the camera images, and compute the object's position when one has been found. This step is achieved using a novel method for real time multiple object detection, based on the concept of randomized trees, and on an efficient computational distribution.

3.1.1. Preprocessing

We learn a representation of each image by using the Randomized Trees framework introduced by Lepetit et al. [LLF05]. Following preprocessing steps are done for each reference image: first, a set of P robust salient points are extracted from the image. This is done by rendering thousands of randomly warped versions of the image (by warping, we mean a realistic projective transformation of the image plane), and running a corner detector on every version. The found corners are back-projected in the original view, resulting in clusters of found corners growing with the number of tries. After a given number of tries (typically several thousands), the P clusters accumulating the most votes are kept as robust corners. For each of these robust corners, the 3D coordinates of the point is computed by making use of the camera pose $[R_n, t_n]$ and of the known 3D model of the object of interest. For the calibration of the reference images, the geometry of the scene can be approximated, as only a rough camera pose has to be found in the initialization phase. For example, a room can be approximated by 4 planar walls, and an outdoor scene can ba approximated by a panoramic plane. We end up having P points with known 3D coordinates for each image. In our experiments, we used $P = 200$, but other values are possible. These points are called *classes* in the next learning process. Second, a number of randomized trees [LLF05] are grown for the P classes by using again the concept of warped versions of the image. A randomized tree is a classification tree that classifies a 2D point by making simple decisions about pixel intensities on a small image patch around the point. It stores the probability of each class in its leaves. While one tree alone is not sufficient for a correct classification, it has been shown [AG97] that using multiple trees, or *forests*, improves the classification rate drastically. For each reference image I_n^{ref}, we construct a forest of M randomized trees $\{T_m^n\}, m \in [0...M]$.

3.1.2. Initialization with temporally distributed computational load

Our idea is to detect an object or a scene by trying to compute the pose of the object. Let's suppose that we want to know if the object from the reference image I_n^{ref} is present in the current camera image. To this aim, we extract the salient points of the current image using a corner detector, and classify them by using the n-th forest $\{T_m^n\}, m \in [0...M]$. All points with a probability above a given threshold are kept as possible 2D-3D correspondence. The pose of the camera is then computed from these correspondences by using standard techniques associated to a robust outlier detection via RANSAC [FB81]. If a pose is found by RANSAC, it means that the object of reference image I_n^{ref} was present in the camera image (detection), and we have found its pose in the same process (pose estimation).

Ideally, for one input camera image, this test should be repeated for the N reference images. Unfortunately, this makes the computational costs grow linearly with the number of reference images, making it impossible to maintain a high video frame rate. Instead, we propose to distribute the computational load over several frames by testing only one reference image at each frame. If a pose is found with this forest, it means that the correct forest was selected for the current view, and the system switches to the frame-to-frame tracking phase. If no pose is found, then either the forest was the wrong one for the current view, or the object was not in the field of view of the camera. In either case, the system does not initialize, and tries the next frame with the next available forest. If N reference images are provided, all the forests can be tried in N frames, which is usually less than a second for about 20 images. This means that the system is able to initialize in about one second while showing a video frame rate of 20 frames per second. More generally, a subset of k forests ($k < N$) can be tried for each frame instead of only one. The number k of forests in the subset depends on the aimed frame rate and on the time required by one pose estimation. We call this technique *temporally distributed computational load* (TDCL).

3.2. Frame-to-frame tracking

Once the tracking system has been initialized (Section 3), we start to follow the position of the object by using feature tracking techniques. Feature tracking describes the process of following 2D points in subsequent images that cor-

© The Eurographics Association 2008.

respond to the same physical point in the environment explored. By 2D points, we mean salient points that can be found by classical corner detectors. These points are described by small image patches around them, and the set point and patch is called *a feature*. In order to follow 2D points, correspondences between points from two consecutive frames of the image stream must be built and maintained as long as possible. The longer features are being successfully tracked the more stable is the camera pose estimation. This means that the point detector is required to be highly repetitive. Of course this condition is vital for the tracking algorithm used, too. From the large palette of possible approaches to that problem the Kanade-Lukas-Tomasi tracker (KLT tracker) described in [TK91] and [ST94] has been chosen. The idea proposed by Tomasi at al. benefits from relatively low computation times and strong spatial correctness. In short, tracking is based on building an image pyramid from the live image, with levels of decreasing scale and increasing Gaussian filtering for smoothing out smaller artifacts in the image. A given feature is searched for in a given neighbourhood around the position where it was detected in earlier frames. This process is done on all levels of the pyramid starting at the coarsest level down to the finest one. This enables us following features over longer distances with high precision. In case not enough features are present for pose computation, new ones are selected from the feature map and used in the tracking further on.

