Machine Vision and Applications (2005) 16(4): 246–257
DOI 10.1007/s00138-005-0179-4

ORIGINAL PAPER

Tomás Rodríguez · Peter Sturm · Pau Gargallo · Nicolas Guilbert · Anders Heyden · Fernando Jauregizar · J. M. Menéndez · J. I. Ronda

Photorealistic 3D reconstruction from handheld cameras

Received: 24 March 2004 / Accepted: 29 March 2005 / Published online: 10 June 2005

Abstract One of the major challenges in the fields of computer vision and computer graphics is the construction and representation of life-like virtual 3D scenarios within a computer. The VISIRE project attempts to reconstruct photorealistic 3D models of large scenarios using as input multiple freehand video sequences, while rendering the technology accessible to the non-expert.

VISIRE is application oriented and hence must deal with multiple issues of practical relevance that were commonly overlooked in past experiences. The paper presents both an innovative approach for the integration of previously unrelated experiences, as well as a number of novel contributions, such as: an innovative algorithm to enforce closedness of the trajectories, a new approach to 3D mesh generation from sparse data, novel techniques dealing with partial occlusions and a method for using photo-consistency and visibility constrains to refine the 3D mesh.

Keywords Photo-realistic 3D reconstruction · Self calibration · Structure from motion · Image based rendering (IBR) · Video analysis

1 Introduction

Traditionally, reconstructing large scenarios in 3D has been costly, time consuming, and required expert personnel. Usually the results showed artificial look and produced unmanageable heavy models. However, recent advances in the areas of video analysis, camera calibration, and texture fusion allow us to think in a more satisfying scenario, where the user just needs to wander around, aiming his camera, making shoots, following the provided guidelines, and the system will automatically do the 3D reconstruction of the desired scenario for him. Our objective is to come closer to this ideal scenario, but it is our belief that current state of the art does not allow still for reliable full automatic 3D reconstruction. For that reason we avoid dogmatic views and accept human cooperation in the 3D reconstruction process whenever it can lead to better results, faster processing, or a personal touch.

In this document, the results of the EC funded project VISIRE (IST-1999-10756) are presented. VISIRE [1] attempts to reconstruct in 3D photorealistic interiors of large scenarios from multiple freehand video sequences, while rendering the technology accessible to the non-expert. VISIRE offers an advanced authoring tool that empowers the user to interact effortlessly with the underlying Computer Vision (CV) software with the aim to process the acquired video material off-line and obtain lightweight 3D models highly resembling the original. VISIRE observed certain basic assumptions that greatly influenced the design of the system: no expert CV personnel should be needed, no knowledge about the camera was assumed (i.e. unknown intrinsic and extrinsic parameters), no proper calibration should be required, and the system should work with the only aid of a domestic camcorder (i.e. professional cameras, tripods, lighting or measurement devices were discarded).

VISIRE deals with several CV disciplines: auto calibration, structure from motion, non-linear robust and iterative methods, texture and geometry representation, etc. There is a general belief in the scientific community that these issues have been mostly solved. This statement is correct with respect to the basic principles, but there is still a big gap that must be filled between the scientific demonstration and a technology that “works." The fact is the problem of automatic 3D reconstruction of complex scenarios remains largely unsolved and the technology never found its way to the market in spite of its unquestionable interest.

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

P. Sturm · P. Gargallo
INRIA Rhône-Alpes Montbonnot, France

N. Guilbert · A. Heyden
Centre for Mathematical Science, Lund University, Sweden

F. Jauregizar · J. M. Menéndez · J. I. Ronda
E.T.S.I. Telecomunicaciones, Universidad Politécnica de Madrid, Spain

VISIRE is application oriented and hence must approach multiple issues of practical relevance. As opposed to previous experiences, aimed at technological demonstrations based on ad-hoc solutions that must be modified by the experts for every new situation or partial 3D reconstructions of elements of the scenario specially selected for the task, VISIRE offers the innovation to consider a "global" approach and proposes instead methods and tools able to solve a number of general situations. Obviously this approach is more challenging and the price to pay is the need to define certain restrictions of use and consider new difficulties previously ignored in this type of application, that now acquires the category of critical problems: reliable tracking in sparse environments, introducing constraints such as planarity and closedness, occlusion reasoning, optimized autocalibration, combining and adapting textures from multiple viewpoints, multiresolution, integrating manual and automatic mesh generation methods, etc.

In that sense, existing autocalibration [2] methods are mainly based on a small set of images, ranging from 2 to 20 in today's very complex systems. VISIRE aims to break this barrier in several orders of magnitude, mainly because it can use the thousands of images typically found in video streams. No system so far has ever tried to accomplish such a complex scenario, in any of the above-mentioned tasks. Another important goal is to produce photorealistic 3D models, i.e. models that may be realistically rendered from synthetic viewpoints. One way of doing so [2–4] is to apply IBR (Image-Based Rendering). The other main method [5] is to enhance a geometrical 3D model with texture maps or other information e.g. surface reflectance properties. Up to now, we concentrate on the second solution, which produces a more compact scene description. The challenge for the 3D mesh generation process is that most existing methods are designed for sets of dense and regularly distributed 3D points. This is usually not the case in automatic structure from motion, so we have developed methods that use information provided by the input images, via visibility and photoconsistency constraints.

VISIRE follows a standard process division (see Fig. 1): we start out in Sect. 2 with Feature Analysis. We continue with a description of Calibration methods in Sect. 3. Next, the approach to 3D Registration and Mesh Generation is introduced in Sect. 4. The authoring tool is illustrated in Sect. 5. Results and experimental evaluation are presented in Sect. 6. Finally, we end in Sect. 7 with the conclusions.

