The ORIGAMI Project: Advanced tools for creating and mixing real and virtual content in film and TV production

O. Grau, R. Koch, F. Lavagetto, A. Sarti, S. Tubaro and J. Woetzel

Abstract: The goal of the EC-funded IST project ORIGAMI was the development of advanced tools and new production techniques for high-quality and seamless integration of real and virtual content in film and TV productions. In particular, the project focused on pre-production tools for automatic virtual set creation through image-based 3-D modelling of environments and objects. One key goal of the project was to achieve real-time in-studio pre-visualisation of the virtual elements (objects and actors) of the extended set. In this contribution the studio pre-visualisation system that has been developed for the project is illustrated, and its usage within a high-end film and TV production environment is described. Furthermore an overview of the developed solutions for automatic generation of 3-D models of environments, static objects and (moving) actors is given.

1 Introduction

Although the use of high-end animated computer models has become quite common in film and TV productions, the seamless interaction and the integration between real and virtual characters are still complex and expensive. In fact, aside from the cost related to 3-D model creation, animation and rendering, there is the actual cost of shooting the 'takes' where the real actors give the impression of seeing the virtual elements of the set and interacting with the virtual actors. The interaction with the virtual elements and characters of the set is usually a very challenging task, as both actors and crew have little, if any, feedback to figure out the exact position of the virtual object that the actors are to interact with. This problem becomes particularly critical when the actor is expected to be engaged in a conversation with a virtual actor, as we are very good at detecting missed eye contact. In addition to such problems, the lack of visual feedback makes it very difficult for actors to correctly guess the timing of their reactions to the virtual actor's activity. As a matter of fact, actors are usually forced to work in rather difficult conditions, as positional and synchronisation cues are usually given in terms of rough marks on the floor and timed gestures made by the crew.

The situation is no easier for camera crew and director. Those film directors who choose to make extensive use of virtual set extension and virtual actors, in fact, must accept dramatic changes in their way of working, as they cannot easily have a preview of what the result will look like at the end of the post-production work. The lack of a 'what-you-see-is-what-you-get' (WYSIWYG) approach to the filming of live action in a chroma-key studio (a controlled environment where actors perform in front of a blue screen for chroma-keying purposes), for example, makes it difficult to plan camera trajectories and lighting. As the film director and the director of photography can do little but guess the final outcome, they often end up either keeping what they blindly obtain (and settling for less than optimal results), or having to pay for a number of expensive screen tests (prototypes of digital effects).

Alternatively the use of 3-D models allows enhanced optical interaction, that means occlusions, casting of shadows and reflections of background, objects and actors. If these are acquired as 3-D models then the camera framing can be chosen in the post-production phase when all virtual components are integrated into the scene.

The ORIGAMI (the project name is not an acronym but a metaphor, as it also is aimed at giving 3-D life to a 2-D reality) project [1] focused on overcoming many of these difficulties, as it developed a set of new tools for planning and pre-visualisation in film and TV production. The techniques range from image-based modelling to real-time visual feedback techniques and can be grouped into two categories, as depicted in Fig. 1. The pre-production tools include solutions for automatic generation of 3-D models of environments and static objects. The studio system is used in the production phase to acquire the footage, i.e. film or video material of the actors. Moreover the studio system provides real-time feedback of the virtual objects, including background and static models and avatars to the actor. A pre-visualisation of the composited virtual components

IEE Proceedings online no. 20045134

doi: 10.1049/ip-vis:20045134

Paper first received 31st July 2004 and in revised form 23rd February 2005

O. Grau is with the BBC Research and Development, Kingswood Warren, Tadworth, Surrey, KT20 6NP, UK

R. Koch and J. Woetzel are with the Institute of Computer Science, Christian-Albrechts-University of Kiel, 24098 Kiel, Germany

F. Lavagetto is with Diploma di Informatica, Sistemistica e Telematica, Via all'Opera Pia 13 - 16145 Genova, Italy

A. Sarti and S. Tubaro are with Diploma di Elettronica e Informazione - Politecnico di Milano. Piazza Leonardo Da Vinci 32, 20133 Milano, Italy

Work funded by the Information Society Technology (IST) programme within the IST-2000-28436 project 'ORIGAMI: A new paradigm for high-quality mixing of real and virtual'.

E-mail: oliver.grau@rd.bbc.co.uk

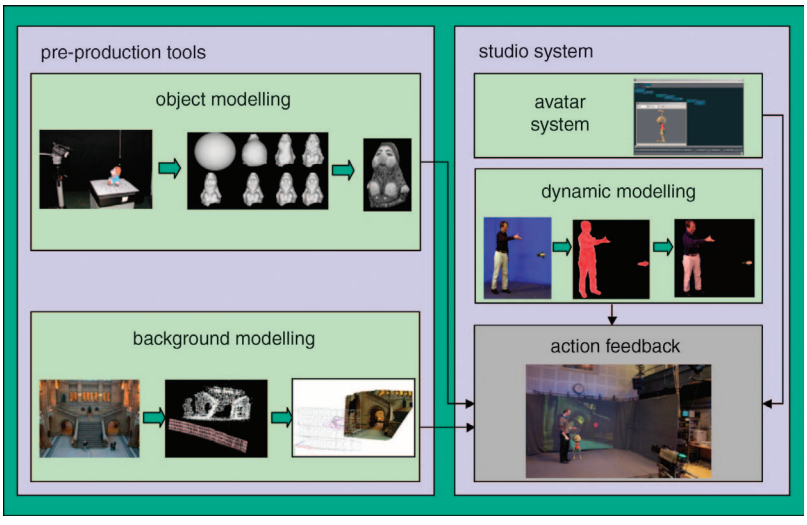


Fig. 1 *Overview of the ORIGAMI tools: objects, background and avatars are modelled off-line*
The actor's model is constructed as the action takes place in the studio, which is able to provide a visual feedback of the virtual scene elements to the actor

and a low-quality real-time 3-D actor model is also provided to the director or camera operators on set.

1.1 Related work

Visualisation in the studio is an increasingly important problem. For the director, camera operators and actors there is a strong need to have an on-set pre-visualisation of the composited scene. This has been addressed in the past by using virtual studio techniques to get the composited scene on a studio monitor [2]. Unfortunately the actor still has the problem of looking into empty space. Hence synchronisation, particularly of the eye-lines with a moving virtual object is a major problem.

An approach to provide the actor or presenter with a visual cue of objects in a virtual set is described in [3]. The system projects an outline of virtual objects onto the floor and walls. However, this method is restricted to show only the point of intersection of the virtual objects with the particular floor or wall. That means a virtual actor in the scene would only be visualised as footprints and the eye-line problem persists.

Systems that do provide the required functionality are projection-based virtual reality (VR) systems, like the CAVE [4]. The main application of these systems is to provide an immersive, collaborative environment. Therefore, a CAVE system tracks the position of the viewer's head and computes an image for that particular viewpoint, which is then projected onto a large screen forming one wall of the environment. Although such collaborative projection-based VR systems would probably provide a good immersive feedback for an actor, they are not designed to create 3-D models of the person.

Our system was therefore designed to allow the creation of the actor's visual hull using chroma-keying based on a special retro-flective cloth, that allows the actor to see the projected images, while the camera is equipped with a ring of blue LEDs. The light from the LEDs is reflected back to the camera and allows a robust chroma key. Section 4.1 gives an overview on this technique. A more detailed description can be found in [5].

The modelling of dynamic actors is briefly described in the Section 4.2. It builds on our work in [5, 10] based on a visual hull reconstruction from a multi-camera system. For the real-time pre-visualisation a fast version has been developed. For the generation of special effects a new improved algorithm was developed that uses super-sampling in order to minimise visual artefacts in the 3-D actor models for use in post-production rendering tools.

To reconstruct the environment that surrounds actors and virtual objects, we developed pre-production tools for automatic modelling and rendering the environment from fully uncalibrated image sequences. The calibration of the images is based on the uncalibrated structure from motion (SFM) approach [7–10]. We extended it by the concept of using a rig of multiple rigidly coupled cameras (see Section 3.1).

With the camera calibration at hand, one can obtain dense disparity maps for image pairs using stereo algorithms [11–16]. We follow a dynamic programming algorithm extended by a pyramidal scheme [17, 18] to compute pair-wise disparity maps. We improved density and accuracy of the depth estimation using a hybrid temporal and spatial linking and fusion algorithm described in Section 3.2.

2 Object modelling

Modelling objects is common practice in film production, as it is a necessary step for the creation of not just set extension elements but also of virtual characters. The typical approach for this purpose starts from the digitisation of an existing object by means of a laser scanner. With this device it is possible to generate a number of dense point clouds, which are finally 'wrapped' by a closed surface (e.g. a triangle mesh). The final step consists of texture mapping the surface and selecting appropriate reflectivity attributes for the final rendering. For reasons of cost, however, laser scanners are normally used just on objects of modest size. When dealing with complex objects of larger size, alternative solutions need to be adopted, and not much is available in the market for this purpose. For example, laser range finders are only able to produce sparse depth maps and the positional feedback is usually quite difficult to have and exploit, which makes it difficult to create a closed (full 3-D) model of the imaged object. The goal of the activity on object modelling was to devise and develop a modelling strategy that enables the fast creation of complete object models, irrespective of the object's size, using a simple acquisition procedure based on commercial cameras. The idea is to acquire a sequence of images (e.g. a video or a set of stills) from all around the

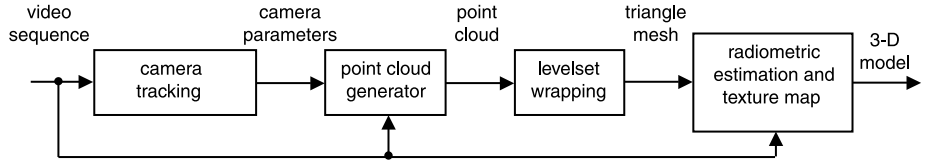


Fig. 2 *Object modelling workflow: from a video sequence to a complete 3-D model*

object without specific and restrictive constraints; automatically estimate from such images the camera parameters (position, orientation and intrinsic camera parameters); and finally generate the object model in an automatic fashion through the analysis of the available images. In order to rapidly achieve such goals in an automatic fashion and with a modest effort on the part of the user, the ORIGAMI consortium developed specific solutions for the following problems:

- image-based camera tracking for 'walk-around' trajectories;
- fast and global modelling of closed surfaces;
- wrapping unstructured point clouds;
- multi-view texture mapping of triangle meshes with view-dependent radiometric compensation.

