Machine Vision and Applications (2005) 16(4): 236–245
DOI 10.1007/s00138-005-0177-6

ORIGINAL PAPER

Tomás Rodríguez · Ian Reid · Radu Horaud · Navneet Dalal · Marcelo Goetz

Image interpolation for virtual sports scenarios

Received: 25 March 2004 / Accepted: 18 February 2005 / Published online: 10 June 2005

Abstract View interpolation has been explored in the scientific community as a means to avoid the complexity of full 3D in the construction of photo-realistic interactive scenarios. EVENTS project attempts to apply state of the art view interpolation to the field of professional sports. The aim is to populate a wide scenario such as a stadium with a number of cameras and, via computer vision, to produce photo-realistic moving or static images from virtual viewpoints, i.e where there is no physical camera.

EVENTS proposes an innovative view interpolation scheme based on the Joint View Triangulation algorithm developed by the project participants. Joint View Triangulation is combined within the EVENTS framework with new initiatives in the field of multiple view layered representation, automatic seed matching, image-based rendering, tracking occluding layers and constrained scene analysis. The computer vision software has been implemented on top of a novel high performance computing platform with the aim to achieve real-time interpolation.

Keywords Image interpolation · Morphing · Layered segmentation · Joint view triangulation · Real time virtual scenarios

1 Introduction

One simple but compelling idea to have emerged in recent times is the notion of a Panorama or Image Mosaic. The general underlying concept is based on a set of overlapped shots taken around a central point using a photo camera, which are spliced or blended together by means of projective geometric transformations. This pervasive idea can now be found in devices such as IPIX or de-facto standards such as QuickTime VR. Current software for panoramas typically runs off-line, is based on a single camera with unchanging intrinsic parameters, viewing a static scene, and uses very simple camera motions (restricted to pure rotation at a single viewpoint) to ensure that image overlap can be described by a simple transformation.

However, as this is the crux of the need for innovation, traditional panorama technology, being based on a single viewpoint, is fundamentally inadequate for the scenarios foreseen in the EVENTS project in which we aim to solve much more complex problems of multiple viewpoints and dynamic scenes. For multiple viewpoints there no longer exist the simple geometric transformations between images that characterize the panoramic approach and new methods are required.

EVENTS is a project sponsored by the European Commission's IST programme whose objective is to populate a wide scenario such as a stadium with a number of cameras observing a specific part of the scene such as a goal-mouth, and via software implementations of computer vision algorithms, to produce photo-realistic views from viewpoints where there is no physical camera. The system will enable the user to select a virtual viewpoint somewhere in between the physical cameras and to generate a novel view from there, or more impressively, command the software to generate a sequence as if a camera had swept through the full range covered by the cameras, smoothly and realistically interpolating intermediate virtual views. EVENTS system is able to process in real-time input images from multiple cameras and compute a dynamic synthetic view of the scenario (i.e. sort of virtual video); empowering the viewer to select his view position in time and space within the range covered by the cameras.

The difficulty associated with creating a novel view of a scene is principally related to the fact that the relative location of scene features is directly dependent on their depth. Thus, for realistic novel view creation one needs to

T. Rodríguez (✉)
Eptron SA. R&D Department Madrid, Spain
E-mail: tomasrod@eptron.es

I. Reid
Oxford University, Department of Engineering Science, UK

R. Horaud · N. Dalal
INRIA Rhône-Alpes, Montbonnot, France

M. Goetz
C-Lab, Paderborn, Germany

compute, either implicitly or explicitly, the depth of scene points. In addition, the following problems must be addressed: how best to match features between views, how to represent the implicit or explicit scene structure, how to render parts of the scene not seen in all cameras, how to deal with occluding boundaries to avoid joining foreground and background, how to deal with natural scene constraints, how to manage independent motion in the scene and finally, how to compute all of the above efficiently. There is no generally accepted solution to these questions in the scientific community, hence the need for innovation arises. Techniques for achieving novel views [1–3] can be broadly categorized as follows:

1. Computing full 3D structure: the most direct approach is to determine camera locations via calibration (possible via self-calibration), to match points between images, and hence to triangulate the depth of every point in the scene; i.e. exhaustively recover the scene structure. If this scene structure were then texture mapped, realistic views from any viewpoint could potentially be created. As well as being computationally intensive and potentially error-prone, this technique does not address the issue of how to render parts of the scene where no structure is available, either because of errors, or because no match exists, such as in occluded regions.
2. Morphing: at the other extreme, involving no structure computation, whatsoever is the technique of 'morphing' developed within the graphics community (in one sense, mosaicing can be thought of as morphing in which the transformation is known a priori to be a homography). Applying morphing techniques to the 3D scenarios envisioned in EVENTS leads to nonsensical intermediate images in which spatial features are at best blurred and do not exhibit realistic parallax. Within the framework of morphing technologies, EVENTS introduced a new paradigm of structure based view morphing, called Joint View Triangulation (JVT) as described in [4–8]. This triangulation technique creates a dense map of point correspondences between views, and piece-wise smooth assumption is used to create triangle matches between two images.
3. Related to 3D reconstruction techniques, but not involving explicit 3D structure, is the notion of point transfer via the so-called multifocal tensors. Here dense disparity matches, together with geometric relationships between images, are used to create physically valid novel views. The idea of using point transfer for visual tracking and novel view generation within EVENTS has been evaluated in [9, 10].