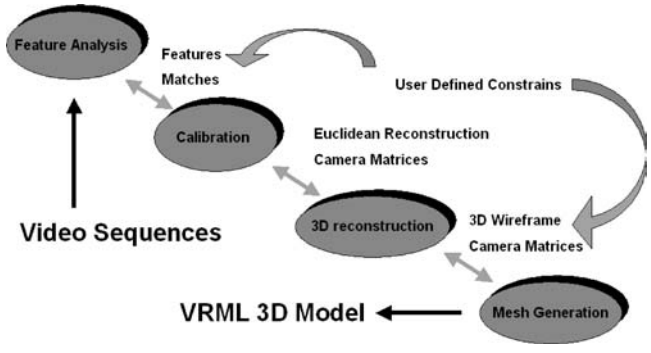

Fig. 1 VISIRE computer vision chain

2 Feature analysis

The method selected for feature extraction in VISIRE relies on the Tomasi/Kanade [6] approach, which first smoothes the image by convolving it with a Gaussian and then the gradients (in x and y) are computed by convolving the resulting image with the derivative of a Gaussian. Later, a measure of cornerness is applied for each pixel, evaluating the minimum eigenvalue of the 2×2 gradient matrix computed in a 7×7 window around the pixel. This measure is used to sort in descending order all pixels in the image, ensuring that selected features are at least 10 pixels away from each other.

Feature tracking is applied through a robust method, based on the Kanade–Lucas–Tomasi approach [6] that relies on the assumption of an affine motion field in the projection from the 3D point motion to the 2D surface of the image, and on the computation of a weighted *dissimilarity* function between consecutive images that is minimized using a Newton–Raphson iterative method, over a limited spatial window. After the trajectories are generated along time, they are validated by checking their compliance to the assumed model of rigid motion of the scene. To this purpose the set of trajectories is processed in two steps corresponding, respectively, to a local processing and a global processing. For the local processing the sequence is divided into non-overlapping temporal windows and for each of them a RANSAC-based calibration is performed, which is later optimized by bundle adjustment. The temporal windows must be short enough to ensure that enough trajectories remain complete within it but long enough to include enough motion. Local data from different temporal windows are consolidated and reoptimized in the global processing step, which operates iteratively consolidating data from pairs of adjacent windows. After this analysis, a trajectory or part of a trajectory results validated when it is successfully approximated by the reprojection of a scene feature.

3 On-line calibration

Once a sufficient amount of reliable image correspondences have been established the overall structure of the scene may be recovered. This recovery is performed in two major steps, namely the recoveries of projective and Euclidian structure, respectively. Projective structure is recovered by extracting camera matrices from the trifocal tensor, see e.g. [7] and followed by a series of the resectionings in order to obtain each of the subsequent cameras. However, the result is only defined up to a projective transformation, and is consequently useless for visualization purposes. However, as shown in [8], very general constraints such as assuming square pixels suffice to establish the in- and extrinsic camera

calibration parameters. In VISIRE, the actual implementation makes use of the Cheirality inequalities, see e.g. [7] followed by an identification of the plane at infinity and eventually the recovery of the intrinsic and extrinsic camera parameters and Euclidian structure.

Nevertheless, the estimation of the initial set of cameras depend on the solution of a linearized problem, and are consequently subject to errors. Hence, in order to achieve a maximum likelihood solution, so called bundle adjustment is applied, see eg. [9] for details. This involves minimizing the reprojection error

$$\sum_{i,j} \|(x_{ij} - p(P_i, X_j))\|^2$$

where x_{ij} indicates the jth image point in the ith image, and $p{:}(P, X) \mapsto \mathbb{R}^2$ projects the 3D homogeneous point X using the camera matrix P. In VISIRE, bundle adjustment is implemented using the Levenberg–Marquardt method and a sparse system solver, allowing for significantly more effective processing and for longer sequences than previously, i.e. sequences of up to 300 views and 15,000 3D points.

In building complete systems for solving structure and motion, new questions and research subjects arise naturally. One unique feature of the VISIRE system is the ability to apply the constraint of *closedness* to a sequence. In a long sequence, the same image feature is likely to appear on several occasions, but will, however, under normal conditions be reconstructed as a different 3D feature each time. For the general case, enforcing identity on these features turns out to be indispensable. Also, small errors and degrees of freedom from partial reconstructions might accumulate to a very large error so that the scene structure or camera motion obtained from the feature the first time it is encountered might not fit at all when it is re-projected to the images where the feature appears later. One way to deal with this problem is to make partial reconstructions from subsequences and then stitch these together by minimizing the distances between corresponding points in 3D via homographies of the substructures as it is done in [10]. Another way would be to impose soft constraints as described in [11], where a penalty term on the difference between expected identical parameters is included as a Lagrange multiplier in the error function.

Both methods have important drawbacks we intended to overcome. Our initial approach was to distribute the accumulated error equally on all the parameters, although in the norm given by their covariance. Specifically, the reprojection error vector and its associated covariance structure are projected onto their respective lower-dimensional manifolds corresponding to the reduced system, i.e. the system where identity of the parameters has been enforced. Using the resulting values, the optimal reduced parameters are calculated through the equivalent of a Levenberg–Marquardt iteration.

3.1 Batch reconstruction from sparse data

This approach evolved to a method that takes closedness constraints into account in the very first reconstruction step (*Batch* process). This is in itself interesting for robustness reasons, since as many constraints as possible should be enforced as early as possible to avoid ending up in a erratic situation. Another robustifying feature of the algorithm is that the auto-calibration step is performed from affine to Euclidian, which is significantly simpler than the original projective to Euclidian. However, the algorithm has turned out to play a more important role: basically, the sequential approach originally used in the VISIRE project (and most other state-of-the-art structure from motion systems) is best suited for applications where decisions need to be made as soon as a new frame becomes available (i.e. robotics). There has previously been no alternative, since existing batch algorithms [12–14], in practice, have required *all* features to be visible in *all* images; something that is unlikely in a real scenario.