We will later see that the most practical way to generate a closed surface from a set of images is to first generate the geometry of the object (point cloud or a registered set of depth maps) and then to retrieve the topology that describes how such points are connected together into a surface (wrapping). A general workflow scheme that describes the process of generation of a complete 3-D object model from a walk-around video sequence can thus be given as in Fig. 2.

In the following Subsections we will describe the walk-around camera tracking process, and the surface generation process. For an account of how radiometric estimation and multi-view texture mapping is performed, the reader may refer to [19].

2.1 Camera tracking for 'walk-around' trajectories

Walk-around trajectories are characterised by the fact that initial and final camera positions are close to each other. This fact can be exploited in order to reduce the unavoidable drift that camera-tracking solutions are affected by. A typical walk-around trajectory is obtained by placing an object onto a rotating turntable and keeping the camera still while the object undergoes a complete revolution. Another example is the free-form walk-around trajectory obtained with a hand-held camera. If we place a copy of the first frame at the end of the sequence, and if we compute the camera motion over the extended sequence, then the inevitable difference between the initial and the final camera location will give us a measure of the drift accumulated by the camera tracker ('closure' error). Even using the best commercial camera trackers, this drift turns out to be quite relevant and its magnitude can easily lead to unacceptable modelling errors. Furthermore, the computational time turns out to be quite relevant.

In order to limit the processing time we adopted and improved an approach to camera tracking based on the extended Kalman filtering [19–22], based on the elimination of the error build-up by the introduction of appropriate closing constraints. The measurements of the EKF are the image coordinates of the tracked features. The state vector includes the 3-D orientation and position of the camera, its focal length, and the position of the cloud of tracked points in the observed 3-D scene, parameterised like in [19, 20, 22]. The 3-D location of each point is, in fact, described by just one parameter that describes the depth associated to an image feature, with respect to the image plane, therefore we have one parameter per tracked point.

This approach is well-known to be computationally efficient but its accuracy and robustness crucially depend on how we manage the appearance and the disappearance of features throughout the sequence and how we deal with estimation drift. In order to achieve a fast, stable and accurate camera tracking, we used the fact that the camera trajectory is closed or nearly closed as a constraint for our estimation, and as additional information for the management of features. In particular, if we know that the camera trajectory is, in fact, closed (turntable case), we can assume that the initial and final camera positions are the same.

In this case we can repeat the first frame at the end of the sequence. If the motion discontinuity that we introduce this way is not too large, then the last motion step can be estimated correctly and the closure constraint on the extended sequence can be successfully exploited.

When dealing with the free-form walk-around motion of a hand-held camera, then we cannot exploit the same constraint as before, but we can still assume that any 3-D point structure that is visible in the first view is also visible in the last view. This is a far less strong condition that is met by the vast majority of walk-around sequences and still guarantees significant improvements in the quality of the results. In order to deal with the continuous disappearing and reappearing of image features, the system tracks a large number of features, and uses a RANSAC (RANdom SAmple Consensus) method [23] for rejecting those features that are poorly tracked. This, of course, needs to be dealt with carefully in order to keep the computational complexity low and achieve an accurate and reliable estimation of the camera trajectory in a very short time.

In order to assess the performance of our camera tracker, we conducted some comparative tests with an open-loop (no closing constraints) camera tracker based on a RANSAC-based selection and tracking of feature points and the application of projective constraints for self-calibration. We performed four tests of decreasing level of difficulty: a textured object; a textured object on a sheet of newspaper laying on a flat surface; an object with triangular markers on it, placed on a white surface; an object with triangular markers on it, placed on a flat surface also with triangular markers. The computational time of the reference camera tracker was about 45 minutes to process a video sequence of 300 frames (using a PC equipped with an AMD processor at 1.33 GHz and 512 Mb RAM). As far as the accuracy is concerned, the translational closure error ranged from 3 to 8 cm (out of a trajectory that was approximately circular with about 80 cm radius); the camera orientation error ranged from 0.25 to 0.3 degrees; while the closure rotational error ranged from 2 to 7 degrees. With our technique, with a

sequence of 720 × 576 pixels, the tracking speed is about 2 frames per second, while with a sequence of 1512 × 1024 pixels, the speed is about 1 frame per second. As far as the accuracy is concerned, the translational closure error ranged from 0.5 mm to 2 mm; the orientation error ranged from 0.05 degrees to 0.15 degrees; and the closure rotational error ranged from 0.05 degrees to 0.12 degrees.

2.2 Direct image-based object modelling

Walk-around camera sequences give us a great deal of information on the object structure, which can be exploited for the automatic modelling of complete objects. In order to obtain a global 3-D model of the imaged object, we developed specialised methods based on the steering of the evolution of the zero levelset of a volumetric function [24].

Levelset-based solutions are able to deal with complex topological structures in a global and robust fashion. The idea is to model a closed surface γ in an implicit form as $\gamma = \{\mathbf{x}|\psi(\mathbf{x}) = 0\}$, where $\psi(\mathbf{x})$ is a properly defined volumetric function of the space coordinates $\mathbf{x}$. It is a common choice to assign $\psi(\mathbf{x})$ the distance d between $\mathbf{x}$ and the surface, and give it a sign that depends on whether the point $\mathbf{x}$ is inside or outside of the surface. We can then proceed by defining a temporally evolving volumetric function whose levelset zero 'sweeps' the whole volume of interest until it takes on the desired shape under the influence of some properly defined 'external action'. The levelset evolution is, in fact, a partial differential equation (PDE). A typical choice in the literature [24] is to use a Hamilton-Jacobi PDE, which can be discretised into the update equation of the form $\psi(\mathbf{x}, t + \Delta t) = \psi(\mathbf{x}, t) - |\nabla\psi(\mathbf{x}, t)|F(\mathbf{x})\Delta t$, where the velocity function $F(\mathbf{x})$ is bound to be orthogonal to the zero levelset and can be quite arbitrarily defined in order to steer the front propagation toward a desired shape. This is a very convenient setup that allows us to define F in such a way to account for a number of steering terms such as: local curvature (for maximally smooth implosion, distance from 3-D data (where available), agreement between textures obtained by projecting the available images onto the evolving front, etc.

The idea of using a modelling solution based on the direct steering of the front evolution based on a measure of the texture agreement was originally proposed by Faugeras and Keriven [25] and was then improved in [26] in such a way to achieve more visually accurate results with a reduced computational effort. One key feature of the method is that it enables the front to evolve in a multi-resolution fashion. In fact, the modelling algorithm start with a very low resolution voxset. When the propagation front converges, the resolution increases and the front resumes its propagation. The process is repeated until we reach the desired resolution. This choice, besides improving the optimisation process, dramatically reduces the number of iterations with respect to a fixed-resolution approach. In order to avoid missing important details as the front evolves at low resolution, the method incorporates a mechanism of recovery of lost details based on front backtracking. The idea is to keep track of all the voxels on the propagating front whose cost is below a certain threshold. After the 'implosion' of the levelset, we let the propagation front evolve while driven by a different cost function that depends on the distance between the surface and such points. This operation makes the surface backtrack and 'climb up' the formerly lost details. The method exhibits a remarkable robustness against lack of texturing and topological complexity and has the advantage of producing a final model in one single step. Alternatively the required computational effort is still quite significant (on a PC equipped with a P4 processor at 1.8 GHz clock, the model of Fig. 3 is constructed in about 45 minutes) and the unavoidable smoothing action of the motion-by-curvature term of the front velocity sometimes tends to eliminate relevant details of the model.

2.3 Indirect image-based object modelling

Despite the significant effort put into reducing the computational complexity of methods based on the Hamilton-Jacobi PDE, the computational effort associated with the above class of solutions is still far from being of practical interest for the applications envisioned in the ORIGAMI project. In fact, we would like the computational cost to be low enough to allow us to see the front evolve and interact with its evolution where needed. One reason for the high cost of the solutions described in Section 2.2 is that the front evolves according to a Hamilton-Jacobi PDE operating on a 3-D function that represents the distance from the zero levelset. This means that the volumetric function tends to die out slowly as we move away from the front. In order to significantly speed up the computation we thus need to think of different PDEs and volumetric functions that describe the evolving front in a more compact and convenient fashion. In order to achieve this goal we decided to start over by separating the problem of geometry estimation (generation of a point cloud or a depth map) from that of the estimation of the scene topology (generation of a triangle mesh from a point cloud). This 'indirect' modelling approach allows us to take advantage of the wide literature available on geometry estimation without giving up the well-known advantages associated to volumetric methods as far as topology estimation is concerned. As far as geometry estimation is concerned, the literature is rich with depth map estimation techniques that are able to achieve high-quality results with a modest computational effort. Examples of such sort are the methods based on graph cuts [13], which are described and extensively used in Section 3.2 for environment modelling. As far as topology

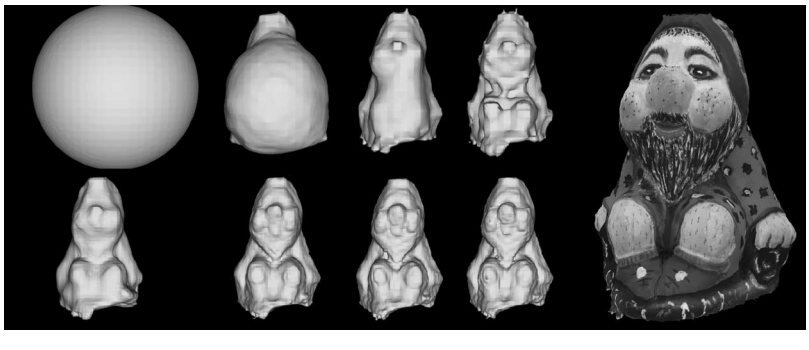


Fig. 3 *Example of front evolution with the direct image-based levelset method and final textured model*

estimation is concerned, we developed a novel method for the fast modelling of closed surfaces from sets of unorganised sample points. This surface 'wrapping' is able to perform correctly even when the point cloud is not complete, i.e. when there is a lack of geometric data in some not-easily-accessible portions of the surface.

