RECOVER3D: A Hybrid Multi-View System for 4D Reconstruction of Moving Actors

Laurent LUCAS[1a], Philippe SOUCHET[b], Muhannad ISMAËL[a], Olivier NOCENT[a], Cédric NIQUIN[b],
Céline LOSCOS[a], Ludovic BLACHE[a], Stéphanie PREVOST[a], Yannick REMION[a]
[a] CReSTIC-SIC, Université de Reims Champagne Ardenne, Reims, France;
[b] XD Production, Issy-les-Moulineau, France

http://dx.doi.org/10.15221/13.219

Abstract

4D multi-view reconstruction of moving actors has many applications in the entertainment industry and although studios providing such services become more accessible, efforts have to be done in order to improve the underlying technology and to produce high-quality 3D contents. The RECOVER3D project aim is to elaborate an integrated virtual video system for the broadcast and motion pictures markets. In particular, we present a hybrid acquisition system coupling mono and multiscopic video cameras where actor's performance is captured as 4D data set: a sequence of 3D volumes over time. The visual improvement of the software solutions being implemented relies on "silhouette-based" techniques and (multi-)stereovision, following several hybridization scenarios integrating GPU-based processing. Afterwards, we transform this sequence of independent 3D volumes in a unique dynamic mesh. Our approach is based on a motion estimation procedure. An adaptive signed volume distance function is used as the principal shape descriptor and an optical flow algorithm is adapted to the surface setting with a modification that minimizes the interference between unrelated surface regions.

Keywords: 3D body modeling system, entertainment, multi-view camera, multi-view stereovision and visual hull hybridization, animated model, dynamic mesh

1. Introduction

According to the increasing fragmentation of the TV audience due to the multiplication of channels and the appearance of new consumption behaviors (VOD, Internet ...), broadcasters and producers seek differentiated and quality contents, produced in optimal economic conditions. Among all paths considered in this regard, the use of 4D reconstruction studios are a sound alternative in the sense that it provides controlled environments generally based around a large room with uniform background equipped with multiple synchronized calibrated video cameras and appropriate illumination.

The main application areas of 4D studios are currently dedicated to computer games, movies, TV productions, interactive media and motion analysis. The "4D studios" term refers to the spatio-temporal domain where 3D reconstructions of non-rigid moving objects are calculated. Most of these systems require a temporal sequence of simultaneous image shots from multiple viewpoints in addition to suitable software solutions to produce a static set of 3D models at each time step.

The development of such studios is still very complex and leads to different technical solutions even if they share the same operation principle (e.g., MIT [1], MPI Informatik [2], INRIA [3], University of Surrey [4], Tsinghua University [5], Libero [6] or Digital Air [7]). Generally, these pipelines exploit the high redundancy of video sequences while leveraging the assumed continuity of movement. A review of recent trends in multi-view dynamic scenes' reconstruction from videos identifies two major building blocks required to, first, reconstruct 3D volumetric or surface data and second, track feature points over time. Our contribution also falls within this context.

This paper presents the project RECOVER3D (acronym for Real-time Environment for COmputational Video Editing and Rendering in 3D). Its purpose is to elaborate an integrated virtual video system for the broadcast and motion picture markets. The innovation brought by RECOVER3D aims at freeing the creation of video images from classic material constraints linked to multi-cameras shooting, thanks to a new "virtual cloning" system of actors and scenes based on smart 3D video capture, natively delivering depth information. This scanning system set up around the shooting area (see figure 2) and driven by software developed by the RECOVER3D consortium will generate the digital transcription of the scene in three dimensions, following a multi-view stereo-based visual hull paradigm.

The RECOVER3D consortium is based on the partnership between academic researchers in computer vision and industrial integrators and producers from the broadcast world. Together, we have described

[1] Corresponding author: laurent.lucas@univ-reims.fr ✆+33 326 918 452

and are implementing the prototype of what could be a suitable shooting facility for the industrial production of 4D images. The constraint is not only to improve the overall esthetical quality of the resulting models, but also to produce them in real time or in reduced post processing delay, providing a credible alternative to standard 2D studios.