The new method developed [15] proposes a batch algorithm that would work on sparse data. The basic idea is to compute matching tensors between the images (fundamental matrices, trifocal or quadrifocal tensors) and finally determine all of the camera matrices in a single computational step. The notion of $\mathbf{F} - \mathbf{e}$ closure constraint, i.e. $\mathbf{F}_{12}\mathbf{P}_2 + [\mathbf{e}_{21}]_\times \mathbf{P}_1 = \mathbf{0}$ was introduced in [16], denoting bilinear constraints between camera parameters and matching tensors. We derived an alternative closure constraint, the $\mathbf{F}-$closure:

$$\mathbf{X}^{\mathrm{T}} \underbrace{\mathbf{P}_1^{\mathrm{T}} \mathbf{F}_{12} \mathbf{P}_2}_{\varphi} \mathbf{X} = \mathbf{0}, \ \forall \mathbf{X} \in \mathbb{P}^3. \tag{1}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ denote the camera matrices, $\mathbf{F}_{12}$ is the fundamental matrix, and $\mathbf{X}$ a set of 3D homogeneous points. In the affine case, the structure of φ becomes particularly simple:

$$\varphi = \mathbf{P}_2^{\mathrm{T}} \mathbf{F}_{12} \mathbf{P}_1 = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{a} \\ -\mathbf{a}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \tag{2}$$

By re-arranging (2) and by denoting the elements of $\mathbf{F}_{12}$ by

$$\mathbf{F}_{12} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix}, \tag{3}$$

we obtain four linear constraints on the coefficients of $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_3 & -e \end{bmatrix}}_{\mathbf{r_{12}}} \tag{4}$$

These constraints apply for each pair of views $\mathbf{P}_{i_1}$ and $\mathbf{P}_{i_2}$, $i_1 \neq i_2$, provided $\mathbf{F}_{i_1 i_2}$ is defined. We construct a

linear system of equations using 4 with the form $\mathbf{S}P = R$:

$$\begin{bmatrix} \mathbf{s}_{12} \\ \mathbf{s}_{1i_1} \\ \vdots \\ \mathbf{s}_{i_k i_m} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{P}_m \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{12} \\ \mathbf{r}_{1i_1} \\ \vdots \\ \mathbf{r}_{i_k i_m} \end{bmatrix} \tag{5}$$

where $\mathbf{r}_{i_1 i_2}$ is the right hand side of Eq. 4 and $\mathbf{s}_{i_1 i_2}$ are $1 \times 2m$ row vectors.

$$\mathbf{s}_{i_1 i_2} = \begin{bmatrix} \cdots & \underbrace{a \;\; b}_{\text{FirstBlock}} & \cdots & \underbrace{c \;\; d}_{\text{SecondBlock}} & \cdots \end{bmatrix} \tag{6}$$

One important advantage of the proposed algorithm is the ability to include different types of constraints, such as equality of given cameras. This feature is illustrated in Fig. 2 where a cubic point cloud is reconstructed given views taken on a circular trajectory.

The point cloud consists of 300 3D points evenly distributed in the cube $[0.5] \times [0.5] \times [0.5]$ and of 30 cameras with focal length $f = 100$ equidistantly placed on a circular path centered at $(0, 0, 0)$. Each frame contains features which are visible in the nine following frames. Gaussian noise with $\sigma = 1$ is present in the images. Figure 2b shows the initial reconstruction of the camera trajectory using affine approximation and in Fig. 2c an alternative reconstruction where equality has been assumed between the first and the last camera in the sequence. The perspective equivalents of the affine cameras were obtained by choosing a focal length $< \infty$ ensuring that all the 3D points would lie in front of the cameras where they had been observed.

Clearly, the initial reconstructions capture the overall structure of the scene and the motion, thus allowing for the subsequent bundle adjustment to converge to the global minimum. One point of special interest is the fact that within this framework, the affine camera model approximates the perspective camera sufficiently well, even though the depth of the object is approximately the same as the distance to the object, i.e. a lot more than the 10% that are usually considered the upper limit.

4 3D registration and mesh generation

Creation of photorealistic models is done in two major steps: based on 3D points reconstructed during on-line calibration or subsequently, we first generate a triangular mesh describing the scene's surfaces; then, texture maps for the surface patches are extracted using all available images.

After on-line calibration, we are provided with a set of 3D points, projection matrices of a set of images, and 3D-to-2D point correspondences. Usually, only interest points that could be tracked reliably, were used for on-line calibration and metric 3D reconstruction. However, once projection matrices are known, additional point tracks can be checked more easily for outliers if the correspondingly reconstructed 3D points are reliable. We may thus enrich the set of 3D points before proceeding to mesh generation.

4.1 Geometric consistency constraints

Most existing methods for mesh generation from 3D points rely mainly on 3D geometric reasoning, e.g. proximity constraints between points, see e.g. [17, 18] and references therein. These methods give unusable results for our input data, because they are designed for rather dense and regular point sets. In order to work with more difficult data, other information besides pure 3D geometry should be used. Since the 3D points are obtained by reconstruction from images, we have such additional information. First, visibility constraints can be imposed to prune incorrect surface patches, e.g. a hypothetical patch that would lie between a 3D point and the optical center of a view where that point is visible, can be rejected (see left part of Fig. 3). Other, more complicated, visibility constraints are also used: especially, a surface patch that partially occludes another one, without occluding an actual 3D point, is rejected (see right part of Fig. 3). The drawback of verifying this situation is quite a large computation time (naively, each hypothetical triangle has to be tested against each triangle already in the mesh). Nevertheless, and although this situation is rare, it occurred in all our experiments; not taking it into account would result in visually unpleasing models. Such visibility constraints were also used in [19], where a surface mesh is built

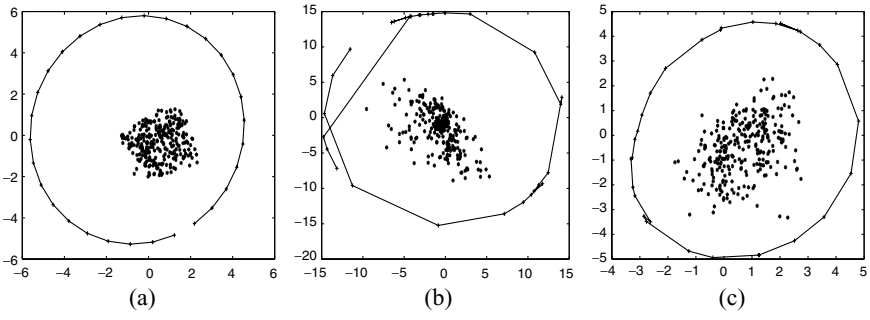

Fig. 2 Reconstruction, object centered configuration **a** original configuration, **b** initial reconstruction using affine approximation, **c** same as, **b** but assuming equality between the first and last camera