In our approach to levelset-based surface modelling, instead of using a traditional Hamilton-Jacobi PDE acting on a distance function, we refer to the Navier-Stokes equations [27], with particular reference to the PDE that describes the conservation of mass in fluid dynamics. This choice corresponds to modelling the front evolution as that of a fluid that tries to fill up the space under the influence of some data-induced action. The volumetric function that this PDE acts upon, in fact, describes the amount of fluid that is present in a voxel. This function is thus zero when the voxel is empty, and takes on a unitary value when the voxel is full. The levelset 0.5 thus represents the surface that bounds the fluid in space. This volumetric function has a significant advantage over the more common choice of the distance function, as it varies from zero to one very rapidly around the surface boundary, therefore its description is extremely compact (with obvious computational advantages).

The idea behind this approach is to choose a voxset that entirely contains the object to be modelled, and initialise all of its voxels to zero. This corresponds to assuming that the space is initially empty. We then place a number of sources of fluid at the boundary of the voxset and let the fluid evolve according to the conservation of the mass PDE while driven by a particular definition of the velocity vector $\boldsymbol{v}$ [27]. The general form of conservation law for a system bounded by a closed surface S can be expressed in terms of variations of the specific fluid mass F owing to the flux contributions of the surrounding points and of the sources. The flux vector $\boldsymbol{G}$ has a diffusive part $\boldsymbol{G_D}$ and a convective part $\boldsymbol{G_C}$. In its general form, a conservation law can be expressed as

$$\frac{\partial}{\partial t}\int_{\Omega} F d\Omega + \int_{S} \boldsymbol{G}\cdot d\mathbf{S} = \int_{\Omega} Q_V \cdot d\Omega + \int_{S} \mathbf{Q}_S \cdot d\mathbf{S} \quad (1)$$

which means that the temporal variation of F within the volume Ω plus the net contribution from the incoming fluxes through the surface S (second term) must be equal to the contributions from the volume sources Q_v and the surface sources $\mathbf{Q}_S$ (written using Gauss theorem). This law can be rewritten in differential form as

$$\frac{\partial F}{\partial t} + \nabla \cdot \boldsymbol{G} = Q_V + \nabla \cdot \mathbf{Q}_S \quad (2)$$

The convective part of the flux vector $\boldsymbol{G_C}$, associated to F in a flow of velocity $\boldsymbol{v}$ is the amount of fluid mass transported with the motion, and is given by $\boldsymbol{G_C} = \boldsymbol{v}F$. The diffusive flow is usually proportional to the gradient of F, i.e. $\boldsymbol{G_D} = \gamma \nabla F$, where γ is the diffusivity constant. In physical systems this flow is normally associated to molecular and thermal motion.

In our algorithm we point the velocity $\boldsymbol{v}$ of the fluid towards the nearest sample point, which acts as an attractor. As the magnitude of the speed vector is proportional to the distance from the nearest point, the fluid boundary tends to gently take on the shape of the desired surface. The diffusive term is included to promote a more physical fluid evolution, which leads to a smoother boundary of the fluid. This diffusive term weighs the contributions of the closest points with a Gaussian function, whose standard deviation σ plays a role that is similar to the diffusivity term γ. A good choice for the evolution equation turns out to be

$$\frac{\partial F}{\partial t} = \frac{\int_{\Omega} F(\mathbf{x})|\mathbf{v}(\mathbf{x})| e^{\frac{x^2}{2\sigma^2}} d\mathbf{x}}{\int_{\Omega} |\mathbf{v}(\mathbf{x})| e^{\frac{x^2}{2\sigma^2}} d\mathbf{x}} - F(\mathbf{x}) \quad (3)$$

These equations describe the basic evolution laws. However, the correct behaviour of the front evolution is achieved by introducing additional terms that provide the user with physical control over the surface evolution. For example, in order to inhibit interstitial flow where samples are sparse, we introduce a viscosity term, whose aim is to reduce the local curvature of the front. There are, however, regions where this interstitial flow is desirable and should be promoted, and these regions are usually those where conventional levelset methods fail. For example, thin plates, spikes and narrow holes are hard to reproduce with levelset methods based on the usual Hamilton-Jacobi PDE as they are rounded off or levelled out by the unavoidable motion-by-curvature term of such class of PDEs. With our fluid-dynamic approach we can encourage the 'interstitial' flow of the fluid through what we call the medial axis steering.

The medial axis [28] provides us with simple topological information on the shape of the features that we want to retain. In our case the concept of medial axis is replaced by its discrete counterpart, which is the Voronoi diagram of the point cloud. If, during its evolution, the front meets the medial axis, then we know that the levelset is passing through one of the interstitial features described above. This means that the fluids stopped too early and failed to flow inside it. The problem can be solved by turning the voxels that lie on the medial axis into sources of fluid (this is achieved easily by assigning the corresponding voxels a fixed value one irrespective of the PDE evolution). This has the effect of locally restarting the evolution of the system. An example of application of medial axis steering is shown in Fig. 4, where the medial axes is used for encouraging the fluid to carve voxels out of the dragon's mouth.

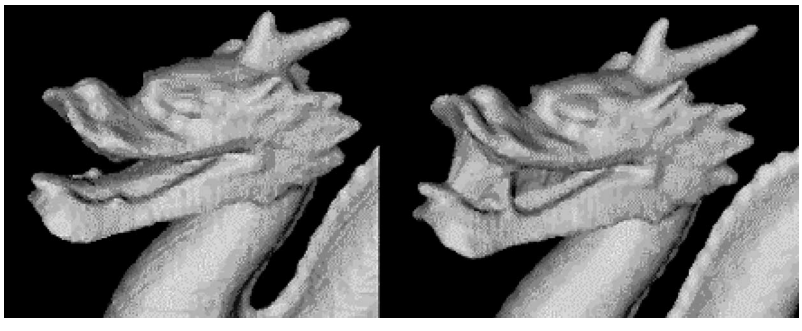


Fig. 4 *Example of indirect image-based surface modelling with medial surface steering of the front evolution*
The medial surface here helps the fluid penetrate the dragon's mouth and carve voxels out of it (left). Front evolution without medial surface steering (right)

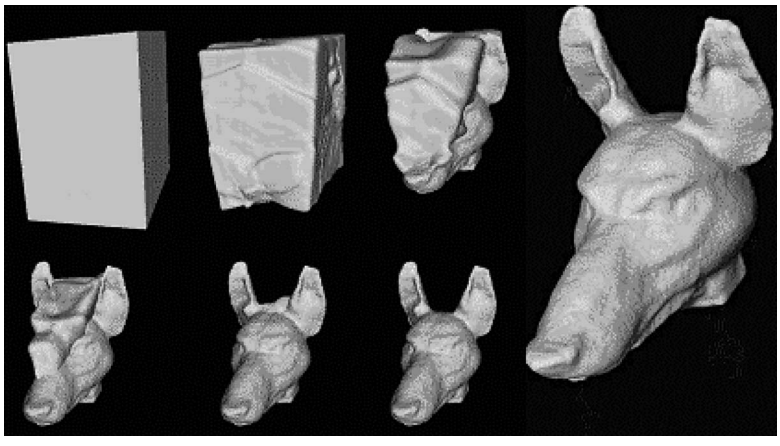


Fig. 5 *An example of indirect image-based surface modelling*
A point cloud was acquired with an image-based method applied to a clay sculpture of a wolf (initial voxset of 300 voxels per side)

One remarkable feature of this approach is its speed. For example, in order to complete the front evolution of Fig. 5 (about 50 million voxels) with an Intel P-IV 3 GHz processor with 1 GB of RAM under Windows™ 2000, our non-optimised algorithm takes about 1.5 minutes (without multiresolution). This includes the time that it takes the algorithm to generate several intermediate triangle meshes (with a marching cube algorithm) in addition to the final one, for a step by step visualisation of the front evolution.

3 Modelling the environment

In order to construct the environment that surrounds actors and virtual objects, we developed solutions for fully automatic generation and rendering of 3-D models of natural environments from fully uncalibrated image sequences. The acquisition is made with a hand-held camera or a multi-camera rig, moving with no restrictions on its motion. Since we want to avoid having to place markers in the scene, the camera pose and calibration is estimated directly from the image sequence. This process is described in Section 3.1.

With the calibrated image sequence at hand, one can obtain dense depth maps with multi-stereo image disparity estimation (Section 3.2). These dense depth maps can be used to generate virtual views of the scene. We follow two approaches to render virtual views. These methods are described in Section 3.3.

3.1 Calibration

Camera tracking and calibration are based on an extension of the uncalibrated structure from motion (SFM) approach [7–10]. An initial relative pose of an image can be estimated by feature point correspondences (see Fig. 7) based on constraints of the fundamental geometry for the first images. Every added view is used to refine the 3-D geometry of the feature points and the estimated camera poses. 3-D feature points are tracked through multiple images to determine reliable features and estimate the camera pose more robustly (see Fig. 7). Constraints on the epipolar geometry and the reliability of the features' 3-D position like the back projection error are used for refinement of the 3-D structure, the matches and the pose of the cameras. With each image of the sequence the pose and covariance of the 3-D feature points is updated.

When the image sequences to be processed are very long, there is a risk of calibration failure for several reasons. One of these reasons is that the calibration, as described above, is built sequentially by adding one view at a time. This may result in accumulation errors that introduce a bias to the calibration. In our visual-geometric approach, multiple cameras may be used to scan a 2-D viewing surface by moving a rigid multi-camera rig through the viewing space of interest. A more reliable means of calibration is therefore to use an n-camera rig that simultaneously captures a time-synchronised 1-D sequence of views. When this rig is swept along the scene of interest, a regular 2-D viewpoint surface is generated that can be calibrated very reliably by concatenating the different views in space and time.