The procedure described up to now provides effective way of estimating the optical flow on a frame-to-frame basis. However the lack of context requires some more temporal information to be collected and used to ensure that tracking is still correct over longer time intervals. Therefore a set of features detected in the beginning of the tracking process are selected and marked as persistent and these are not being replaced or removed from the current set of features throughout time. This set of anchor features help stabilising the tracking process. In case of relatively stable lighting conditions and static scenery as the one we had in our case a set of anchor features can be extracted from reference images taken before the actual tracking. This gives the opportunity to prepare the tracking process for given positions in advance and thus ensure correct augmentation without the need of immediate calibration each time the tracking starts.

After the KLT features have been successfully tracked in 2D, we end up again with 2D-3D correspondences. It is then possible to compute the camera position with usual minimisation techniques as in the initialization step.

4. Tracking system for outdoor scenarios

In most outdoor scenarios, the considered objects are located far from the user (landscape, buildings, ruins). In this case, the objects can be considered as a panoramic view around the user. Our system exploits this property by using a simplified camera motion model for such scenarios. In scenes with static viewpoint (the user has a fixed position on the ground, and is only allowed to rotate the camera around), in order to correctly estimate observer's viewing angle, a simplified position estimation can be used. Computing the transitions of well distinguishable points in live image flow provides enough information to compute the camera pose with 3-DoF (degrees of freedom), representing its relative rotation according to the camera's optical centre. Once the camera orientation is successfully computed virtual objects can be correctly overlaid on the live image at real-time speed. In this section, we describe our simplified model for controlled motions.

4.1. Camera model

The camera pose is estimated as rotation R around the centre of the camera coordinate system. This can be assumed to coincide with the center of the world coordinate system the camera resides in. As we are interested in a pure rotation, camera motion can be represented by a homography at infinity H_{inf} as described in [HZ01]. However in our implementation the homography matrix is never explicitly computed. Instead a rotation is estimated based on a non-linear minimization of the re-projection error of the tracked features. The estimate of the previous frame serves as starting point.

The general model of a 6DOF camera projection is described by the following equation :

$$p^T = K[RP+t] \tag{1}$$

Where P is the position of a given point in world coordinates, K is the matrix of the camera's intrinsic parameters, R is a 3x3 rotation matrix, t a translation vector and p^T the homogeneous projection of P in the image. The intrinsic camera parameters matrix has the following form :

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Where f_x and f_y express the focal length and the pixel sides ratio, s is the skew factor of the pixel matrix and c_x and c_y are the optical centre in pixel coordinates.

In our model, we are assuming that $t = (0,0,0)^T$, so that the following formula applies:

$$p^T = KRP \tag{3}$$

4.2. Rotation estimation

From equation (3) we can derive a formula that computes the 2D translation of a given point in the image when the camera switches from a known position R_{init} to a new, estimated one R_{est}. Let's assume we have a point P with its coordinates in an image p_{init} taken from a camera with rotation matrix R_{init}.

© The Eurographics Association 2008.

Its coordinates in world coordinate system are given by the equation up to an unknown scale factor :

$$P = R_{init}^{-1} K^{-1} p_{init} \tag{4}$$

Now let's assume R_{est} is the rotation matrix of the current frame, hence if both R_{init} and R_{est} are correct the same point P must be present in the image at coordinates p_{est} given by the following equation (subsituting (4) in (3)):

$$p_{est} = K R_{est} R_{init}^{-1} K^{-1} p_{init} \tag{5}$$

Using this formula and for an estimated rotation R_{est} the re-projection error $\|p_{est} - p_{act}\|$ can be estimated as the distance between the actual feature coordinates and the projected coordinates. We make the assumption that the motion between consecutive frames is small enough, so that we can use a non-linear least squares minimization of the the re-projection errors of all successfully tracked features, that converges to the correct rotation.

To further improve the correctness of the results a two-stage estimation process is employed. First a rough pose is computed from all points marked as correctly tracked by the KLT tracker. However this set often contains outliers and could be unsuitable for precise augmentation. Therefore once initial pose estimate is present all features that have re-projection errors of more than a few pixels are excluded from the computation and second more precise estimate is established.

One could compute the re-projection error between the current and the last frame thus not storing any further information longer than 2 frames, but then drift and jitter are inevitable, because of the jitter of the localisation of features. Therefore in our implementation error is estimated between the frame where a feature is first found and the present frame. Hence most features are found at different frames jitter is being smoothed out. Moreover the presence of many features helps for a fast minimization convergence and keeps the whole processing time low enough to achieve constant frame rate of 30fps on a P4 2.4Ghz and satisfying 15fps on a Sony Vaio Micro PC with Intel Core Solo 1.3Ghz - the intended target of application.