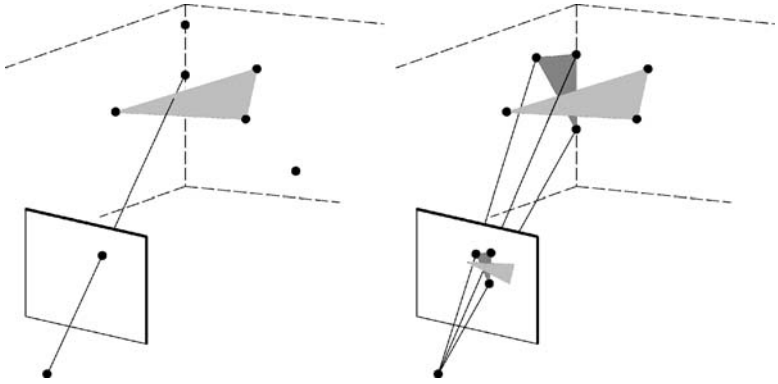

Fig. 3 Visibility constraints. *Left*: the triangle will not be accepted since it would occlude a 3D point from a view where that point is visible (a corresponding 2D interest point was extracted). *Right*: the light triangle would not be accepted since it would partially occlude a triangle already existing in the mesh

incrementally, starting with a mesh obtained by a Delaunay triangulation in one view, and then rejecting and adding triangles based on visibility constraints of one additional view after the other. We proceed differently, by iteratively adding new triangles to automatically selected seed triangles, and thus by letting a mesh grow, directly ensuring all available visibility constraints (and other constraints, see below). This way, we may end up with a better connected surface mesh, which may be easier to edit/complete if necessary.

4.2 Photometric consistency constraints

Another constraint we use to test hypothetical surface patches, is photoconsistency: a planar patch is acceptable, if its projections into all images where the patch's vertices are visible, correspond to image regions with the same "texture." This is verified by the following process: image regions corresponding to a planar 3D surface patch, are warped (using homographies associated to the 3D plane) to some common frame (to undo perspective effects). The simplest method to measure photoconsistency would then be a "multi-image cross-correlation" using all warped image regions (e.g. compute variance of greylevels) [20]. This can be problematic, for example in cases where some images of a surface patch are partially occluded or show specular highlights. Also, individual images are taken from different viewpoints which usually result in changes of perceived intensity values. To this end, we measure photoconsistency using the following general and robust approach: we estimate an "average" texture map for the considered patch as well as parameters for intensity transformations for the input images. This is a non-linear optimization problem, whose cost function, in its most general form, is as follows (for a patch with m pixels, seen in n images):

$$\sum_{p=1}^{m}\sum_{i=1}^{n}\sum_{k=R,G,B}\rho(I_{ikp}-\alpha_{ik}T_{kp}-\beta_{ik}) \tag{7}$$

Here, T_{kp} is the intensity value of the p-th pixel of the generated mean texture map, for color channel k. I_{ikp} is the corresponding intensity value, measured in image i. The α_{ik} and β_{ik} are parameters for affine intensity transformations, for image i and color channel k (we have implemented several modes for intensity transformations: one affine transformation per color channel, but also restricted modes with the same affine transformation for all channels or only intensity scaling or offset for example). Finally, $\rho(\cdot)$ is an influence function, that serves for weighting residuals; $\rho(x) = x^2$ corresponds to a least squares cost function which is highly non-robust. Here, we use robust influence functions [21], for example the Huber-function, that downweight the influence of outliers, which thus allows to handle specular highlights in some images etc.

Optimization is done for the T_{kp}, α_{ik}, and β_{ik} and is carried out using an M-estimator [21] (IRLS, Iteratively Reweighted Least Squares). The estimation is initialized as follows: we initialize the T_{kp} by the average of the corresponding input greylevels I_{ikp}, i.e.:

$$T_{kp} = \frac{1}{n}\sum_{i=1}^{n} I_{ikp}$$

We then compute the initial values for the affine transformation coefficients α_{ik} and β_{ik} by minimizing (7) over these parameters (keeping the T_{kp} fixed), and with $\rho(x) = x^2$ as influence function. This is thus a linear least squares problem, solved using an SVD (Singular Value Decomposition) [22]. After this initialization, we optimize the T_{kp}, α_{ik}, and β_{ik} using IRLS, as mentioned above, now using a robust influence function for ρ. This proceeds in iterations, as follows [21]: at each iteration, we first compute weights w_{ikp} by evaluating the influence function for each residual (each term $I_{ikp} - \alpha_{ik}T_{kp} - \beta_{ik}$ in (7)), to be precise by evaluating it at residuals after they are scaled by a global factor (see [21] for details). Then, we solve the weighted least squares problem:

$$\sum_{p=1}^{m}\sum_{i=1}^{n}\sum_{k=R,G,B} w_{ikp}(I_{ikp}-\alpha_{ik}T_{kp}-\beta_{ik})^2$$

This is a non-linear least squares problem, which we solve using the Levenberg–Marquardt method [22]. Here, we exploit the sparse structure of the normal equations to

drastically reduce the computation time, as it is common practice for e.g. bundle adjustment, cf. [7].

The process of computing weights and solving the weighted least squares problem, is iterated until convergence (we use a small, fixed number of iterations, which proved to be sufficient in practice). This optimization process is rather time-consuming and we thus do not use it routinely for testing hypotheses of surface patches. However, it is sometimes employed as such to generate texture maps for the final surface mesh, depending on the desired visual quality.