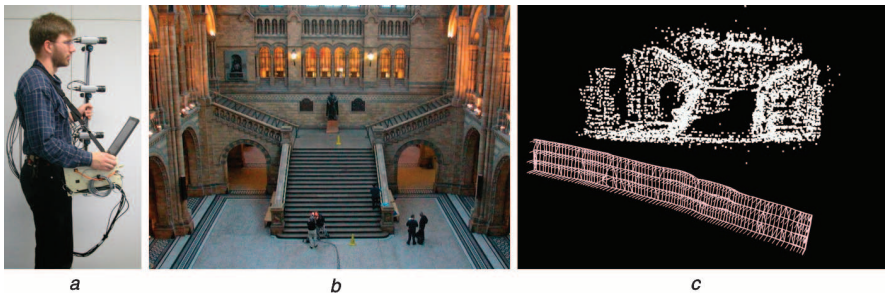


Fig. 6 *Portable multi-camera rig, overview of the museum scene captured at ground level, and perspective view on the reconstruction result of 3-D feature points and path of the 4-camera rig moving horizontally*
a Multi-camera rig
b Overview of museum scene
c Perspective view on the reconstruction result

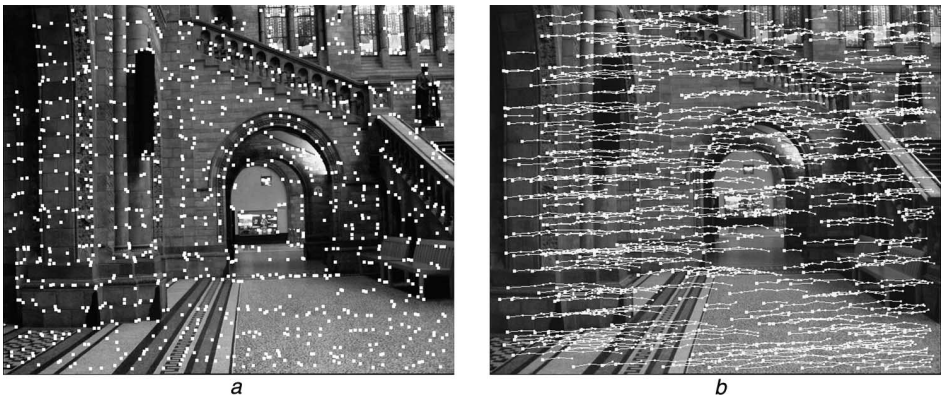


Fig. 7 *Detected corners for one image of the 'museum' scene and tracked features for a sequence of images from one camera*

a Detected corners
b Tracked features

For each recording time, a number of n simultaneous real views of the scene are obtained and can be used to reconstruct even time-varying scenes. When the camera rig is moved, a sequence of k images is obtained for each of the n cameras. Thus, one may obtain a 2-D viewpoint surface of $k \times n$ views by simply sweeping the camera rig throughout the scene [29]. By concatenating the camera motion in time and the different cameras on the rod, a 2-D viewpoint surface is built that concatenates all real views. We have therefore developed a multi-camera calibration algorithm that allows to actually weave the real views into a connected 2-D viewpoint mesh exploiting the quasi static relative pose of the n cameras of the rig [29]. Please note that the relative pose of the rigidly coupled cameras of the rig can be estimated directly from the static scene. The rigidity is exploited to estimate the relative pose of the rigs from all cameras simultaneously. To facilitate correspondence estimation between the views, we use a rectifying homography that compensates relative camera rotation between the n coupled views.

Our multi-camera tracking system was used in a test production to generate models of the entrance hall of the London Museum of Natural History. Four cameras were mounted in a row on a vertical rod and the rig was moved horizontally along parts of the entrance hall while scanning the hallways, stairs, and a large dinosaur skeleton. While moving, images were taken at about 3 frames/s with all 4 cameras simultaneously. The camera tracking was performed by 2-D-viewpoint meshing [9] with additional consideration of camera motion constraints. Prediction of potential camera pose is possible because we know that the cameras are mounted rigidly on the rig. We also can exploit the fact that all 4 cameras grab images simultaneously [29]. Figure 6 shows the portable acquisition system with 4 cameras on the rod and 2 synchronised laptops attached by a digital firewire connection, and an overview of parts of the museum hall. The camera rig was moved alongside the stairs and 80×4 viewpoints were recorded over the length of the scan. Figure 6 displays the camera tracking with the estimated 320 camera viewpoints, their topology and the reconstructed 3-D feature point cloud obtained by the SFM method. The outline of the stairs, the back walls and the floor is visible in the reconstructed feature point cloud directly.

3.2 Depth estimation

Once the cameras are calibrated we can use techniques based on image correspondences to estimate the scene depth. We rely on stereo matching techniques that were developed for dense and reliable matching between adjacent views. The small baseline paradigm suffices here as we use a rather dense sampling of viewpoints.

3.2.1 Stereoscopic disparity estimation:

Dense stereo reconstruction has been investigated for decades but still poses a challenging research problem. This is because we have to rely on image measurements alone and still want to reconstruct small details (requires a small measurement window) with high reliability (requires a large measurement window). Traditionally, pair-wise rectified stereo images were analysed exploiting geometrical constraints along the epipolar line as described in [11, 12]. More recently, generalised approaches were introduced that can handle multiple images, varying windows and higher order constraints [12, 13]. Also, real-time stereo image analysis has become almost a reality with the exploitation of the new generation of very fast programmable graphical processing units for image analysis [14–16]. We are currently using a hybrid approach that needs rectified stereo pairs but can be extended to multi-view depth processing.

We use an area-based disparity estimator on rectified images for dense correspondence matching along the epipolar lines with the similarity measure normalised cross correlation with a small kernel size of typically 5×5 pixel along the scan-lines. Dynamic programming is used to evaluate extended image neighbourhood relationships and a pyramidal estimation scheme allows to reliably deal with very large disparity ranges [17].

3.2.2 Multi-camera depth map fusion:

For a single-pair disparity map, object occlusions along the epipolar line cannot be resolved and undefined image regions (occlusion shadows) remain. One should notice that occluded regions are partially detected by the ordering constraint in the pyramidal dynamic programming approach. They can be filled with multi-image disparity estimation. The geometry of the viewpoint mesh is especially suited for further improvement with a multi-viewpoint refinement [18]. For each viewpoint a scan of a number of directly adjacent or nearby viewpoints exist that allow correspondence matching. Since the different views are rather similar and the scene is static we will observe every object point in many nearby images even through successive frames. This redundancy is exploited to improve the accuracy of the depth estimation. Disparity estimations between adjacent viewpoints with small baseline are linked to longer chains to establish correspondences for distant

cameras. The depth value of each object point is refined to higher precision by Kalman filtering of all triangulated depth values in one chain.

We can further exploit the imaging geometry of the multi-camera rig to fuse the depth maps from multiple spatially neighbouring images into a dense and consistent single depth map. For each real view, we can compute several pair-wise disparity maps from adjacent views in the viewpoint surface. The topology of the viewpoint mesh was established during camera tracking as described in the previous Section. Since we have a 2-D connectivity between views in horizontal, vertical, and even diagonal directions, the epipolar lines overlap in all possible directions, as illustrated in Fig. 8. Hence, occlusion shadows left undefined from single-pair disparity maps are filled from other view points and a potentially 100% dense disparity map is generated.

In addition, each 3-D scene point is seen many times from different viewing directions. This allows to robustly verify its 3-D position. For a single image point in a particular real view, all corresponding image points of the adjacent views are computed. After triangulating all corresponding pairs, the best 3-D point position can be computed by robust statistics and outlier detection, eliminating false depth values on the cost of slightly less dense depth maps [18]. Thus, reliable and dense depth maps are generated from the camera rig. The linking scheme of spatially and temporally adjacent camera shots and results are illustrated in Figs. 8 and 9.

As an example, a scene with a dinosaur skeleton of highly complex geometry was processed and depth maps were generated with different neighbourhoods. Figure 9 shows an original image and the corresponding depth maps for a varying number of images taken into consideration. The depth maps become denser and more accurate as more and more spatially and temporally adjacent images are evaluated. A single raw disparity estimation of a vertical stereo image pair results in a fill rate of only 63.2% owing to the extremely high amount of occlusions (Fig. 9*b*). Exploiting the 2-neighborhood, known as trifocal stereo, improves the density to 73.0% but is still insufficient (Fig. 9*c*). For

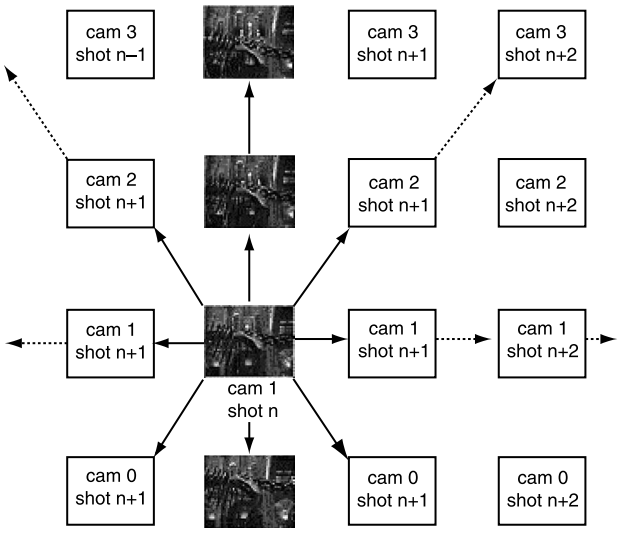


Fig. 8 *Linking scheme of multi-stereo fusion for 8-neighbourhood*

Four temporal synchronised shots (vertical) of the multi-camera rig are linked to their spatial neighbour scans in horizontal and diagonal directions