   Image-based rendering: Various ideas from the so-called image-based rendering can achieve photo-realistic results, but suffer from a number of significant drawbacks with respect to EVENTS' goals. The so-called lumigraph [11], or light-field was developed as a representation of a scene, which enables relatively easy generation of novel views with photorealistic results. However, its computational and storage requirements limit its application in dynamic scenes. An alternative, which has become popular in recent years is based on the observation that a point in a novel view should be rendered in a colour that is consistent with all the physical views. This 'naive' statement hides the technical detail of what it means to be consistent, but the eventual algorithm implicitly computes a depth for every point in the scene by looking for a match, which is consistent in a large number of views. Generally such methods are slow, requiring orders of minutes or even hours for a single view.

   In EVENTS, we initially evaluated this technique [9] with the conclusion that its effectiveness is crucially dependent on having a large number of views, which are very accurately calibrated (close to the idea of 'space-carving'). Our experiments also revealed that for football scenarios, where there are large areas of uniform colour/texture, a technique based on extracting the consistent colour at every point is highly ambiguous and error-prone. However, the project has since developed a novel version of the algorithm that performs better on small numbers of real views (as few as 2 or 3), and which is considerably faster. Though being too slow for direct use in EVENTS, it is likely to be useful for improving the quality of rendering the 'static' background.

By extending and combining some of the above mentioned research paths, EVENTS was expected to improve over other existing initiatives like Kewazinga Corp.'s SwingView and Eyevision, that currently exploit hardware based view interpolation to create *'Matrix'* like effects for professional applications. In contrast, EVENTS is a software only solution that offers better image quality, wider viewing angles and real-time interpolation as its most attractive features. This paper describes the different elements composing the EVENTS system. We start out in Sect. 2 with an introduction to the computer vision algorithms used in the project. Section 3 follows with a description of the EVENTS' platform. The evaluation of the results obtained and comparison with previous experiences are presented in Sect. 4. We end in Sect. 5 with the conclusions and proposals for further improvement.

2 Computer vision software

In EVENTS, we have developed a new image interpolation method that automatically synthesizes new views from existing ones, rapidly, robustly, and realistically. The method first started with a so-called 'joint view triangulation' (JVT) [12, 13] for the complete image; later, it was improved by applying JVT independently to motion-segmented background and foreground; finally it evolved to a fully layered JVT interpolation.

The first step of the computer vision chain described in Fig. 1 consists of automatic matching of corresponding points between views for sufficiently textured images, leading to a quasi-dense matching map. Next, a generalised

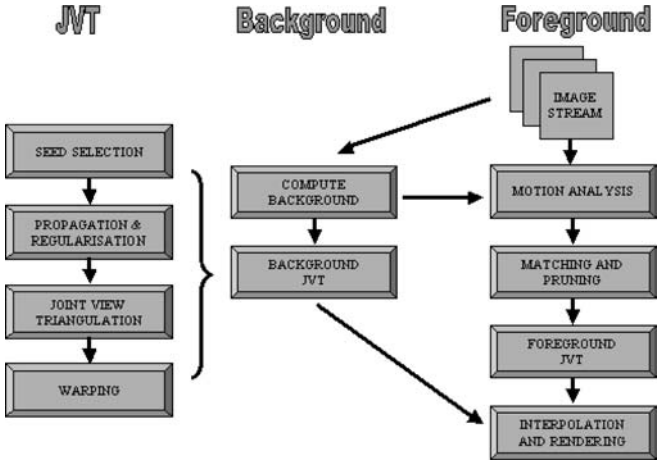

Fig. 1 The Computer vision chain. Background and foreground layers are computed separately and then rendered together

motion-segmentation method segments the scene into layers; one of them being the static background. Subsequently, JVT is computed independently for background and foreground layers and a robust algorithm is invoked for separating matched from unmatched areas and handling of the partially occluded areas. Finally, we developed an original algorithm for image warping, which renders background followed by foreground JVTs.

2.1 Joint view triangulation

Since image interpolation relies exclusively on image content with no depth information, it is sensitive to changes in visibility. Therefore, to handle the visibility issue, we proposed a multiple view representation called Joint View Triangulation (JVT), which triangulates simultaneously and consistently two images without any 3D input data. JVT combines the best of the structure-based approach by computing dense disparity matches but bypassing the explicit construction of a 3D model. Instead an intermediate JVT representation captures the scene structure and a novel-rendering algorithm creates a novel intermediate view. The JVT is composed of two main steps: quasi-dense disparity map construction and merging and triangulation.

2.1.1 Quasi-Dense disparity map construction

An initial pre-processing step is applied with the objective to: remove black boundaries, reduce resolution, detect points of interest and correlate between images using reduced maximum disparity (1/10) due to repetitive textures. The method starts from extracting points of interest, which have the highest textureness, from two original images using a *Harris* corner detector. Then we use a zero-mean normalised cross-correlation (ZNCC) to match the points of interest across two images. This gives the initial list of point correspondences, or matches, sorted by the correlation score.

Next, the points of interest are used as seeds to propagate the matches in its neighbourhood from most textured (therefore most reliable) pixels to less textured ones, leading to a quasi-dense matching map. At every step, the seed match with the best ZNCC is removed from the list. Then, for each point in a 5×5 neighbourhood window centred at the seed match in the first image, we use again ZNCC to construct a list of tentative match candidates in a 3×3 window in the neighbourhood of its corresponding location in the second image. Successful matches (i.e. when the ZNCC score is greater than some threshold) are added simultaneously to the match list and the seed list while preserving the unicity constraint. The process is repeated until the seed list becomes empty.