4.3 Overall procedure for mesh generation

We have so far described the geometric and photometric constraints used for mesh generation. The overall process is as follows. One or several "seed triangles" are created. This can be done manually, since it creates little overhead, but we also tried a simple automatic procedure: determine the smallest roughly equilateral triangles in the reconstructed point cloud, and accept them if they have a good photoconsistency measure. The mesh is thus initialized as the set containing one or several such seed triangles. The edges that are at the border of the mesh, are stored in a list. As soon as the list is not empty, the following operations are run. We first randomly pick one edge of the list. Then, all 3D points are determined for which the triangle formed by a point and the edge is not too thin (no angle smaller than 10°, for example). Sort these points by increasing distance to the edge. For the closest point, check if the triangle it would form with the edge, satisfies all geometric and photometric constraints (see above). If this is the case, accept the triangle in the mesh and update the associated data structures (e.g. remove the edge from the list, and add the new outer edges to it). In the opposite case, proceed with the next point. If none of the points satisfies all criteria, the edge is removed from the list. After having thus processed one border edge of the mesh, we iterate by randomly selecting another one, as explained above.

4.4 Other mesh generation approaches

Other approaches for mesh generation were also developed. Instead of iteratively growing a surface mesh, an alternative approach is to perform a volumetric reconstruction: starting with a discretization of 3-space (typically, a 3D Delaunay tetrahedrization), one iteratively prunes volumes (here, tetrahedral), based on similar constraints as those used above. The outer surface of the final volume is then taken as surface mesh. Constraints used for pruning tetrahedra are visibility constraints (same as above), and photoconsistency constraints, which are now applied differently: a tetrahedron is kept if its visible faces have a good photoconsistency score; if a face has a low score, the tetrahedron is pruned only if this would increase the photoconsistency of the entire model (pruning a tetrahedron makes other tetrahedra visible, which might have an even lower score). This is still work in progress, but it has already produced better results than the mesh-growing method in cases where feature points were not tracked over many frames. In the opposite case, however, the mesh-growing method tended to stick better to the true object surface.

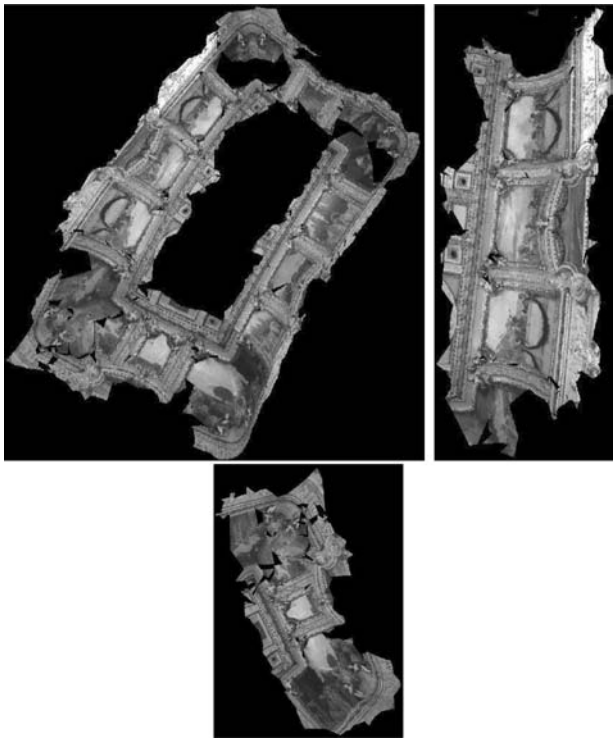

Fig. 4 Ceiling of the Casino Royal Hall. Raw Automatic 3D Model

4.5 Refining a mesh

We recently have developed an approach to refine 3D structure to get more dense reconstructions. The approach is inspired by [23], and roughly works as follows. 3D structure is estimated as a set of depth maps (as opposed to using a single depth map as in [23]). These are initialized from the coarse triangular mesh, or even by directly interpolating from the 3D point cloud. Then, they are optimized based on photoconsistency and visibility reasoning, as well as a 3D shape prior. The method resembles that of [23], but many details are novel. It would be beyond the scope of this article to completely describe this approach; it will be published separately. Results are shown in Sect. 6.

Results are shown in Figs. 4–6. Figure 4 shows a detail of the Casino scene, cf. Fig. 8. Figures 5 and 6 show results for outdoor scenes. Especially the scene of Fig. 6 is very challenging, due to the enormous depth discontinuities. For Fig. 5, the calibration provided by the authors of [23], was used, whereas for Fig. 6, the self-calibration and reconstruction tools described in this paper, were applied.

5 The authoring tool

Fully automatic 3D reconstruction of complex environments is currently not a practical possibility for a number of

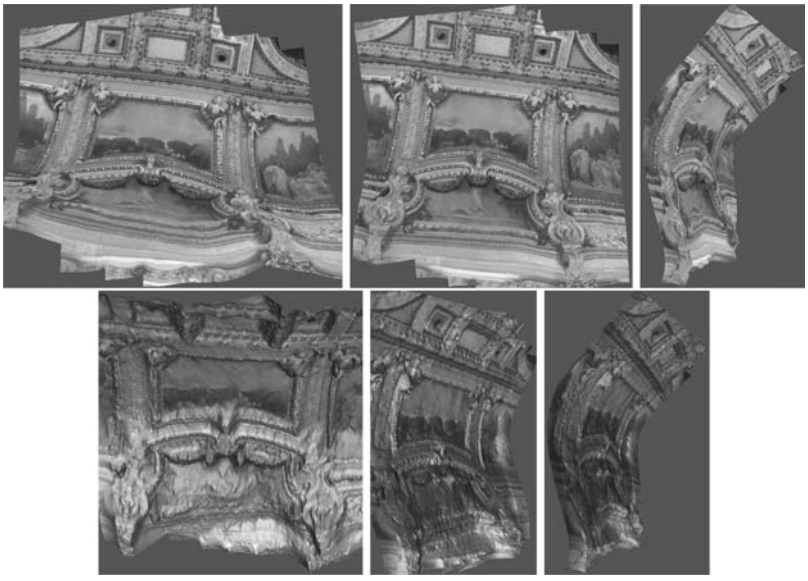

Fig. 5 Casino sequence. *Top row*: rendered images. *Bottom row*: rendered images with artificial specular component added to textures, to better show the underlying geometry