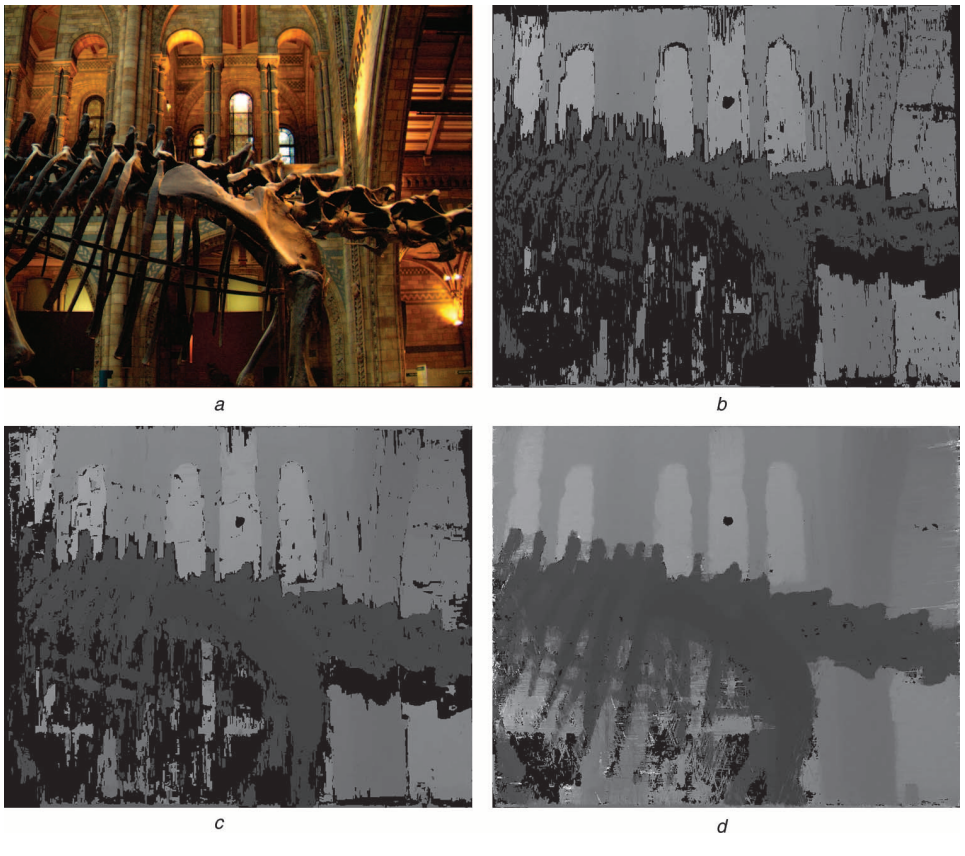


Fig. 9 *One original image (a) and depth maps computed from the dinosaur sequence*

b Depth map from single image pair, vertically rectified (light = far, dark = near, black = undefined)
c 1-D sequential depth fusion from 3 adjacent views
d Result using our depth fusion with 8-neighbourhood linking scheme

Fig. 10 *Original image and corresponding depth map*

a One of the original images
b Corresponding depth map, light grey = far, dark grey = near, black = undefined

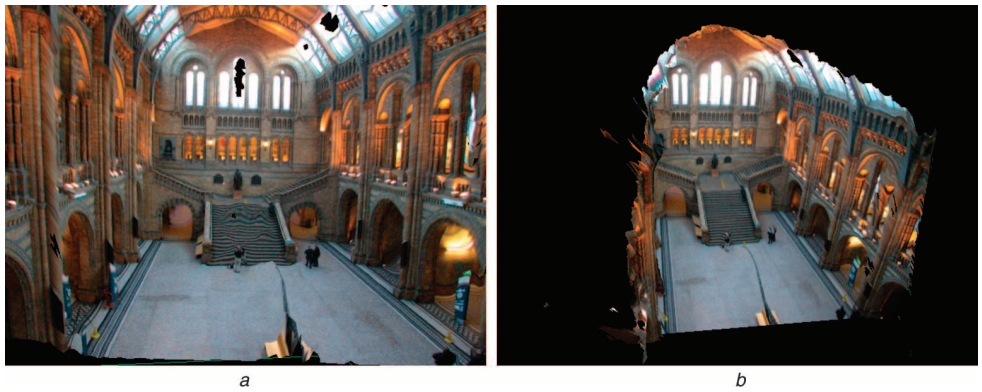


Fig. 11 *Two virtual views on the reconstructed geometrical VRML model*

a Full-automatically generated model
b Model manually refined fitting planes in the area of the stairs, the windows and the floor

images in an 8-neighbourhood as shown in Fig. 8, the fill rate of 95.5% (Fig. 9*d*) is very dense. However, some outliers (white streaks) can be observed which are due to the repetitive structures of the ribs consistently matched wrong in all adjacent images. These errors must be eliminated using prior scene knowledge since we know that the scene is of finite extent. A bounding box can be allocated that effectively eliminates most gross outliers.

3.3 Virtual views

Starting from the acquired calibrated views and associated depth maps, we can generate geometrical surface mesh models. The depth maps are triangulated, meshed and textured with the original colour images. Meshing is based on the 2-D topology of the images with simultaneous consideration of truncation at depth discontinuities. These geometrical models can be saved in standard formats, e.g. VRML and are useful for planning and pre-visualisation applications. Remaining holes and artefacts can be removed easily by post-processing through local interpolation or plane fitting (Fig. 11) if higher quality is desired. For example some applications may require a perfectly planar surface for realistic floor contact of actors in the virtual environment and shadow casting. Figure 10 shows an image of the scene and the computed depth map. Moving persons violate the static scene constraint resulting in small artefacts. Minimal user interaction is required to select artefacts and automatically replace their geometry by interpolation from correctly reconstructed spatially adjacent areas. Figure 11 illustrates two virtual views of the generated geometrical 3-D environment model.

However, owing to incorrect camera calibration and non-static scenes it is often not possible to generate one globally consistent 3-D-model from long image sequences. In this case we follow a novel approach for interactive rendering of virtual views combining the concepts of depth-compensated image warping and view-dependant texture mapping following the idea of plenoptic rendering.

4 3-D studio production system

The studio system provides three main functions to support the on-set work in real-time and the post-production in an off-line phase. The generation of dynamic 3-D models of actors, an avatar animation module and an on-set feedback module. These modules are described later in this Section. The following Section gives a brief overview of the components of the studio system.

4.1 Overview of the studio system

Physically the studio system is composed of a number of modular components, as depicted in Fig. 12. The communication and data exchange between these components is predominantly based upon a local area network and standard IT components. The number of cameras can be varied depending on the available space in the studio and the specific production needs. Each camera (Cam-1, ..., Cam-N) is connected to a capturing server (Cap-1, ..., Cap-N). The capturing servers are standard PCs, equipped with a frame grabber card and a RAID disc array. This allows the incoming video from the cameras to be saved to disc as uncompressed video for processing in a later off-line

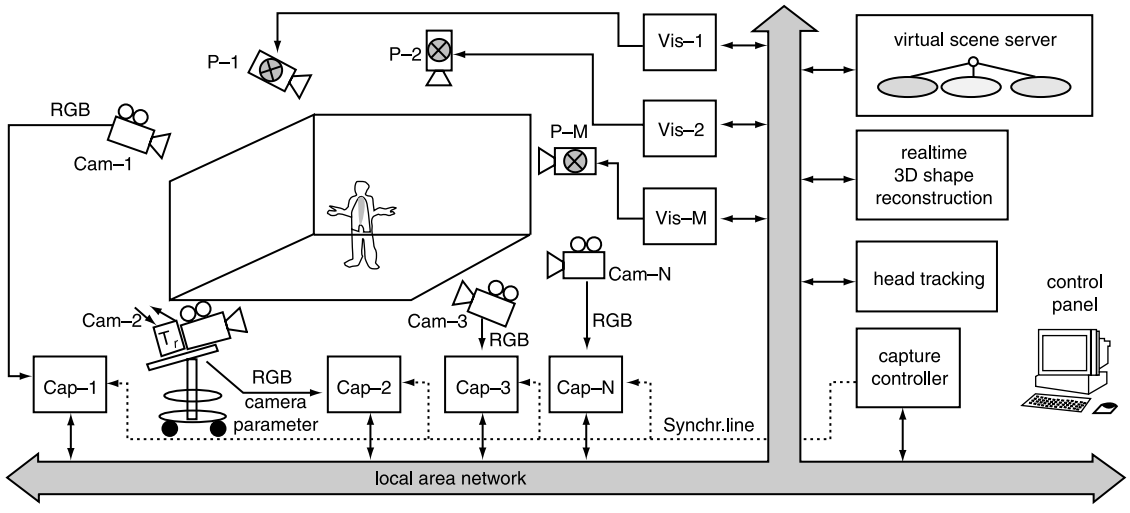


Fig. 12 *Overview of the studio system*

phase to produce high-quality 3-D models for the final programme.

Usually the cameras are fixed and their parameters are determined with a calibration procedure. In addition, cameras equipped with a real-time tracking system can be used, as shown for Cam-2 in Fig. 12. The real-time tracker delivers very accurate position and orientation and the internal camera parameters. We are using our previously-developed *Free-D* camera tracking system [30] for our experiments.

In addition to their function as an image sequence recorder, the capturing servers provide several online services over the network. On request they can send the latest grabbed image and provide a chroma-keying service. One of the components making use of this is the real-time 3-D shape reconstruction, which synchronously requests the keyed images as alpha masks from all the capture servers, and uses these to compute a 3-D model of the actor. The methods used by this module are discussed in the next Section. The 3-D data is used by the head tracker and passed to the virtual scene server. The actors' heads are passively tracked using a fast template matching filter that operates on the volumetric reconstruction of the actors shape (for details see [5]). The tracking accuracy is therefore limited by the volumetric resolution of the real-time 3-D reconstruction, which is currently approximately 8 cm. The head tracking is then computing the head position as the centre of mass of the identified voxels. The head position of the actor is used by the projection system to render a view-dependent image of the scene.

The virtual scene server provides a description of the virtual scene. This includes static scene elements, usually the virtual set, dynamic parts, for example any virtual characters involved in the scene, and the actor, provided by the 3-D shape reconstruction module. The virtual scene server synchronises all scene updates and distributes the scene updates to the visualisation servers (Vis-1, . . . , Vis-M). These are standard PCs that take into account the head position of the actor and render an image of the virtual scene, that is projected onto the walls and floor of the studio for the actor or presenter. Alternatively a composited view of the scene including the virtual scene and a 3-D model of the actor can be rendered on a normal PC display as a feedback for the director or camera operators of the production. On this basis the production team can decide on changes in the action on-set and do not have to wait until the scene is composited in post-production.