This paper is structured as follows. In section 2, a brief overview of recent advances in 4D reconstruction and model-based tracking is given. In the next sections, the descriptions of the project architecture (section3), studio setting (section 4) and reconstruction components (section 5 and 6) are discussed. Some preliminary results are then presented in section 7 as well as the experience we have gained to date. Finally, our future plans are exposed before we conclude.

2. Previous work

This section gives a brief overview of the existing techniques for acquiring a 4D model of moving actors. There are two general classes of approaches. On the one hand, the 4D reconstruction step computes geometry without specific assumptions about the actors. On the other hand, the model-based tracking step search to determine over time an articulated motion model and its settings. Those two approaches are discussed below.

4D reconstruction literature abounds of relevant publications [8, 9] which usually extract shape from stereo [10, 11] and/or from silhouettes [12, 13] to reconstruct the scene independently for each time stamp. The first set, based on stereo extraction, is one of the classical methods for reconstructing 3D information from images. The advantage of this method is the ability to reconstruct surface details and concave regions. However, it is not able to handle textureless surfaces or repeated texture surfaces because the core computational process of this method depends on the information contained in the surface texture. The reconstruction of human shape by this technique is possible to lead to incorrect reconstruction or surface leaking, because most cloths have less texture. On the contrary, shape from silhouettes is very useful to real time applications and in multi-camera environments [14]. It is also able to deal with textureless and specular surfaces. However, the quality of the reconstruction from this technique is limited and the visual hull [15] cannot recover concave regions.

Both shape-from-stereo and shape-from-silhouette techniques are complementary to each other. This brought some combinations of them in order to exploit the benefits of both techniques. RECOVER 3D project is set right on this line. One of these possible combinations aims at building visual hull at first and then carving it according to the photo-consistency of each voxel [16, 17]. To achieve this, the scene volume is usually discretized. The goal is then to find the voxel which has the same color in each view, known as a photo-consistent voxel. The benefits of such an approach include the ability of modeling occlusion and possibility to implement in real time application. Unfortunately, discretizing space volumetrically leads to sampling and aliasing artifacts.

Li and Shirmacher [18] proposed to use the information from visual hull to improve stereovision-based 3D reconstruction system by reducing the depth map computation time and deleting outliers' information. However, this method suffers from the small amount of captured information (3 stereo pairs of low resolution cameras) which yields raw results. Furthermore, their local stereo matching choice negatively impacts these results. This is due to lack both of stereo redundancy and of global matching coherence insurance. Hilton and Starck [19] proposed another method to merge the two techniques by optimizing a mesh of a generic model of human shape to minimize an energy function that expresses the constraints of the visual hull and stereovision. The main drawback of this method is the restriction of the dynamic content of the scene captured because the object that can be modeled by this method is always a human shape. It is also irrelevant to non-rigid loose-fitting clothes such as a loose dress, whereas our project objectives are in particular to be able to reconstruct such cloth pieces. In section 5, we give some hints on our method to combine shape from stereo and silhouettes in order to overcome most of the limitations of the methods cited above while following some of their relevant ideas.

Once data are acquired and scene is reconstructed as a temporal succession of independently-generated 3D models, it becomes necessary to structure this models' set so that it can be used easily in any 3D design and creation commercial suites. This stage is essential to all post-production tasks especially if it is necessary to ensure geometric and colorimetric consistency of animated models.

Several methods have been proposed in order to address this major issue. Most of them rely on a model-based approach [20, 21]. An articulated mesh representing a generic human body is used as a template model. This mesh can also be obtained by a 3D scan of the actor. This model is moved according to a set of constraints (optical flow, silhouette matching …) which are extracted from the sequences. This animation is proceeded by use of a skeleton. Local deformations are then performed on the mesh in order to match non-rigid motions (clothes, hair …). The scene flow notion has been generalized by Vedula *et al.* [22]. This can be considered as the three-dimensional equivalent of the optical flow and it is computed by merging the optical flows from each point of view. It can be represented as a set of vectors which corresponds to the motion of 3D points in space. These vectors can also be computed from 3D reconstruction, by matching the vertices of the mesh according to curvature or color criteria. It is then used to establish a correspondence between each frame of the reconstruction and guarantee temporal consistency [23, 24, 25, 26]. Finally, we can cite the method proposed by Nobuhara *et al.* [27] which computes an equivalent of scene flow from a volumetric silhouette-based reconstruction and then uses it to animate a template mesh. The motion estimation is performed by matching the voxels of reconstructed discrete volumes. The mesh is obtained by a marching cube triangulation [28] of the first frame volume