2.1.2 Merging and triangulation

The brute quasi-dense matching disparity map may still be corrupted and irregular. Although the unconstrained view-interpolation approach makes no rigidity assumptions about the scenes, we also assume that the scene surface is at least piece-wise smooth. Therefore, we will use local geometric constraints encoded by planar homography to regularise the quasi-dense matching by locally fitting planar patches in both the images. The construction of the matched planar patches starts by partitioning the first image into regular square patches of different scales. For each square patch, we obtain all matched points within the square from the quasi-dense matching map and try to fit the corresponding matched patch in the second image. A plane homography H between the two images is tentatively fitted to the matched points to look for potential planar patches. Because a textured patch is rarely a perfect planar facet, the putative homography for a patch cannot be estimated by standard least squares estimators. Hence, a robust RANSAC method have been adopted, which provide a reliable estimate of the homography even if some of the matched points of the square patch are not actually lying on the common plane. For each RANSAC trial, four matches are selected in the neighbourhood of the four corners of the square patch in the first image (see Fig. 2). Then we count the number of matches in the square compatible with the tentative homography defined by the selected four matches. RANSAC trials are continued until we find the best homography, i.e. the one which maximises the number of inliners.

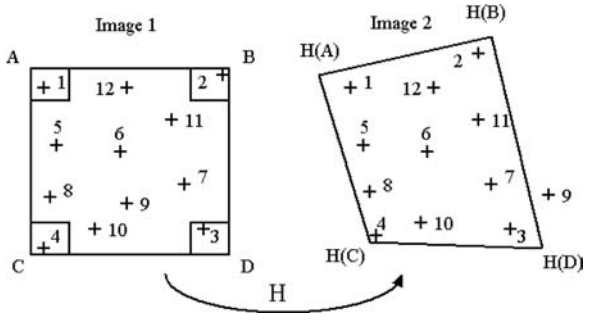

Fig. 2 Matches 1, 2, 3 and 4 have been selected by the RANSAC trial and define the optimum homography, which maps the square patch in image 1 to the distorted patch in image 2

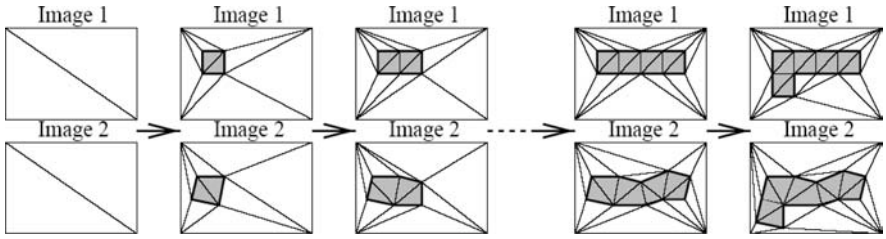

Fig. 3 Every square patch is divided in two triangles. Then, triangulation is incrementally constructed starting from two triangles in each image. Consistent matched triangles (gray) are added incrementally in the triangulation row by row from top to bottom. The set of matched triangles grow simultaneously in both images in a coherent way

This process of fitting the square patch to a homography is first repeated for all the square patches of the first image from the larger to the smaller patches. Since the resulting matched patches are not exactly adjacent in the second image, we next perturb the vertex locations to eliminate small discontinuities and overlaps between the patches.

The previous steps produce a globally incoherent set of matched patches because of intersections. The merging step selects one of these and converts it to an incomplete but coherent JVT (i.e. a one to one correspondence between all vertices and contours). A JVT is then grown simultaneously by successive merging of the previous matched patches in the two images.

The algorithm (Fig. 3), described in detail in [6, 8, 13, 14], can be summarised as follows:

1. The JVT starts from two triangles in each image.
2. Starting from left to right, top to down in the first image, each matched planar patch is divided in two triangles and incrementally inserted into each triangulation if the coordinates of its three point match(es) in the two images are consistent with the current structure. A new point match is consistent if its two points are corresponding vertices of a match of the contour, or are both outside the set of matched triangles and must also verify that the resulting constrained edges would not intersect a current contour in either image. Otherwise, the triangle is left unmatched.
3. The structure is improved in a last step by further checking if the remaining unmatched triangles could be fitted to an affine transformation. If an unmatched triangle succeeds in fitting an affine transformation, its label is changed from unmatched to matched.

After computing the JVT (Fig. 4), a robust algorithm is invoked for separating matched from unmatched areas and handling partial occlusions.