5. Reality Filters

As a first feature of our system, we can apply different filters on the real video stream. These filters can be chosen by the application conceptor, or can be switched on the fly during the use of the system. In this section, we present two of the filters we are using.

5.1. Drawing Filter (Sobel)

The following approach is a solution to adapt the real view of a scene to blend well with black and white drawings in Augmented Reality applications by applying an inverted sobel filter on the video stream. In contrast to the methods described in [Fis05] we are concentrating on 2D textures instead of 3D models. This avoids the expensive and longsome process of producing high quality 3D models and the realistic integration into the video stream. The result is a reduced aesthetic defined by the original material and an affordable application. This applies not only to cultural heritage sites with only historic material available for visualization but also for example for architectural visualizations of 2D plans.

In cultural heritage this visual effect results in a visual time machine via Augmented Reality because the whole scene is rendered like a real time drawing. A drawing that is controlled by the user's movement and is displaying real buildings and people like a sketch (see Figure 1).

Due to the reduced black and white style of the environment accentuations are much stronger than on a real colorful background. That enables a more efficient visualization of points of interest and drives a viewer's attention on them.

5.2. Tranquility Filter

Science fiction writer Charles Stross writes in his book *Accelerando* [Str05] about a reality filter technique called *Superplonk*. It enables persons to remix their perception and filter out certain persons, objects and sounds. What he describes is an advanced version of our Tranquility Filter we realized with today's technology: video see through and computer vision.

Our actual prototype filters people out of a video stream. Crowded places with a lot of moving people are rendered empty (Figure 3). This does not bring back the original population in the first place. But it removes everyday car traffic and tourists from a historic scene. Only ghost-like silhouettes are remaining. The environment like moving clouds, weather and lighting condition are remaining. Adding the audio layer of the original situation via headphone revives the original tune of a place.

We are reconstructing a place's original image with simple motion detection and an array of reference images. The current image is compared to older images and moving areas are removed. The final empty image is calculated from the average of the resulting images. This leads to the current restriction to stationary mounted cameras like telescopes and surveillance cameras.

6. Application

6.1. Main application

This approach is already used for the field tests of the EU funded project iTACITUS. Within this project, one of the field test areas of the Augmented Reality applications is Reggia Venaria Reale, an UNESCO World Heritage site in Italy

© The Eurographics Association 2008.

Figure 3: *Unfiltered and filtered webcam image of Rotonda Square in front of the Pantheon in Rome*

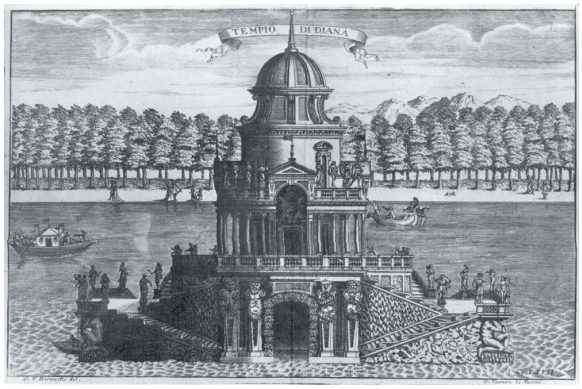

Figure 4: *Drawing of* Tempio di Diana

close to Turin. The former residences of the Royal House is comparable to the french Versailles. The site has been restored over the last years and was opened to the public in fall 2007. While there are only a few 3D reconstructions of some buildings there is a vast archive of historic drawings and paintings. There are frontal drawings of fasades of complete streets and the main palace's buildings.

This vast archive is the basis of our visual time machine. With the Reality Filters we are able to create this effect through the display of a Sony UX Ultra Mobile PC. In order to match the source material we are rendering the video stream like a black and white drawing. Thus overlays of frontal drawings of buildings onto the real field of vision do not look like foreign objects. The whole scene looks like a real time ancient drawing. Even visitors, guides and guards are rendered in black and white.

The application consists of three spots at the site. The first Tempio di Diana (Figure 4) was located at the end of a long path along a small artificial creek. It was surrounded by water and only accessible by boat. Only ruins of the fundament are left.

At the horizon we are displaying the drawing of Tempio di Diana (Figure 4) on top of these ruins. Standing at a viewing platform visitors can look around and watch through the display of a Sony UX via video see through (Figure 5). As soon as they are looking at the direction of the temple the video

Figure 5: *Drawing of Tempio di Diana seamlessly integrated into the environment*

Figure 6: *Palazzo di Diana's architecture over the centuries.*

turns black and white and the drawing of the temple is superimposed. The *Fountain d'Hercule* in front of the viewing platform will be superimposed the same way.