In section 6, we focus on the enhancement of this hybrid reconstruction. Our volumetric reconstruction algorithm allows us to generate a volume at each frame of the multi-view video sequences. These volumes are then transformed into a 3D mesh sequence. The main problem is that there is no correspondence, in term of structure and topology, between the meshes at each frame. This makes it very difficult to use these meshes in post-production and 3D animation software. Our goal is to transform these mesh sequences in a single animated mesh.

3. Reconstruction pipeline

This section describes the components of our 4D reconstruction system. From a practical point of view, such studios must comply to several constraints. They must be in particular:

- easy to configure, set-up and run,
- fast to deliver the reconstructed models (ideally in real-time),
- and robust to track models evolving with non-extreme velocity.

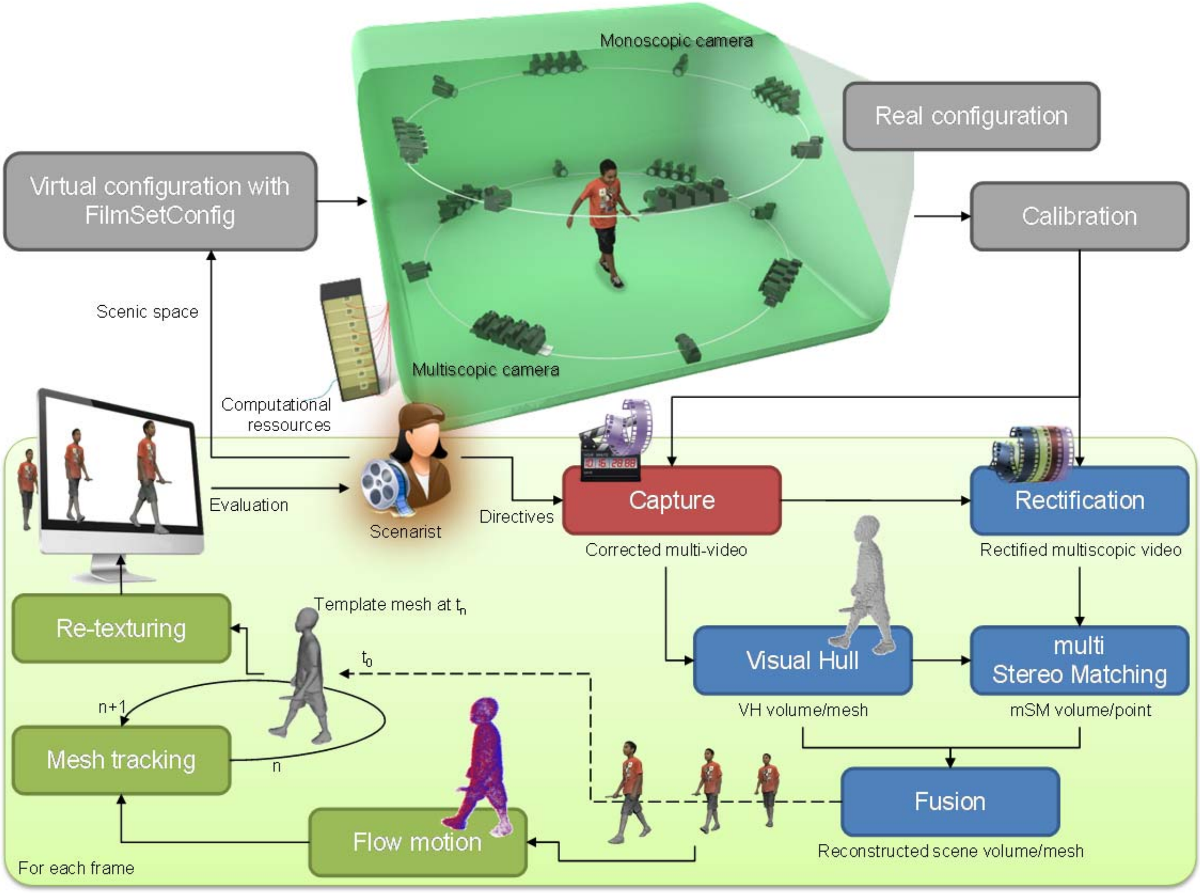


Figure 1. Process flow diagram

Figure 1 shows the modules and the functional blocks of our system. Challenges of such an implementation are synchronization and control of the flow of information in a production environment.

In practice, the main functional blocks are implemented in a number of IT components of a distributed system.

The first one "Studio setting" (gray modules) delivers a convenient studio which layout has been optimized according to the scenarist' needs concerning useful scenic space. It begins with an interactive virtual configuration yielding a