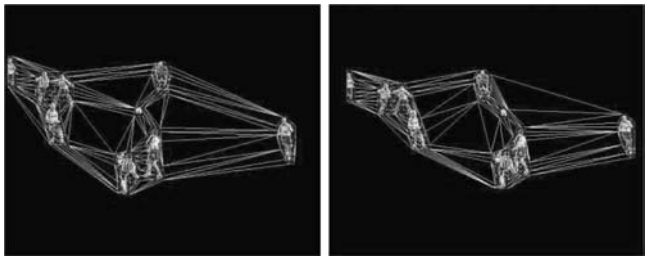

Fig. 4 Example of a JVT for the left and right images

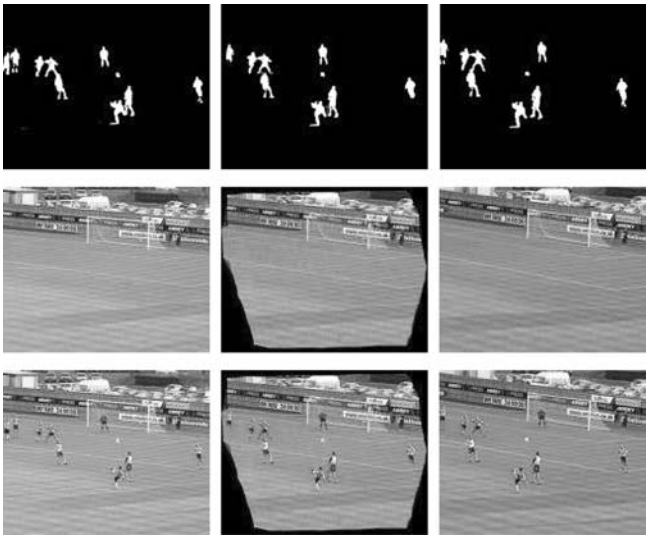

Fig. 5 *Left to Right*: Original Left, Interpolated in between, Original *Right*. *Up to down*: Segmented, Background, Rendered results

2.2 Motion segmentation

Representing a moving scene as a static background and a moving foreground (Fig. 5) is a powerful, though by no means a novel idea. Within the context of EVENTS, such a segmentation has two principal benefits: it helps avoid the danger of foreground objects being 'joined' incorrectly to the background in the novel view (common in triangulation based methods such as JVT and Taylor's), or foreground regions being obliterated by unmatched regions, by explicitly rendering a background JVT followed by a foreground. Also, since JVT for the background is unchanged over time, considerable savings can be made by only computing foreground triangulation for each frame. More generally, JVT rendering is prone to errors in regions where there are occluding boundaries in the scene, and this applies both to moving and static regions. Furthermore, the same problem (albeit on a smaller scale) is still exhibited when two moving foreground regions occlude one another (as often happens in a crowded penalty area).

Our approach was to generalise the foreground/ background method to a layered segmentation and tracking of non-rigid objects in one or more views using a probabilistic model of object occupancy and visibility in order to reason about object locations and occlusions. The method is most significantly distinguished from previous layered models [15, 16] by its use of multiple views across time. Work in layered video has mostly concentrated on a single camera and independently moving objects; i.e. temporal segmentation. In other less proliferous work there have been attempts to achieve spatial segmentation into layers (typically for stereo or moving cameras). In EVENTS we tackled both. Although other methods have variously used appearance models, motion models, new proposals for layers, etc, our method uniquely combines all of these and could further support shape models.

We developed a method to segment automatically a pair of densely matched views into layers based on compliance with a homographic model of transfer. Our basic premise was that the scene can be represented as a collection of $n+1$ independent depth ordered layers (homographies for the background, homographies or affinities for each foreground object), which in general, may overlap and therefore occlude each other. According to this model, the value of an observed image pixel is generated by the foremost or visible layer at that point.

The layer model (Fig. 6) at time t is denoted as $L_t = (L_t^0, L_t^1 \ldots L_t^n)$, where $L_t^i = (O_t^i, A_t^i, \Phi_t^i)$ are the parameters of the ith layer. Occupancy (O_t^i) and Appearance (A_t^i) are computed from the input images, while Alignment $(\Phi_t^i = \{\phi_t^{ij}\}$, one per view) encode the transformation relating the coordinate frame of the layer to each view.

An observed image I_t^j in the jth view is generated at time t from a mixture distribution, where the value of each pixel x is sampled according to the realisation of a random variable described by the appearance model of the foremost layer at x:

$$P\left(I_t^j(x)\right) = \sum_{i=0}^{n} P\left(I_t^j(x) \middle| V_t^j(x) = i\right) P\left(V_t^j(x) = i\right) \quad (1)$$

The foremost layer is obtained with the aid of the view dependent Visibility indicator V_t^j and the probability that a particular layer i is visible in the jth view can be computed in terms of the occupancy of all layers:

$$P\left(V_t^j(x) = i\right) = O_t^i\left(\phi_t^{ij^{-1}} x\right) \prod_{k=i+1}^{n} \left[1 - O_t^k\left(\phi_t^{kj^{-1}} x\right)\right] \quad (2)$$

On the other hand, the observed intensity is assumed to be distributed normally, conditioned on the Visibility and has mean given by the aligned Appearance map:

$$P\left(I_t^j(x) \mid V_t^j(x) = i\right) \sim N\left(A_t^i(\phi_t^{ij^{-1}} x), \sigma_I^2\right) \quad (3)$$

New views can be generated simply by rendering each layer appropriately, and the fact that visibilities are probabilistically represented yields beneficial smoothing at boundaries. If we know which layer is visible at each pixel, then our problem is partitioned into $n+1$ sub-problems. The method models visibilities as hidden variables and uses an EM-algorithm to solve the problem. We wish to compute the parameters of the layered model that maximise the posterior likelihood of current image pair, given the previous layered parameters using a Bayesian network that takes the form of a hidden Markov model. The joint probability of our Bayesian net can be factored as:

$$P(L_0) \prod_{\tau=i}^{t} P(I_\tau \mid L_\tau) P(L_\tau \mid L_{\tau-1}), \quad (4)$$

where the function to maximise is: $F(L_t) = lnP(I_t \mid L_t) + P(L_t \mid L_{t-1})$. In our case, the E-step corresponds to:

$$Q^{(k)}(V) = P\left(V \mid I_t, L_t^{(k-1)}\right), \quad (5)$$

while the M-step is denoted by:

$$L^{(k)} = \underset{L_t}{\operatorname{argmax}} \sum_{V} Q^{(k)}(V) lnP(V, I_t \mid L_t) + lnP(L_t \mid L_{t-1}) \quad (6)$$

We partition the M-step into three parts, in which the alignment, occupancy and appearance for each layer are computed in sequence. (See Ref. [10] for implementation details).