Fig. 6 Cityhall sequence [23]. *Top row*: the three input images. *Middle and bottom rows*: rendered images

reasons. Manual intervention is still required whenever CV software is not able to cope with singular situations, when automatic procedures are too slow, or simply when users desire to add a personal touch; there are situations when a few keystrokes may save hours of computational time or significantly improve the quality of the results. The VISIRE Authoring Tool (VAT) has been designed specifically to operate the underlying CV software. The tool (Fig. 7) supports the following functions: manipulate the video material required to construct a 3D model, guide and operate automatic CV processes selecting the various parameters, and visualize 2D and 3D results.

The VAT offers options to monitor the different CV steps, visualize intermediate results, and modify parameters

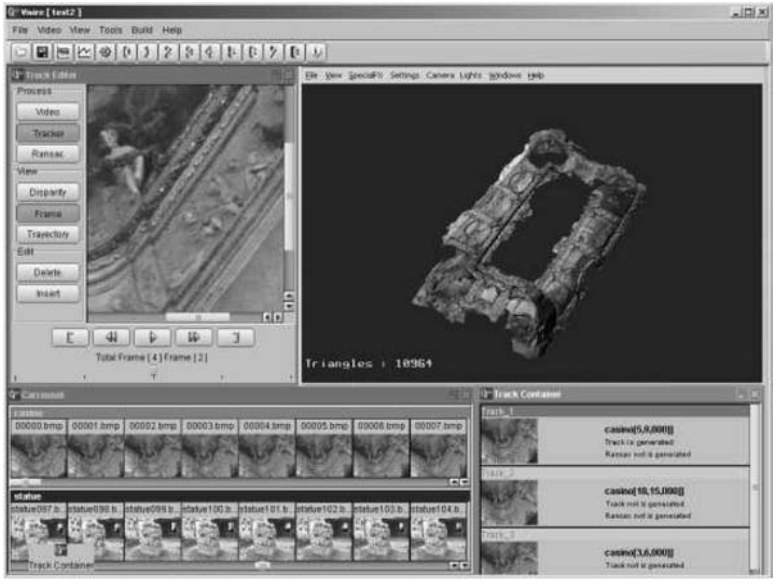

Fig. 7 VISIRE Authoring Tool Screenshot

Fig. 8 Statue sequence. *Top row*: the five input images. *Middle and bottom rows*: rendered images. The background is not perfectly reconstructed, since much of it is only seen in two or three images

with the aim to improve the finished results. The user may, for example, visualize trajectories in the track editor and manually insert or delete features that are automatically tracked by the system. The VAT follows a “project" approach: a project gathers all the information necessary to manipulate and process one or more video sequences. When the user modifies a parameter, the system automatically recalculates obsolete links and updates the resulting 3D mesh.

Project information is stored in three main structures: Videos (2D image sequences), Tracks (structures containing the information required to track individual features in one or multiple video sequences), and Geometries (3D points, 3D meshes, and textures). The process of constructing a 3D model is progressive. Different parts of the scenario are processed independently and then stitched together. Three containers store the respective structures as they are created and allow the user to manipulate them using a

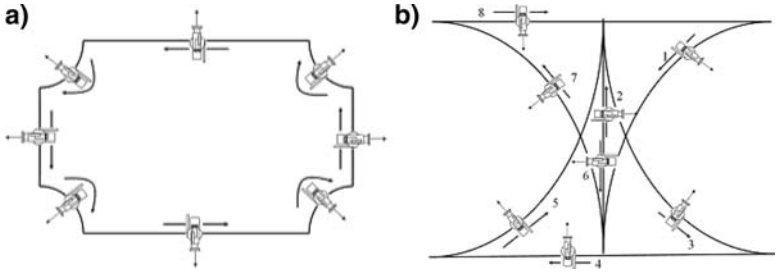

Fig. 9 Example of a shooting plan: **a**) Large room, **b**) Small room

drag and drop interface. Several combinations are possible: object structures can be added, deleted, and joined (i.e. it is possible to join videos, tracks, or geometries). This last is a unique feature since it is almost impossible to cover a complete scenario in a single shot.

The VAT implements the following additional characteristics: integrated video editor, support for multiple input video sequences, compatible with most popular video formats, VRML output, integrated 3D browser, multiple interactive tools (feature analysis, calibration, 3D rendering, texture processing), and hot display capabilities (the result of changed parameters is immediately updated in the 3D model).

6 Results and evaluation

An exhaustive evaluation of VISIRE methods and software in real conditions was performed using video material acquired in different museums (Casino de Madrid, Uffizi Gallery, and Palazzo Pitti in Florence), as well as available test image libraries. Early in the project stage it was acknowledged that image shooting procedures would become critical and might determine the quality of the resulting 3D models. For that reason precise guidelines (Fig. 9) were produced so that any person with minimum skills and basic training could do the shooting. In most cases a mid size room could be completed in no more than 2 h if guidelines are closely followed. Tests were made using professional digital Betacam cameras and domestic minDV camcorders. Due to unusual shooting requirements, miniDV cameras performed better than their professional counterparts since miniDV cameras are lighter and easier to aim and differences in image quality were hardly noticeable.

The abundant video material compiled was used to construct several complete 3D models. Experience showed VISIRE performs better in richly textured scenarios. Painted walls (i.e. frescos) or textured materials (i.e. marble) produced very good results. The system offers remarkable accuracy in the reconstruction of vaulted scenarios and irregular shapes where human modellers would have difficulties to achieve similar results. In planar surfaces where extremely good accuracy in the assembly of the geometry is required and even small errors are highly noticeable, performance was more noisy; but nevertheless very good for an automatic method.