convenient layout. Then the real configuration step places each camera in the studio according to this specification. Afterwards, the calibration process delivers extrinsic, intrinsic and deformation parameters for each camera. These parameters are mandatory for the incoming shooting and reconstruction processes.

The second block "4D shooting" (red module), is in charge of the scene shooting(s). It provides synchronized corrected videos from all cameras for each rush.

For each rush, the third functional block "Per frame 3D reconstruction" (blue modules), reconstructs instant 3D models of the scene, one at each time-stamp It combines a global visual hull with multiple (one per multiscopic capture unit set in the studio) local multi-stereo matching in pre-rectified images.

These produced 3D models sequences are then transmitted to the last block "4D model tracking" (green modules) in which motion flows are estimated in order to animate a template mesh.

This paper focuses on the first block (section 4) which is actually operational and the third (section 5) and fourth (section 6) which are both under development and, as such, have not yet totally reached our expectations. Nevertheless some of their preliminary results will be discussed in section 7.

4. Studio architecture and setting

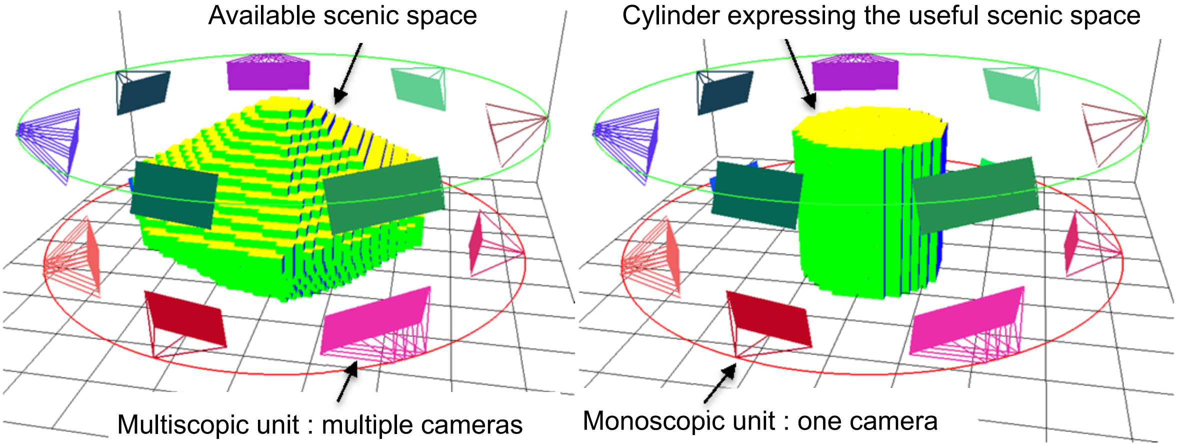
This section describes how we propose to optimize the specifications of a RECOVER3D studio, the hardware and calibration process that has been used to produce the first results shown in this paper.