An important element for both the dynamic 3-D modelling component in addition to the action feedback module is the active chroma-keying system. It is based on a special retro-reflective cloth that was previously developed by the BBC R&D Department and is now available commercially under the name 'Chromatte™'. The studio cameras are equipped with a ring of blue (or any other colour) LEDs. The LED light is reflected back from the retro-reflective cloth to the camera and allows a robust chroma-key. Owing to the properties of the cloth the reflected light is even brighter than lights from studio lights and the data projectors used for the action feedback. A more detailed description of the studio system can be found in [5].

4.2 3-D modelling of dynamic content

Three-dimensional models of the actors are used for the real-time on-set visualisation and off-line for post-production. In the first case a rough quality is sufficient since the purpose is to generate a pre-visualisation, but it has to be in real-time. In the latter case the aim is to get a 3-D quality that can be used for special effects in final programme quality. In both cases we developed methods based on the shape-from-silhouette or visual hull concept. For the real-time version we implemented a technique based on line-segments [5].

For the use in post-production the 3-D models have to fulfil certain quality requirements. Visual hull reconstructions show a number of typical artefacts. The off-line version we implemented was addressing these problems and is based on an octree-representation and a new super-sampling technique [6], that gives smooth 3-D surfaces that can be used to generate video sequences. This approach reduces the sampling error that would otherwise be caused by a conventional volumetric reconstruction and the use of the marching-cubes algorithm for the generation of a surface model. The new approach extends the accuracy of the volumetric shape reconstruction by super-sampling without increasing the number of triangles in the 3-D model: the leaf nodes of an octree are further subdivided and the value of the original node is replaced by a counter of the number of sub-nodes that are found as belonging to the object. This value is then used in a standard marching cubes algorithm to compute a smoother 3-D surface.

Furthermore we are applying Gaussian smoothing to the 3-D models that further suppresses temporal artefacts that would be visible in a synthesised video sequence. The resulting 3-D model sequences can be loaded into a standard animation package together with background models and other virtual objects. The results Section gives an example of their use in a test production.

4.3 Pre-visualisation of avatar animation

While filming live action in the virtual studio actors are asked to simulate verbal and physical interaction with synthetic objects and characters that are not present in the scene but that will be added only later in post-production. The task of moving and acting coherently in space and time with this imaginary world is usually left to their imagination and skill with the only occasional help of rough marks on the floor and timed gestures made by assistants. To improve the actors' performance and simplify the post-production composition of natural video with 3-D animated graphics, we have integrated a new functionality in the studio system for pre-visualising the synthetic video components to the actors just at shooting time.

One of the most complex examples of natural-synthetic interaction is definitely that of a real human interacting with a virtual human. Humans, in fact, interact using high-level cues associated with body gestures, speech, face expressions and conversational prosody. For such reasons we have focused our work, specifically, on the pre-visualisation of synthetic human characters (avatars) able to perform body and face gestures in synchronisation with speech.

The implemented pre-visualisation of the avatar, though achieved with low complexity 3-D graphics, is nevertheless able to guarantee the animation flexibility and quality necessary to convey the required audio-video cues to the actors and let them synchronise their movements in the studio with a gesture of the avatars like a hand shake, a glance, a smile, or a kiss. The limited complexity of the graphics allows real-time animation of the avatars and has no impact on the final quality of the product since it can be replaced in post-production, as usual, by graphics and effects of arbitrary complexity.

We chose to implement a player capable of animating a human character starting from a script containing the

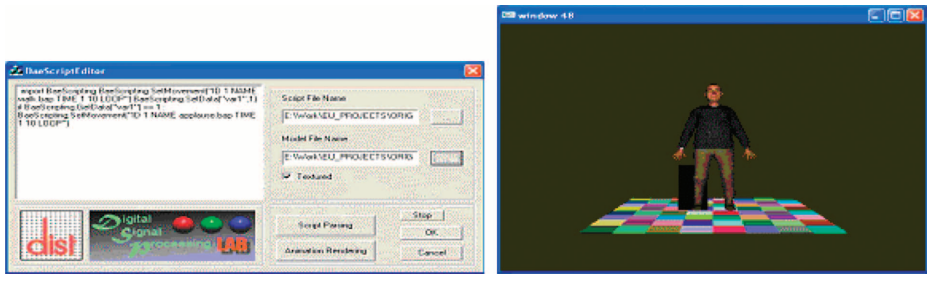


Fig. 13 *Graphical user interface of the movement compositor*

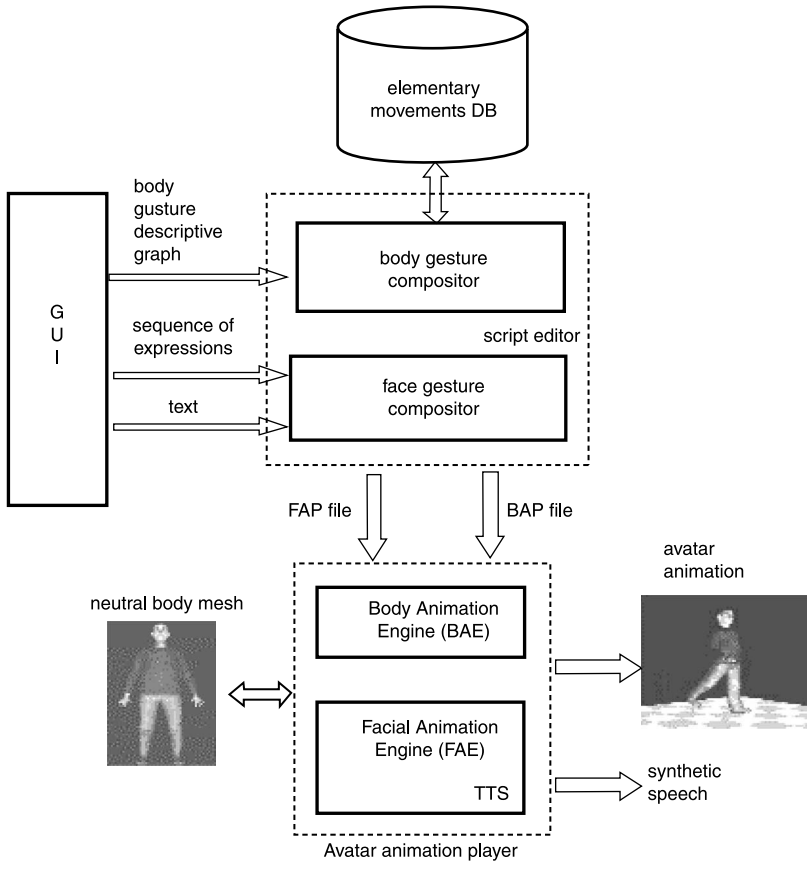


Fig. 14 *System architecture for the generation of avatar animations*

description of the gestures to perform together with synchronisation time stamps. The script is generated by a movement compositor through a friendly graphical user interface as shown in Fig. 13. The library of elementary gestures is made of motions such as 'crouch', 'raise an arm', 'walk', 'run', 'shake hand', 'turn around', 'turn' (left/right), etc.

The technology of the animation player is MPEG-4 compliant [31] and its architecture, as shown in Fig. 14, is based on two major blocks being the facial animation engine (FAE) and the body animation engine (BAE).

The composition of body gestures is obtained by parameterising and concatenating [32] a sequence of movements, selected from a library of elementary gestures acquired through a 3-D full-body motion capturing system, starting from an initial neutral body configuration representing the virtual actor in a pre-defined position and pre-defined posture. The elementary movements that must be concatenated are supplied to the system in the exact temporal order they must be synthesised on the avatar with the option of triggering the activation of different action subsequences on external events that are typically in the hands of the director. Similarly, also the duration of a single elementary movement (for instance a walk) can be triggered.

The body gestures descriptive graph produced by the movement compositor is encoded in MPEG-4 body animation parameters (BAP) format that is interpreted by the BAE.

The composition of the facial gestures, as shown in Fig. 15, is achieved through a twofold mechanism able to specify a sequence of elementary expressions as in the case of body gestures or, alternatively, through a text input that is transformed both in speech output (through a text-to-speech synthesiser) and in speech-synchronised lip movements. There is finally the possibility to mix these two modalities by means of expression tags that can be interleaved with the text: in this way facial expressions are reproduced on the avatar in synchronisation with the speech synthesis of the exact text segment they have been interleaved with.

Facial gestures are encoded in MPEG-4 facial animation parameters (FAP) format that is interpreted by the FAE [33, 34].

The full-body model used in the project is based on a 3-D mesh subdivided into two smoothly connected submasks, the former for the head, with around 2 000 polygons, is animated by the FAE and the latter for the body, with around 16 000 polygons, is animated by the BAE.

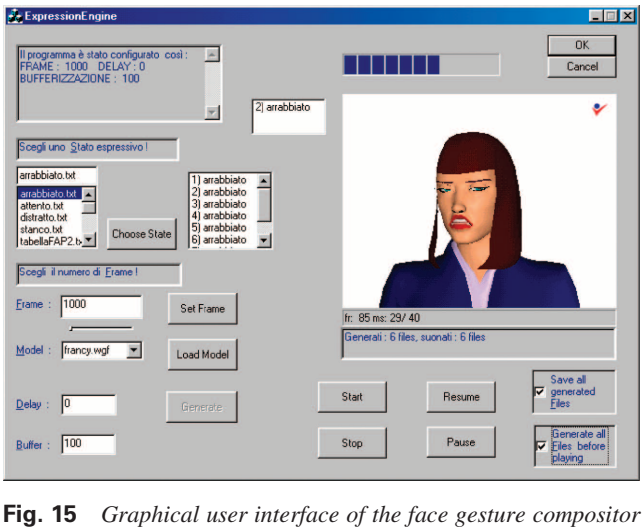


Fig. 15 *Graphical user interface of the face gesture compositor*

4.4 Action feedback module

For the director or a camera operator a renderer is provided that gives a pre-visualisation of the final composited scene, i.e. the virtual and real scene elements. This renderer receives updates from the virtual scene server and grabs the latest 3-D-shape model of the actors. It can then generate a 3-D representation of the scene and allows the director to view the scene from any position. This position can be dynamically updated to allow simulation of shots where the camera is moving. In order to give a more realistic image the 3-D-shape model of the actor is textured with a view from one of the cameras. Therefore the renderer determines the studio camera that has the smallest angle to the virtual camera. The 3-D shape model is stamped with the time-code of the alpha masks used to generate it, so the renderer requests the image from that time-code from the relevant capturing server, and uses it to texture the 3-D shape model.