As compared with the state of the art, VISIRE does a great job handling occlusions by using information from alternative views when part of the scenario is occluded. However, in certain circumstances the algorithm incorrectly joins polygons from different objects. In most cases, those problems are due to irregular sampling and can be corrected manually during post-processing (i.e. inserting seed features manually). VISIRE also has difficulties with reflections (i.e. mirrors) or unstable illumination. It must be considered the system was required to use handheld cameras and no professional lighting or measurement devices were allowed. Ill posed situations, can be found in some cases due to improper shooting of the images; most frequently when disparity between the acquired images is not sufficient. Fortunately, the system detects this situation automatically. There were also rare situations when the algorithm joining sequences did not perform as expected and the model required manual adjustments for a proper joining.

Generally speaking, VISIRE textures and lighting are more realistic than human produced models. The geometry, while closer to the real thing, tends to be more noisy and error prone. In that sense the system obtains a typical RMS of no more than 0.338 in the reprojected points after projective bundle and a maximum of 0.5 after Euclidian bundle. Those figures are very good considering they were obtained using fully automatic methods.

Results are shown in Figs. 4–6 and Fig. 10. Figure 4 shows a detail of the Casino scene, cf. Fig. 8. Figures 5 and 6 show results for outdoor scenes. Especially the scene of Fig. 6 is very challenging, due to the enormous depth discontinuities. For Fig. 5, the calibration provided by the authors of [23], was used, whereas for Fig. 6, the self-calibration and reconstruction tools described in this paper, were applied. In Fig. 10 an almost complete model of the ceiling of the Palazzo Pitti is presented.

Next, we will analyze in more detail one of the video sequences evaluated (Fig. 8). The Casino sequence consisted of four slightly overlapping shots describing the edges of a near-rectangular camera path. The whole sequence lasts several min and was in the process reduced to 281 views and 16,000 3D points. In order to bundle efficiently, only the 835 longest point tracks were kept in the global bundle adjustment. Figure 11a shows the magnitude of the trajectory mismatch is such that usual bundle turned out to be insufficient. Results improved dramatically when constraint enforcement was performed as it is apparent in Figs. 11b

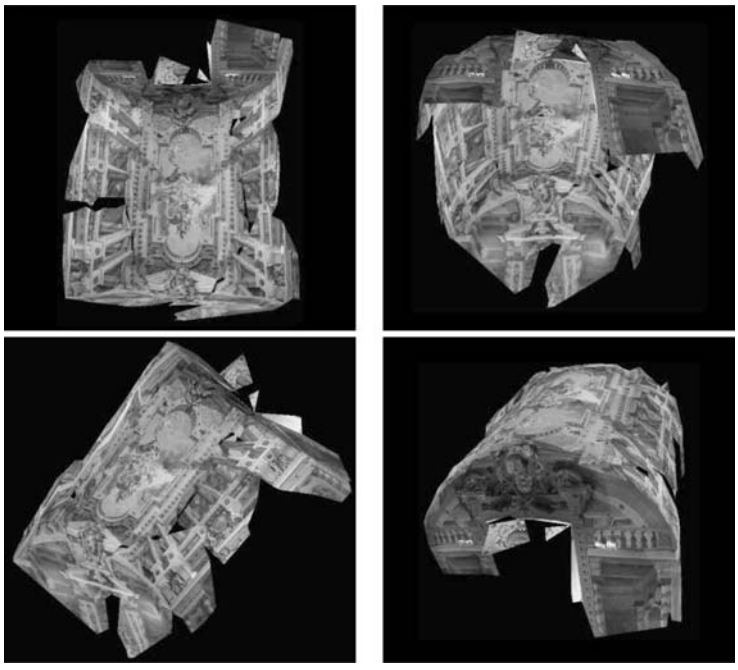

Fig. 10 Different views of a model from Museo degli Argenti in Palazzo Pitti

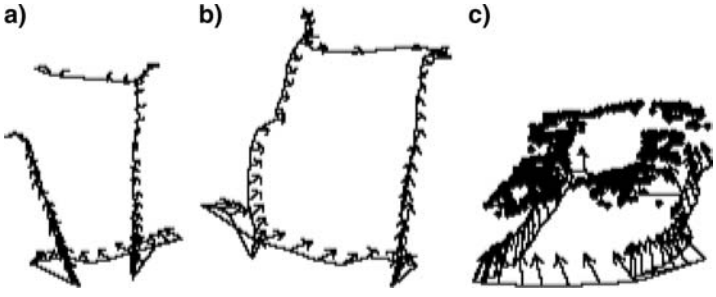

Fig. 11 Casino sequence: **a** open sequence trajectory, **b** trajectory after closing, **c** final reconstruction of closed sequence with 3D points

Table 1 Reprojection errors (RMS) for the Casino sequence

(Pixels)	Open sequence	Constraint enforcement	Closed sequence
Overall	0.3340	15.54	0.3667
Closure	777.6	10.55	0.6098

and c. The first row in Table 1 presents the overall RMS reprojection error for all image points, while the sequence is still open, after imposing the closure constraint and after bundle convergence of the closed sequence, respectively. The second row provides similar information when reprojecting the five 3D points used for merging (hence visible in both the last and the first images) onto the first image. Note that the overall reprojection error increases when the constraint is enforced, since the error on the constraining image points is in a sense distributed on the whole structure. The important measure is, however, how well the algorithm subsequently minimizes the reprojection error, which is seen to fall very close to the error for the non-constrained structure.

7 Conclusion

In the paper a “complete" approach to 3D reconstruction in real environments has been presented. The VISIRE system succeeded in the use of video information, acquired from handheld camcorders, as input to a near-automatic 3D mesh generation system. A full functional authoring tool has been developed to allow graphics professionals create photo-realistic 3D models with less effort and better quality than it was possible before. The system is specially well suited for scenarios where architecture is rich and textured, with few first plane objects occluding the view. Vaulted scenarios, painted walls, or irregular geometries are excellent candidates, while scenarios with simpler geometries, fewer textures, or showing big symmetries (i.e. where replicated geometrical primitives can be used) are better suited for manual modelling. In general, best performance is obtained when VISIRE is used as an initial automatic step that is completed by human post-processing. To some extent manual and automatic processes are

complementary, but there still exists important challenges as to how to combine both approaches in a most efficient way.