4.1. Virtual studio specification

Our acquisition system consists of several tens of HD video cameras disposed either isolated as monoscopic capture units or grouped in multiscopic capture units. Those units and their camera components are distributed around the scene upon multiple horizontal coaxial circles placed at different heights in order to diversify the viewpoints. This core layout principle does not suffice as a number of parameters have to be tuned to achieve 3D scene reconstruction in optimum conditions: circles' height and radius; number of multiscopic and monoscopic units; number of cameras in a multiscopic unit; distribution of the units on the circles; main view axis of each unit. This tuning aims at maximizing the "available scenic space" defined as the area within which 3D reconstruction can be considered intrinsically robust (assuming no self-occlusion of the actors occur) due to sufficient numbers of units' covering. Physical tuning of an actual studio involves several iterative configuration tests each including camera positioning, calibration, test scene acquisition, raw reconstruction evaluation. This time consuming process bears several drawbacks: it consumes much man-power (and associated salary); it implies a long delay between the expression of a scenarist's needs and the availability of the studio for actual scene shooting; it prevents any other production in the studio during the tuning step.

These drawbacks would be greatly reduced by lowering the number of tuning iterations thanks to a prior knowledge of a convenient layout. To deliver this information, we developed a simulation software ("FilmSetConfig") of this tuning process. It first simulates the camera layout from several parameters set in a dedicated interface. Once a configuration is set, the simulation software qualitatively and quantitatively evaluates the scenic space within which a robust reconstruction may be expected with the chosen layout. This virtual evaluation replaces and surpasses physical tuning calibration – acquisition – reconstruction – evaluation pipeline. FilmSetConfig digitalizes the studio space in a voxel volume with a spatial resolution chosen via the interface and sets as "in available space" any voxel verifying some qualitative criteria. This outlined available space is then rendered for qualitative evaluation (see Figure 2.(a)). As we seek to use both shape from silhouette and shape from (multi-)stereo, the local criteria are chosen as voxel visibility from a minimum number of units either monoscopic or multiscopic (for silhouette) and from a mimimum number of multiscopic units with, for each of them, a visibility from enough of their constituting cameras (for multi-stereo). Those threshold numbers of units or cameras are freely chosen by the user in order to account for expected cases of self-occlusion. Any combination of the criteria can also be applied interactively in order to evaluate which criterion is most narrowing the available scene space for the chosen layout.

As one can see when comparing scenic space of (a) and (b) in Figure 2, available scenic spaces are polyhedra with some barely usable inner spaces near some vertices or edges. A scenarist' needs would more conveniently be expressed as "actors should be able to freely evolve in a hemisphere (or vertical circular or elliptic cylinder) of given minimal dimensions". To meet these requirements, once the available scenic space is known, FilmSetConfig quantifies the dimensions of the largest instance of a chosen shape that fits in the available space. This instance expresses the "useful scenic area" which is illustrated in Figure 2.(b) for a vertical circular cylinder choice.


Available scenic space
Cylinder expressing the useful scenic space
Multiscopic unit : multiple cameras
Monoscopic unit : one camera

(a) (b)

Figure 2. FilmSetConfig rendering: two rings capture units layout and associated scenic spaces

4.2. Hardware setting and calibration

The first RECOVER3D studio, considered as the prototype of a future product, has been installed in the premises of the industrial partner XD Productions, near Paris. It is a green chromakey studio of 100 square meters, which is 4.5 meter high. The results shown in this paper have been produced by 24 full HD cameras (1920x1080 pixels), at 25 frames per second, but the system has been designed to be scalable up to 40 cameras, recording 60 frames per second.

The set of cameras is divided into multiscopic units, each unit being constituted of 4 cameras linked to a capturing PC by HD-SDI cables. The cameras receive signals from external generators, ensuring the synchronized frame acquisition and unique time stamping of all capturing devices (Time Code).