At each time t a set of new views is obtained from the cameras, and pixel intensities/colours in each image are combined with a prediction from the previous frame (based on both a motion model and a measurement of the visual motion from a simple affine tracker); i.e. the use of a motion model enables consistent tracking even under extreme

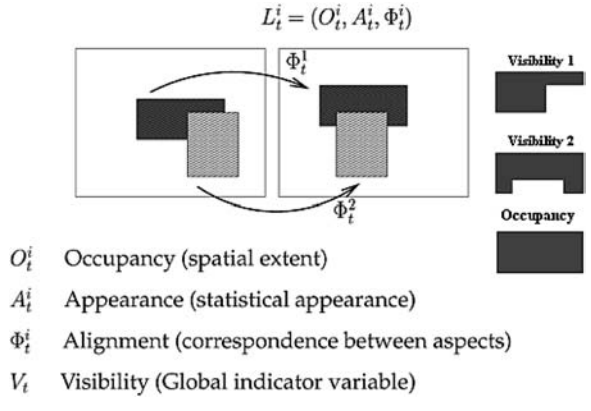

Fig. 6 Layered model parameters: O_t^i is a probabilistic map denoting the shape of the object; A_t^i is an intensity map representing the appearance of the pixels composing the object; Φ_t^i is a transformation relating the coordinate frame of the layer to each camera view; V_t^i is computed from the Occupancy of all layers using relation (2).

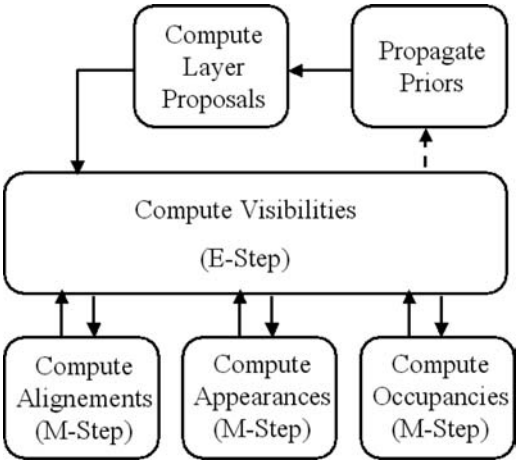

Fig. 7 The steps shown are performed as one cycle per frame. However, the EM steps may be iterated. We found that two or three iterations are sufficient

occlusion. The layer parameters are propagated from those computed at the previous time according to the mode of the prior distributions. This procedure acts much like a prediction and serves as the starting point of the EM-algorithm (see Fig. 7). The next stage is to reconsider the order of the model, i.e does the model explain the data well and if not should there be additional layers? New objects are initialised automatically when dense clusters of unexplained pixels of a given minimum size are found. The new layer is initialised by setting its occupancy to 0.8 inside the region and taking current image pixel values as appearance. Once new layer proposals have been dealt with, we can solve for the new parameters. A more detailed description of the implementation of the EM-algorithm can be found in [10, 17].

The background layer is composed using constrained analysis to account that various frames are coherent in time. Assuming the fixed field of view, we estimate the static background model and then compute the View Interpolation for this static background. We performed various experiments to estimate the reliability of our background estimations using a mosaicing technique. These experiments showed that given enough interest points are present on the static background, background can be estimated reliably. Robust statistical methods were employed to further improve our estimation by taking into account the information from all views simultaneously.

Tackling panning, tilting and zooming cameras is a natural extension to the described procedure. Here, the background does not appear as static anymore and it becomes necessary to estimate the camera motion parameters (pan, tilt, and zoom) in order to be able to characterize image regions corresponding either to the background or the foreground. Within this task, a mosaicing technique was implemented, which proceeds as follows: the image-to-image transformation parameterized by pan, tilt, and zoom is roughly estimated by localizing a few static points in the scene and tracking them through the image sequence. This transformation is then optimized such that textures coming from different images perfectly overlap. Based on three consecutive images, dynamic regions are detected. Finally a mosaic composed of static regions from several images is built.

2.3 View interpolation

In-between images are obtained by shape interpolation and texture bleeding of the two original images, which are the endpoints of the interpolation path. We have developed a simple OpenGL interface, which renders background JVTs in real-time and is compliant with visibility constraints previously defined (i.e. layers are rendered from down to top). This is done via geometric transfer using the trifocal tensor and a JVT of two views.

The background JVT is rendered using the pseudo painter's algorithm described in [12]. Here, the triangulation of both the original images is warped to the intermediate view and rendering starts with the unmatched, followed by the matched triangles. Finally a weighted blending of the textures is performed.