Even if VISIRE achieved important advances, the field is still open to new improvements. In particular, it will be highly desirable to implement methods to apply constraints (i.e. planarity and pure rotations) to the produced 3D models in order to simplify the 3D reconstruction process and improve the accuracy. One of our goals for future research is to combine 3D and IBR approaches; applying IBR for scene parts whose geometry cannot be modelled well. Another possible point of improvement is to develop more advanced methods for representing objects. Instead of triangular meshes, more general object surfaces could be used, e.g. level-set frameworks. Here it would also be desirable to include other surface properties, such as reflectance and local geometry. More research will be also required to improve some of the still pending problems arising from: severe occlusions, texture-less or repetitive structures, irregular sampling, etc. Finally, it must be mentioned it still remains a difficult task to acquire sequences that cover every part of a scene from several viewpoints. It is imaginable to develop a system that detects parts of the scene that were not modelled precisely, and guides the user to acquire the additional video material.

Our main conclusion is the system is capable to perform a good job in a broad range of conditions. There are still problems related to CV, which presents limits that may cause certain defects on the reconstructed models. This is certainly a limitation, but is well compensated by the automatic functioning of the system and the processing speed, which allows to achieve a model of reasonable good quality in a very short time. In our opinion the system can indeed simplify the process of building up 3D models or at least provide a good prototype of a model to work on.

References

1. Rodríguez, T., Sturm, P., Heyden, A., Menéndez, J.M., et al.: Visire. photorealistic 3d reconstruction from video sequences. In: IEEE International Conference on Image Processing, pp. 705–708. Barcelona, Spain (2003)
2. Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. IJCV **59**(3), 207–232 (2004)
3. 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)
4. Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., McMillan, L.: Image-based 3d photography using opacity hulls. In: Proceedings of the ACM SIGGRAPH 2002, p. 427–437 (2002)
5. Takashi Machida, H.T.: Dense estimation of surface reflectance properties based on inverse global illumination rendering. In: ICPR'04, vol. 2, pp. 895–898. Cambridge, UK (2004)
6. Shi, J.C.T.: Good features to track. In: CVPR'94, pp. 593–600 (1994)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, UK (2000)
8. Heyden, A., Åström, K.: Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 438–443 (1997)
9. Triggs, W., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. In: Vision Algorithms: Theory and Practice. Springer, Berlin Heidelberg New York (2000)
10. Fitzgibbon, A., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: Proceedings of the European Conference on Computer Vision, vol. I, pp. 311–326. Freiburg, Germany (1998)
11. Triggs, B., McLauchlan, P., Ri, H., Fitzgibbon, A.: Bundle adjustment—a modern synthesis. In: Vision Algorithms'99, pp. 298–372. in conjunction with ICCV'99, Kerkyra, Greece (1999)
12. Reid, I., Murray, D.: Active tracking of foveated feature clusrters using affine structure. In: International Journal of Computer Vision, pp. 41–60. Seattle, WA (1996)
13. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B., Cipolla, R. (eds.) Computer Vision – ECCV'96, Lecture Notes in Computer Science, vol. 1065, pp. 709–720. Springer, Berlin Heidelberg New York (1996)
14. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vis. **9**(2), 137–154 (1992)
15. Guilbert, N., Bartoli, A.: Batch recovery of multiple views with missing data using direct sparse solvers. In: British Machine Vision Conference. Norwich, UK (2003)
16. Triggs, B.: Linear projective reconstruction from matching tensors. Image Vis. Comput. **15**(8), 617–625 (1997)
17. Hoppe, H.: Surface reconstruction from unorganized points. Ph.D. thesis, Department of Computer Science and Engineering, University of Washington (1994)
18. Petitjean, S., Boyer, E.: Regular and non-regular point sets: Properties and reconstruction. Comput. Geom.—Theor. Appl. **19** (2001)
19. Manessis, A., Hilton, A., Palmer, P., McLauchlan, P., Shen, X.: Reconstruction of scene models from sparse 3d structure. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. Hilton Head, USA (2000)
20. Morris, D., Kanade, T.: Image-consistent surface triangulation. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. Hilton Head, USA (2000)
21. Huber, P.: Robust Statistics. Wiley, New York (1981)
22. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes in C. Cambridge University Press, Cambridge, UK (1988)
23. Strecha, C., Fransens, R., Gool, L.V.: Wide-baseline stereo from multiple views: a probabilistic account. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 552–559 Washington, DC (2004)

Tomas Rodriguez was born in Madrid in 1961. Bachelor in Physics and Master in Electronics by the Universidad Complutense de Madrid. He started his career in the private R&D sector in 1987, when he specialized in computer vision and parallel processing systems. In the early nineties he participated in the EVA project; one of the most outstanding traffic monitoring system of the time. For more than 10 years, he has been involved in international research projects within the ambit of EUREKA, ESPRIT, V and VI Framework programmes. During this time, he

coordinated eight international projects (CAMELOT, CITRONE, ON-LIVE, SAID, VISIRE, EVENTS, ITALES, HOLONICS) and acted as principal investigator in two additional ones (CITRUS and VICTORIA). Since the early days, he had the opportunity to collaborate with some of the most prestigious research institutions in Europe: Franhoufer Inst., INRIA, CNRS, University of Oxford, University of Lund, DFKI, Siemens C-Lab, Philips Research Labs, etc. Evaluator of R&D projects for the Spanish Ministry for Science and reviewer of international scientific journals, he is currently the R&D manager and coordinator for European projects at Eptron SA. His recent interests include: computer vision, real time software, industrial control, parallel processing, iTV, and mobile technologies, etc.