The rendering engines in the visualisation server for the projectors request the latest head position from the head tracker and use this to calculate and render the projected image from the point of view of the actor. The view-dependant rendering allows the actor, if required, to keep looking at the face of the virtual character as he walks around him. That means the virtual scene components appear in space and the actor is immersed, as in a CAVE system developed for virtual reality applications [4]. The technical challenge in our system was to integrate the view-dependent projection into a production environment without interference with the chroma-keying facility used for the 3-D shape generation.

The integration was achieved by using an active chroma-keying based on a special retro-reflective cloth in conjunction with cameras equipped with a ring of monochrome (blue) LEDs. This configuration allows the projection of images onto the cloth giving visual feedback to the actor and still provides a robust chroma-key. Another problem that was addressed is that there should not be any light projected onto the actor, since this would be visible in the camera images. For this purpose a special mask is rendered over the projected image. This mask is generated from the 3-D surface model. The most recently computed surface model of the actor is therefore placed into the virtual scene and rendered completely in black with the z-buffer of the rendering system disabled. This guarantees that from the viewpoint of the projector all light rays that could fall onto the actor surface are masked. Owing to the latency of the system, light may still fall on the edge of a moving actor, particularly during fast movement. Therefore the mask is enlarged by a security factor that can be adapted to the latency and the fastest motion of the actor that is expected.

The renderer also receives any updates in the scene from the virtual scene server. If the virtual character were to move then the actor would see these movements. The combination of the scene updates and the viewpoint-dependent rendering thus allow complex interaction between the virtual and the real scene elements.

5 Results

The techniques discussed in this paper have been tested in an experimental production. For this purpose 3-D models of the entrance and the main hall of the Natural History Museum in London were produced, as depicted in Fig. 11. These have been used for planning and as background models in several scenes of a short demonstration video.

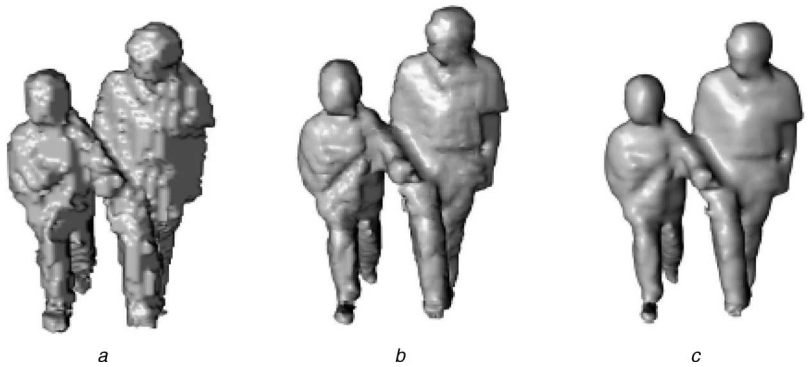
The foreground action was recorded in the experimental studio of BBC R&D using the techniques described in Section 4. Figure 16 shows the set-up in the studio that used four projectors to project a view-dependent pre-visualisation of a flying pterosaur, animated by another project partner (Framestore CFC). In Fig. 17 two frames of that scene from the final demonstration video are shown. Owing to the projection system the boy was able in all takes to keep his eye-line exactly towards the pterosaur flying virtually through the entrance hall. The director of the production could later select the best take based purely on artistic reasons.

Figure 18 shows a reconstruction using the octree reconstruction with super-sampling and Gaussian smoothing, as described in Section 4.2. The models contain approximately 5000 triangles. The resulting models were then textured and imported into a standard animation package (Softimage|XSI) to render the final image quality, including shadows, as depicted in Fig. 19. Further information and a video is available for download at [35].

As far as object modelling is concerned, we followed the workflow of Fig. 2 in order to develop a high-quality model of a Neanderthal skull for the demo video production of

Fig. 16 *The flight of a pterosaur was projected onto the studio walls using four projectors*

Fig. 17 *Two frames from the final video with inserted high-quality image of the flying pterosaur*



a *b* *c*

Fig. 18 *3-D reconstruction from 12 cameras using octree 128 × 128 × 128 resolution, result using super-sampling and additional Gaussian smoothing*

a Octree resolution
b Super-sampling
c Additional Gaussion smoothing

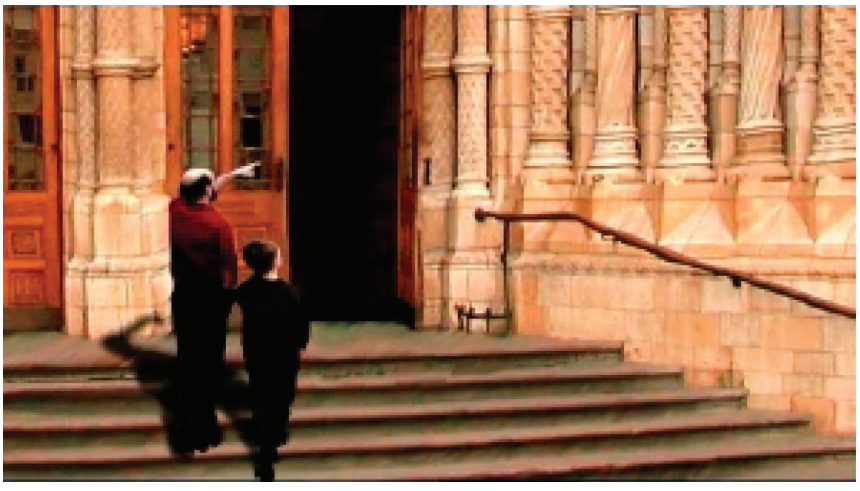


Fig. 19 *Resulting frame of a final quality video sequence with 3-D model integrated into virtual background*

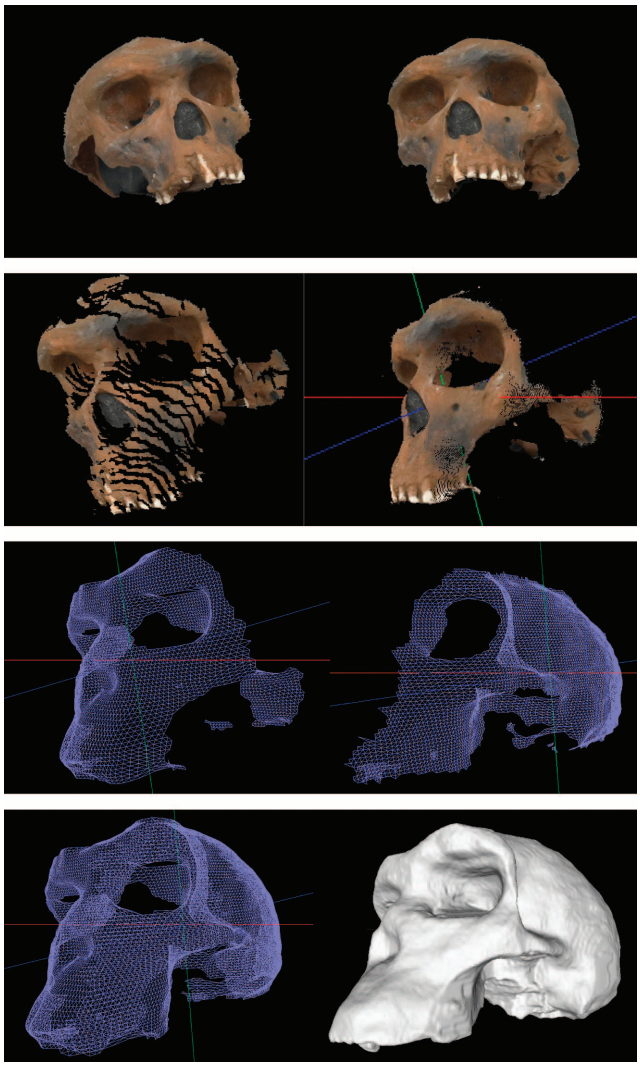


Fig. 20 *Development of a high-quality model of a Neanderthal skull*

In lexicographic order: two of the original views of the object; one of the depth maps of the object obtained with graph cut method; depth map smoothed through depth dithering; two triangle meshes corresponding to different partial reconstructions; combinations of two of the partial meshes; final complete 3-D model after fast levelset wrapping (64 million voxels)

the ORIGAMI project. Several depth maps were computed (see Fig. 20) using a graph-cutting method (each depth map was computed in just a few seconds). We then zippered and fused together such models using the levelset method described in Section 2. The front evolution was completed in just over 2 minutes on a 64Mvoxel dataset using a PC equipped with 3 GHz P4 processor and 1 GB of RAM, running Windows™ 2000. The final model after multi-view texture mapping is shown in Fig. 21 and in Fig. 22 (composited in the final scene).

6 Conclusions

In this article we provided a description of the goals and achievements of the ORIGAMI project, and we described the developed technologies for the pre-production phase, namely for the modelling of static objects and background. The approaches discussed in this contribution provide image-based modelling techniques based on one camera or a rig of four cameras that are moved freely. New techniques for camera calibration and 3-D reconstruction that have addressed the specific project requirements have been developed. The developed studio system provides an avatar system, a modelling system for dynamic actors and an action feedback component for real-time pre-visualisation.

The components have been used in an experimental production, that involved two post-production houses (Framestore CFC, London, and Chinatown, Milan) and the BBC. In particular the action feedback component has shown a high potential for improving the efficiency of the production flow, since the creative production crew on-set gets an immediate feedback. Therefore typical production problems, like wrong positions of real and virtual scene objects and wrong actor eye-lines, can be avoided.