Reggia Venaria Reale's Palazzo di Diana's architecture was modified several times over the years. Each state of the buildings was documented on drawings. We are superimposing these drawings of the modifications of the architects Castellamonte (1674) and Garove (1700-1713) on the facade of the main building. Visitors standing in the large courtyard are seeing the fountain and the current restoration of the palazzo rendered like a drawing. While listening to the audio guide's story about the palazzo the buildings appearance is switching through the centuries while seamlessly integrated into the environment (Figure 6).

6.2. Creating a tracker

In order to let application developers and designers create the tracking for a scene we created a simple automation in our software. The user only selects a reference image for the tree generation. In our example of Temple Diana we used tourist's images of Reggia Venaria Reale found on the in-

© The Eurographics Association 2008.

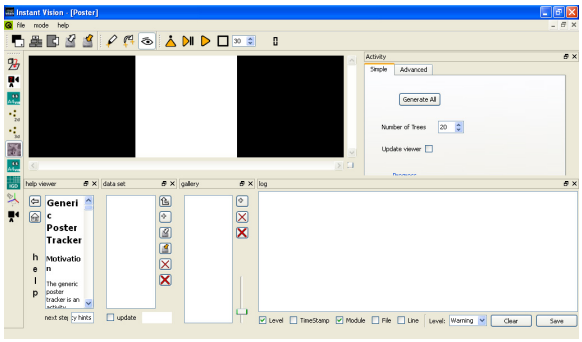

Figure 7: *Automation enables everyone to create markerless tracking*

ternet for tracking. The automation creates the trees and an example X3D scene with the tracking (Figure 7).

There are two advantages of this system compared to prior Augmented Reality content creation and tracking. First we are able to use a printed poster for testing the tracking and the placement of the content. Even whole rooms can be tested with cardboard miniatures. Thus traveling time and costs for on-site testing are reduced and the stability of the application increased. Once arrived at the site the application runs out of the box. Our application of Tempio di Diana trained with sunny tourist photos even worked on a rainy april day with desaturated colors. Only some modifications where needed for a first presentation.

6.3. Interactive time journey

We have integrated in the application a mode where the surroundings are shown without filters as long as the object (here: Tempio di Diana) is not in the camera's field of view. In this mode, only the shape of the object is shown, together with a view finder in the middle of the image (see Figure 8, top). When the user places the view finder inside the shape, the object is shown, and the surrounding environment is filtered (Figure 8, bottom). This way, the user can interactively play with the scene, and make the object appear or disappear, switching back and forth in time.

Figure 8: *Interactive time journey. (Top) Present - only the shape of the object is seen. (Bottom) Past - the ancient temple is shown, and the reality adapts to the drawing*

7. Conclusion

In this paper, we have presented our Augmented Reality application called Reality Filtering. Its main features are the possibility to show original content superimposed on lookalike reality, its robust markerless tracking and its portability on small devices like an UMPC.

The adaptation of the reality is made possible by the use of interchangeable filters that enable a better integration of the ancient content in the reality. The tracking technology is based on a fast initialization followed by a robust 2D tracking of local features. This enables the suppression of planar makers usually seen in Augmented Reality scenarios.

The system can be used with two camera models: the classical 6-DoF one and a simplified one where only the camera's rotation is computed. We showed results of our application in the area of architecture and cultural heritage, where the system runs on an Ultra Mobile PC (Sony Vaio UX) with 15 frames/sec. For now the filters can be chosen by the application developer or online by the user. In a future version of our system, we will investigate the automatic detection of the right filter for the best integration of the content in the real image.

References

[AG97] AMIT Y., GEMAN D.: Shape Quantization and Recognition with Randomized Trees. *Neural Computation 9*, 7 (1997), 1545–1588.

[FB81] FISCHLER M., BOLLES R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[Fis05] FISCHER J.: Stylized augmented reality for improved immersion. *Proceedings of IEEE Virtual Reality* (2005), 195–202.

[HZ01] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, 2001.

[LLF05] LEPETIT V., LAGGER P., FUA P.: Randomized trees for real-time keypoint recognition. *Conference on Computer Vision and Pattern Recognition 2* (2005), 775–781.

[ST94] SHI J., TOMASI C.: Good features to track. *EEE Conference on Computer Vision and Pattern Recognition* (June 1994), 593–600.

[Str05] STROSS C.: *Accelerando*. Ace Hardcover, 2005.

[TK91] TOMASI C., KANADE T.: Detection and tracking of point features. *Carnegie Mellon University Technical Report CMU-CS-91-132* (Apr. 1991).

© The Eurographics Association 2008.