Our chosen approach [9] to foreground image-based rendering is summarised in Fig. 8. To choose a foreground colour to render at a particular point x in the novel view, we obtain in the nearby cameras the epipolar lines of the line of sight through x. The pixel colours along these lines are sampled and rectified into the canonical frame of one reference image from the input set. The rectification is equivalent to a trifocal transfer as described in [18]. In the rectified frame, colours are compared between images at each depth, to give a colour consistency measure, representing the likelihood of scene structure at this depth being the source of the rendered point. For this purpose, we use a simple 5×5 normalised intensity gradient texture classifier. To avoid loosing low frequency variations in images, we employ a scale space concept, where searches for texture consistency in the epipolar lines is done at multiple scale resolutions. Candidate depths are isolated first at low resolution and then propagated to higher resolution in a region around the chosen depth to refine the solution. Consequently, the cost of the search has been reduced, while the method incorporates global changes in the images that would otherwise be overlooked by the use of local operators only (Fig. 9). The algorithm is implemented using a *pyramid* in the input images and increases the efficiency of the scale space algorithm by generating multiple resolution novel views and retaining the likelihood maxima found at each pixel for propagation to the next layer. In practice, the algorithm renders coarse level pixels only when required to by the need to render nearby pixels at finer resolutions. See Refs. [9, 19] for implementation details.

3 EVENTS platform

EVENTS is composed of two main applications: the Image Interpolation Engine and the Interpolated Video Player. The

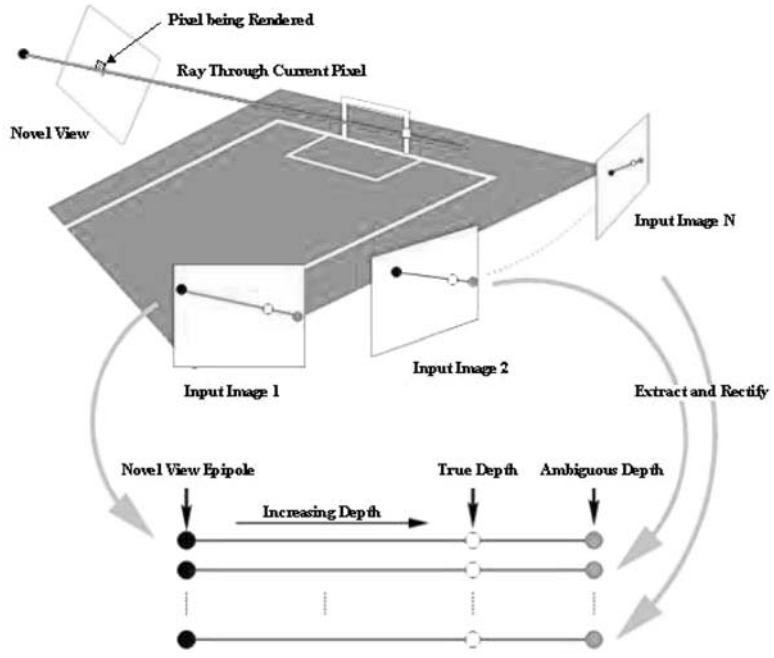

Fig. 8 Image-based rendering process. Epipolar lines in the real cameras are sampled and rectified. The novel viewpoint will be typically between the real cameras, but has been placed to one side for clarity

Fig. 9 Rendering an intermediate novel view from 2 cameras 13 m apart. Note that the reconstructed novel view preserves both high (pitch lines) and low (grass stripes) frequencies

first tool processes video data captured from multiple cameras and produces the stream of interpolated images. The second tool offers the viewer the possibility to change the view position and display the output of the Interpolation Engine, thus allowing independent production and playback of interpolated video material.

The user interface of the Image Interpolation Engine (Fig. 10) implements the following functions: aid the user to install and configure the interpolation engine, operate the underlying computer vision software and monitor in real time the interpolated images stream. Among others, the tool enables the user to: select and link cameras from the provided graphical interface, start/stop the interpolation engine, visualise input images from the different cameras configured, modify computer vision parameters and select manual seed matches. This last option is seldom required since automatic matching is sufficient in most cases.

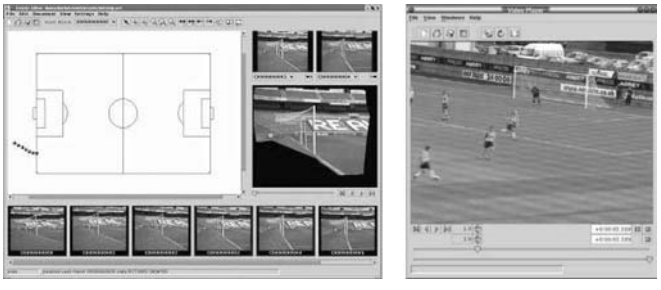

Fig. 10 Screen shots of EVENTS': **a** Image Interpolation Engine and **b** Interpolated Video Player

The Image Interpolation Engine platform has been implemented by means of a high performance computer-cluster with real-time communication support, based on the Scalable Coherent Interface (SCI) environment. In order to assure the required time constrains, a channel-oriented, component-based, real-time communication middleware, with bandwidth reservation support, has been developed on top of an SCI-cluster.

The Player is a downloadable Java application (Fig. 10) aimed to display interpolated images streams in real time. In addition to the usual video controls, the Player presents two sliders that enable the viewer to modify the view position in time and space respectively (i.e. change the virtual viewpoint of the interpolation process). The GUI also provides controls to move automatically to predefined viewpoints or to swap automatically through the full range covered by the cameras with selectable steps. The player runs over a standardised media format that basically contains the JVT, plus original images data. In this way, since the player takes care of the final computation of the rendering, changing the view point is possible while still minimising the amount of information that needs to be pre-computed and optionally transmitted from the Interpolation Engine.