The modeling quality we want to achieve puts a tremendous demand on the calibration procedure, which is used to estimate the intrinsic and extrinsic camera parameters, lens distortion coefficients, etc. Our calibration method is fully automatic, and a freely moving bright spot is the only calibration object [29]. The projections of the bright spot are detected independently in each camera. The points are validated through pairwise epipolar constraints. Projective motion and shape are computed via rank-4 factorization. Geometric constraints are applied and motions are stratified to Euclidean ones. A final bundle adjustment stage is then used to finely tune the positions of the scene points and the entire set of camera parameters (including the intrinsic ones and the distortion coefficients) in a single non-linear optimization [30].

The capturing units have also been designed to allow real time visualization of the animated volume reconstruction, despite the complex algorithms required. This involves the use of several multi-cored CPUs and latest generation graphic processing units (GPU). To address the new parallel calculation possibilities provided by these heterogeneous processing systems, we intensively use OpenCL and NVidia's CUDA computing languages in our software solution. We won't give more details here to ensure industrial secrecy.

5. Model reconstruction

The "per frame 3D reconstruction" functional block (blue modules in Figure 1) aims at computing, for each time-stamp, a textured geometric model of the moving actor(s) from the set of relevant simultaneous frame of each camera.

Real-time applications using wide-baseline camera set-ups often rely on visual hull (VH) computation. Despite their overall robustness and their almost closed-shape delivery, such methods are not able to carve concave zones of the actual models which yield inconvenient results that need to be refined by man-powered post-processing. Those concave areas are more accurately reconstructed by multi-stereo-matching (mSM) methods which, unfortunately, are prone to outliers' errors, rely on more compact camera set-ups yielding only very partial reconstruction, and do not yet reach real-time processing.

5.1. Functional block design

As described before, the project ambitions rather precise reconstruction in an automatic way by combining global VH and mSM from different viewpoints thanks to a dedicated studio layout design (see section 4). Each camera, either stand-alone as a monoscopic unit or part of a multiscopic unit, is actually implied in the VH process. Furthermore, each multiscopic unit provides an mSM capability that enables carving visible concave zones, roughly in the direction of the median view axis of its cameras. Relying on this specific studio layout, the reconstruction software under development thus combines methods of both VH and mSM classes. This hybrid method aims at retaining the robustness of VH which helps reducing outliers' errors of each mSM process while more conveniently reconstructing by mSM concave parts visible by one or more multiscopic units.

This assumption being set, multiple paths could be followed to achieve our goal. We chose to rely upon the VH process, altogether with its global calibration component, already available at XD Production and to qualitatively improve the mSM method previously developed at URCA while adapting it to the studio context with chromakey masks. As this mSM method better works on rectified images, a usual rectification process has been developed by XD Production.

5.2. Hints on chosen methods

The VH process implies a rather usual volumetric method. Masks' extraction for each camera is followed by a volume filling (in or out) based on mask reading for the projection of each cell on each camera. A cell which, at least for one camera, lies in its frustum and projects outside its mask is set as "out" in the volume. Otherwise, the cell is set as "in". Afterwards, a marching-cube process [28] on the VH volume delivers a mesh model of its surface.

The studied mSM method improves our previous one [31] while conserving its characteristics of directly multiscopic matching (altogether in every available image) ensuring the scene geometric consistency (depth maps coherence for each camera of the multiscopic unit) and encompassing occlusion detection. Furthermore, it's driven by a global optimization process that helps to avoid some outliers. The direct multiscopic matching searches for pixel homology among all available multiscopic coherent sets of pixels. These multiscopic coherent sets of pixels are composed of one pixel per image, altogether verifying epipolar constraints. These usual geometrical constraints insure that these pixels are projected by optical rays that actually intersect in the 3D scene space.

Classically (see Figure 3 for example and [32] for more details), in order to simplify the epipolar geometry to be used in order to compose those multiscopic coherent sets of pixels, we need a preprocessing step in order to rectify the pictures as if they were shot in a simplified decentered parallel layout. This "simplified geometry" classically relies on: optical centers both aligned and evenly distributed; coplanar sensors (and thus parallel optical axes); sensors' rows parallel to the optical centers' line; and, optionally, decentered sensors regions of interest (ROI, rectangular areas actually capturing the intended view). The last optional condition (decentered ROI) yields convergent view axes (lines passing through both the optical center and the ROI center).