Furthermore, the fact that all major scene elements are captured in 3-D open new creative ways for future productions that were also demonstrated in the experimental production, the final camera perspective can be decided in post-production. That means the director has the means to correct the camera angles in a later phase. Moreover physically impossible or expensive camera shots, like crane shots can be simulated in an animation package.

The concept also allows the change of the scene lighting. In particular the object modelling technique has addressed this issue. The background models are currently not intended for scene re-illumination. The dynamic actor models allow only a moderate variation of the scene illumination. Current work is addressing a better radiometric modelling of actor models that allows the

Fig. 21 *Two views of the final 3-D model of the Neanderthal skull used in the demo video of the ORIGAMI project*

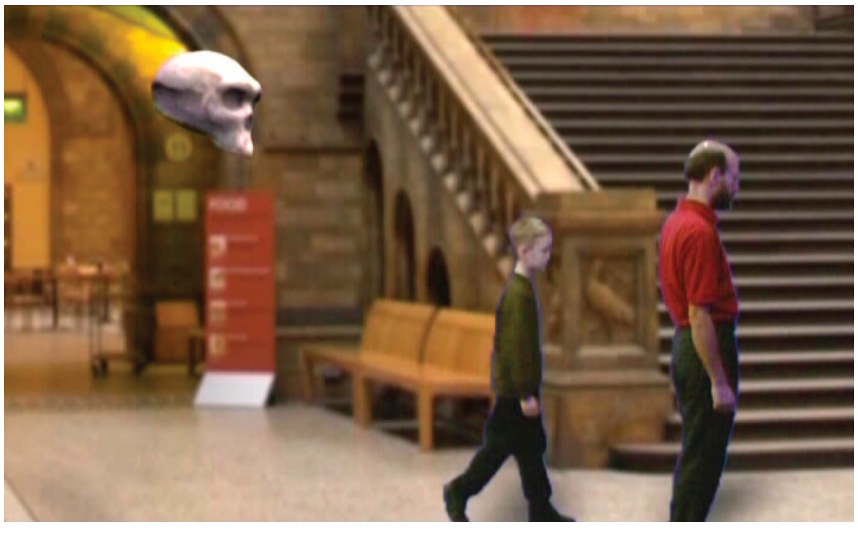


Fig. 22 *One frame of the 'floating skull' model in the ORIGAMI demo video production*

neutralisation of the studio illumination and to capture the surface reflectivity of the actors.

7 References

1 IST-2000-28436: 'ORIGAMI: A new paradigm for high-quality mixing of real and virtual', Information Society Technologies (IST) Prog. – V Framework Programme, http://www-dsp.elet.polimi.it/origami/
2 Rosenthal, S., Griffin, D., and Sanders, M.: 'Real-time computer graphics for on-set visualization: 'A.I.' and 'The Mummy Returns'. Conf. Proc. Sketches and Applications, Siggraph, 2001
3 Tzidon, *et al.*: 'Prompting guide for chroma keying', March 1996, United States Patent, 5,886,747
4 Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., and Hart, J.C.: 'The CAVE: audio visual experience automatic virtual environment', *Commun. ACM*, 1992, **35**, pp. 67–72
5 Grau, O., Pullen, T., and Thomas, G.A.: 'A combined studio production system for 3-D capturing of live action and immersive actor feedback', *IEEE Trans. Circuits Syst. Video Technol.*, 2004, **14**, (3), pp. 370–380
6 Grau, O.: '3D sequence generation from multiple cameras'. IEEE Int. Workshop Multimedia Signal Process., Siena, Italy, Sept. 2004
7 Beardsley, P., Torr, P., and Zisserman, A.: '3d model acquisition from extended image sequences', ECCV, number 1064 in LNCS, Springer, 1996, pp. 683–695
8 Hartley, R.: 'Estimation of relative camera positions for uncalibrated cameras'. ECCV, 1992, pp. 579–587
9 Koch, R., Pollefeys, M., Heigl, B., Van Gool, L., and Niemann, H.: 'Calibration of hand-held camera sequences for plenoptic modelling'. ICCV, Korfu, Greece, Sept. 1999
10 Pollefeys, M., Koch, R., and Van Gool, L.J.: 'Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters', *Int. J. Comput. Vis.*, 1999, **32**, (1), pp. 7–25
11 Hingorani, S.L., Cox, I.J., and Rao, S.B.: 'A maximum likelihood stereo algorithm', *Comput. Vis. and Image Underst.*, 1996, **63**, (3), pp. 542–567
12 Scharstein, D., Szeliski, R., and Zabih, R.: 'A taxonomy and evaluation of dense two-frame stereo correspondence algorithms'. IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, HI, USA, December 2001
13 Kolmogorov, V., and Zabih, R.: 'Multi-camera scene reconstruction via graph cuts'. ECCV, 2002, Vol. 3, pp. 82–96
14 Woetzel, J., and Koch, R.: 'Real-time multi-stereo depth estimation on GPU with approximative discontinuity handling'. 1st Eur. Conf. Vis. Media Production (CVMP), London, UK, March 2004
15 Woetzel, J., and Koch, R.: 'Multi-camera real-time depth estimation with discontinuity handling on PC graphics hardware'. 17th Int. Conf. Pattern Recognit. (ICPR), Cambridge, United Kingdom, August 2004
16 Yang, R., and Pollefeys, M.: 'Multi-resolution real-time stereo on commodity graphics hardware'. Conf. Comput. Vis. Pattern Recognit. (CVPR), Madison, Wisconsin, USA, June 2003
17 Falkenhagen, L.: 'Hierarchical block-based disparity estimation considering neighbourhood constraints'. Int. Workshop on SNHC and 3D Imaging, Rhodes, Greece, 5–9 September 1997
18 Koch, R., Pollefeys, M., and Van Gool, L.: 'Multi viewpoint stereo from uncalibrated video sequences'. Proc. ECCV, No. 1406 in LNCS, Springer-Verlag, Freiburg, 1998
19 Pedersini, F., Piccarreta, L., Sarti, A., and Tubaro, S.: 'Estimation of radiometric parameters for a realistic rendering of 3D models'. Int. Conf. on Image Processing (ICIP), Kobe, Japan, October 1999, Vol. 4, pp. 376–380
20 Jebara, T., Azarbayejani, A., and Pentland, A.: '3D structure from 2D motion', *IEEE Signal Process. Mag.*, 1998, **16**, (3), pp. 66–84
21 Soatto, S., Frezza, R., and Perona, P.: 'Motion estimation via dynamic vision', *IEEE Trans. Autom. Control*, 1996, **41**, (3), pp. 393–413
22 Dell'Acqua, A., Sarti, A., and Tubaro, S.: 'Effective analysis of image sequences for 3D camera motion estimation'. Proc. Int. Conf. on Augmented Virtual Environments and 3D Imaging (ICAV3D), Mykonos, Greece, 30 May–01 June 2001, pp. 307–310
23 Hartley, H., and Zissermann, A.: 'Multiple view geometry in computer vision' (Cambridge University Press, 2000)
24 Sethian, J.A.: 'Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry', *in* 'Fluid Mechanics, Computer Vision, and Materials Science' (Cambridge University Press, 1999)
25 Faugeras, O., and Keriven, R.: 'Variational principles, surface evolution, PDE's, level set a methods, and the stereo problem', *IEEE Trans. Image Process.*, 1998, **7**, (3)
26 Sarti, A., and Tubaro, S.: 'Image-based multiresolution implicit object modeling', *EURASP J. Appl. Signal Process.*, 2002, **2002**, (10), pp. 1053–1066
27 Marcon, M., Piccarreta, L., Sarti, A., and Tubaro, S.: 'A fast level-set approach to 2D and 3D reconstruction from unorganized sample

points'. 3rd Int. Symp. Image and Signal Process. Anal. (ISPA), Rome, Italy, 18–20 September 2003
28 Giblin, P., and Kimia, B.B.: 'A formal classification of 3D medial axis points and their local geometry'. IEEE Conf. Comput. Vis. Pattern Recognit., 2000, Vol. 1, pp. 566–573
29 Koch, R., Frahm, J.F., Evers-Senne, J.-F., and Woetzel, J.: 'Plenoptic modelling of 3d scenes with a sensor-augmented multi-camera rig'. Tyrrhenian Int. Workshop on Digital Communication (IWDC): proceedings, Sept. 2002
30 Thomas, G.A., Jin, J., Niblett, T., and Urquhart, C.: 'A versatile camera position measurement system for virtual reality TV production'. Conf. Proc. IBC, 1997
31 Preda, M., and Preteux, F.: 'MPEG-4 human virtual body animation', in Bourges-Sevenier, M. (Ed.): 'MPEG-4 jump-start' (Prentice Hall, Upper Saddle River, NJ, January 2002)
32 Bartels, R.H., Beatty, J.C., and Barsky, B.A.: 'An introduction to splines for use in computer graphics and geometric modeling' (Morgan Kaufmann, Los Altos, CA, 1987)
33 Lavagetto, F., and Pockaj, R.: 'The face animation engine: towards a high-level interface for the design of MPEG-4 compliant animated faces', *IEEE Trans. Circuits and Syst. Video Technol.*, 1999, **9**, (2), pp. 277–289
34 Lavagetto, F., and Pockaj, R.: 'The facial animation engine (FAE)', in Pandzic, I.S., and Forchheimer, R. (Eds.): 'MPEG-4 facial animation' (John Wiley & Sons Ltd, UK, 2002), pp. 81–101
35 , http://www.bbc.co.uk/rd/projects/virtual/origami
36 Azarbayejani, A., and Pentland, A.P.: 'Recursive estimation of motion, structure, and focal length', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995, **17**, (6), pp. 562–575
37 Jin, H., Favaro, P., and Soatto, S.: 'Real-time 3-D motion and structure from point features: a front-end system for vision-based control and interaction'. Proc. IEEE Int. Conf. on Comput. Vis. Pattern Recognit., June 2000
38 Chiuso, A., Favaro, P., Jin, H., and Soatto, S.: '3-D motion and structure causally integrated over time part II: implementation'. Proc. Eur. Conf. Comput. Vis., June 2000