4 Results and evaluation

A typical EVENTS scenario (Fig. 11) consists of a number of cameras distributed in a football stadium with maximum inter-viewing angle of 20°. For practical reasons the system is arranged in two independent units, each covering a goal area, and controlling between 5 and 10 cameras. EVENTS uses low cost cameras that must be carefully aimed at the goalmouth and 18 yard area to ensure sufficient overlapping between cameras. Using images acquired at Real Madrid and Oxford United stadiums, several tests were arranged to assess the suitability of EVENTS system for the target application purposed, which according to specifications consisted of a fast replay system able to produce 30 s of delayed interpolated video in less than 60 s at 12 img/s and 800 × 600 resolution. The quality of the resulting images was evaluated in terms of image resolution, frame rate, colour consistency, severity of interpolation artifacts, image definition, etc.

The effective image resolution that can be obtained by the system is only limited by the resolution attainable from the input images (the cameras) and the computer power required to fulfill performance requirements. The optimum resolution will depend on the application scenario selected. For TV broadcasting applications the minimum resolution considered was 640 × 480, while maximum resolution would be 800 × 600 (EVENTS standard). Higher resolutions would imply increased processing requirements without noticeable quality improvements; while lower resolutions would result in poor images. For Internet applications, however, resolutions of 320 × 240 and lower could be accepted.

Once fixed the standard image resolution, the frame rate can be scaled with available computing power. EVENTS defines 12 img/sec as the adequate balance between cost and quality (higher rates can be achieved as more processors are added or individual processors are faster). According to these requirements, the system must be able to process 360 images in no more than 90 s with a maximum latency of 60 s. Figure 12 shows the target can be achieved using a 4 nodes cluster. Other combinations of resolutions versus

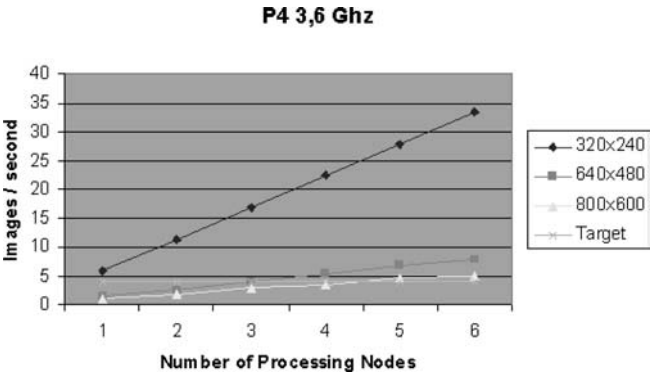

Fig. 12 Performance for different image resolutions (target 4 img/s)

number of processors can be seen in the figure: i.e. a five processor machine is able to compute in real-time full frame rate (25 img/s) interpolated video at 320 × 240.

In terms of image quality, EVENTS interpolated images were, in general, crisp, correctly focused and showed adequate accuracy in the details (see examples in Fig. 13). No noticeable blurring has been detected in the tested image sequences and irregular borders caused by the interpolation process were automatically removed by the system. When generating novel views, it is common to have different illumination levels in each of the cameras. EVENTS uses techniques for automatic illumination compensation and blending of textures to ensure colour consistency. However, when analysing individual images, minor inconsistencies (Fig. 13) could still be detected close to the borders, where discrepancies between image pairs is greater. The effect is more apparent in homogeneous areas, such as the green, and depends on the view position: the closer you are to a real camera view, the less noticeable. Scenes with strong sun lighting are also more prone to these type of problems. On the other hand, the effect is hardly perceptible in moving image sequences since the eye is highly tolerant to errors that do not affect the dynamics of the scene.

Due to the intrinsic nature of the problem, no known complex image processing process (i.e. interpolation, compression) is free from image artifacts. The important issue is to determine how the mentioned artifacts could affect the perceived image quality. It is obvious that as more defects are produced, affecting greater areas of the image, during more time, the resulting quality gets more degraded. Defects continued in time are more noticeable than sporadic

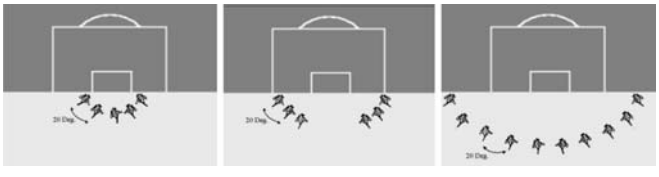

Fig. 11 Possible camera configurations for: 5, 6 and 10 cameras

Fig. 13 **a** Original left. **b** Interpolated. **c** Original right

defects, especially in dynamic scenes when some small artifacts 'are' routinely tolerated by the eye (i.e. MPEG2 artifacts). However, concentration of wrong pixels, especially if they affect systematically certain classes of objects, may disturb seriously the development of the scene. This is the case, for example, when defects concerns the dynamic parts of the scene, or objects of special relevance (i.e. active players). In that sense, EVENTS' mission is to produce highly dynamic and realistic image sequences that are pleasant to the eye. For this purpose, the following criteria for success was used to evaluate EVENTS' results:

1. Large artifacts must be eliminated anywhere in the scene.
2. All artifacts must be removed around players participating in the action.
3. All artifacts must be eliminated at the ball.
4. Small artifacts are tolerated around inactive players and the background.

Results in Fig. 13 show active players are reasonably rendered most of the time, the ball is correctly tracked and only small defects appears on players off the main scene; hence the system complies with requirements: 2, 3 and 4. However, some large defects still remain in the static part of the scene (i.e. the goal post in Fig. 13). Those artifacts in the background can be eliminated to a high degree using a tool to segment manually the offending parts before launching the interpolation engine. Attempts to achieve this objective automatically are still under development. On the other hand, EVENTS successfully handle situations that plague other interpolation approaches: i.e. players close to the end of the field of view suddenly appearing in scene, errors arising from occluding views, abrupt changes from image to image, rendering incorrectly the background, etc.

5 Conclusion

An innovative view interpolation system has been presented. The paper outlines how it is possible to exploit the advantages of JVT in sports scenarios while compensating its shortcomings using novel multi-layered video segmentation and constrained scene analysis. EVENTS did important contributions in an number of computer vision processes of general application: we developed a Wide Base Novel View Synthesis algorithm using JVT to triangulate simultaneously and consistently two images without any 3D input; we implemented a method based on a probabilistic model of object occupancy and visibility to segment and track non-rigid objects in one or more views; we implemented an EM method for automatic segmentation of a pair of densely matched views into layers based on compliance with a homographic model of transfer; we developed a novel Background Estimation Method using constrained analysis to exploit temporal coherency.

The results obtained demonstrate EVENTS correctly handles most of the situations caused by occluding boundaries and improper seed matching due to dominant textureless features. As compared with the state of the art, summarised in Sect. 1, EVENTS displays: crisper interpolated images, better colour consistency, fewer artifacts and smoother transition between cameras. The system offers a software only solution that allows flexible camera distribution and wider viewing angles (20° compared with 7° or less), resulting in less cameras required for the same area. The computing platform is scalable and may perform in a range of resolutions and frame rates according to the needs.

Sports is a demanding environment for view interpolation. EVENTS was initially aimed at football, but other application scenarios have been also tested successfully: athletics, icehockey, etc. EVENTS technology might have an impact on any application demanding arbitrary novel views in real time: live events (theater, concerts, TV contests), special effects, games, etc. Moreover, some of EVENTS' underlying technologies, such as improved JVT and real-time segmentation methods would be suitable for applications requiring high quality real-time segmentation in dynamic scenarios dominated by a background. The system is especially well-suited for outdoors where there exist a number of potential applications: video matting (either for highlighting, removal or replacement of objects), intrusion detection, tracking of simultaneous objects (tracking persons, traffic monitoring), etc. Compression technologies may also benefit from the technology; i.e. MPEG-4 and its successors are layer based, and rely on similar techniques.

At this moment, several research paths are open with the aim to reduce more the presence of interpolation artifacts and come closer to broadcasting image quality: improving reliability of automatic seed matching using recent ideas from scale invariant features; merging image-based rendering and JVT with the aim to improve the rendering of the static background; designing a layered segmentation method that combines multi-view appearance and motion models, and could further support shape models if required; extending EVENTS' features to rotating and zooming cameras; etc.

References

1. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. SIGGRAPH (2004)
2. Xiao, J., Shah, M.: Tri-view morphing. Comput. Vision Image Understanding (2004)
3. Vedula, S., Baker, S., Kanade, T.: Spatio-temporal view interpolation. Eurographics Workshop Rendering (2002)
4. Horaud, R., Dalal, N.: Indexing key positions between multiple videos. In: Proceedings of the IEEE Workshop on Visual Motion. Florida, USA (2002)
5. Horaud, R., Dalal, N.: From video sequence to motion panorama. In: Proceedings of the IEEE Workshop on Visual Motion. Florida, USA (2002)
6. Lhuillier, M., Quan, L.: Robust dense matching using local ang global geometric constraints. IAPR 00 (2000)
7. Lhuillier, M., Quan, L., Tsiu, T., Shum, H.: Relief mosaic by joint view triangulation. In: Proceedings of the IEEE Computer Vision and Pattern Recognition Conference. Hawaii (2001)

8. Quan, L., Lu, L., Shum, H., Lhuillier, M.: Concentric mosaic(s), planar motion and 1d cameras. ICCV
9. Connor, K., Reid, I.: Novel view specification and synthesis. In: Proceedings of the 10th British Machine Vision Conference. Cardiff (2002)
10. Connor, K., Reid, I.: A multiple view layered representation for dynamic novel view synthesis. BMVC03 (2003)
11. Gortler, S., Grzeszczuk, R., Szeliski, R., Cohen, M.: The lumigraph. In: Proceedings of the 23rd Conference on Computer Graphics and Interactive Techniques, pp. 43–54 (1996)
12. Lhuillier, M.: Joint view triangulation for two views. Vision Interface 99 (1999)
13. Lhuillier, M.: Towards automatic interpolation for real and distant image pairs. INRIA, Technical Report 3619 (1999)
14. Lhuillier, M., Quan, L.: Edge-constrained joint view triangulation for image interpolation. CVPR'00 (2000)
15. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cut. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2004)
16. Ayer, S., Sawhney, H.: Layered representation of motion video using ro-bust maximum-likelihood estimation of mixture models and mdl encoding. In: Proceedings of the International Conference on Computer Vision (1995)
17. Connor, K., Reid, I.: Tracking occluding layers in multiple views. University of Oxford, Department of Engineering Science, Technical Report 21125-0401 (2004)
18. Laveau, S., Faugeras, O.: 3d scene representation as a collection of images and fundamental matrices. Inria, Technical Report 2205 (1994)
19. Knight, J., Connor, K., Reid, I.: Image based rendering using scale-space. University of Oxford, Department of Engineering Science, Technical Report 21125-0412 (2004)