The rectification process is practically achievable, thanks to calibration results, if actual cameras of the multiscopic unit have almost well-placed optical centers (alignment and even distribution) and roughly parallel or convergent optical axes.

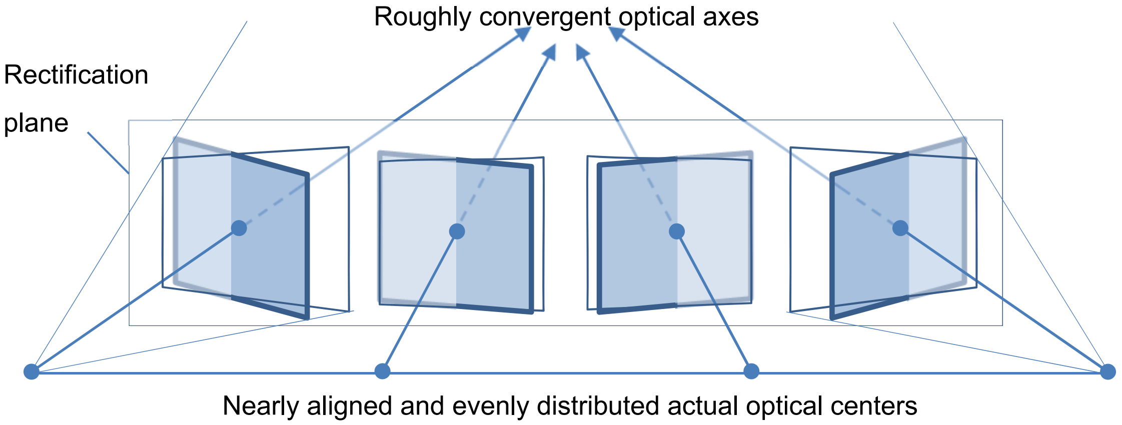

Roughly convergent optical axes
Rectification
plane
Nearly aligned and evenly distributed actual optical centers

Figure 3. Multiscopic rectification: captured optical beams are cut by a single plane parallel to the line holding the optical centers; this implies images skew (rectangular initial ROI becoming trapezes).

Thanks to this simplified layout, projections of a 3D point lie on the different rectified pictures on lines of same rank at abscissa translated one from the next (in the next image) by a fixed value, called disparity and decreasingly related to the depth of the projected 3D point.

The matching decisions are delivered by an iterative global energy optimization process evaluating various disparity assumptions for each pixel. The main energy term penalizes photometric dissimilarity of neighborhoods of the tested pixels. Others are related to global, geometrical inconsistency of the reconstructed points related to the chosen matches.

The occlusion detection is embedded within the photometric dissimilarity term. In fact this term tests incomplete multiscopic pixel sets when similarity drops for some image(s) and the implied pixels already have good matching with higher disparity (meaning lower depth).

We are currently working on the hybridization of VH with a single mSM. The main idea we follow consists in excluding of the matching process disparity assumptions associated with 3D points lying outside the visual hull. Calibration data and binary volumetric VH are of great usefulness for this scheme that removes any possible outlier outside the VH. While not complete this partial outlier exclusion both retains mSM result inside the VH (a necessary condition) and helps focusing on the assumption for actual disparity, thus increasing the overall matching accuracy. mSM usually delivers disparity or depth (multi-)maps. Our direct multiscopic matching naturally gives a structured set of 3D points easily derived from the matched multiscopic sets of pixels. In fact the coordinates of one of its pixels and the depth associated with its disparity give quick access to the 3D point associated with a pixel set. This structured set of 3D points can easily be expressed as a mesh by connecting 3D reconstructed points projecting as neighbors in the images.

Further work on this topic will imply the last module of the block - fusion of VH and mSM results as one reconstructed model either as a mesh (by local selection of mesh primitives from VH or SC sources) or as a volume (by removing voxels in VH volume that lie in front of the mSM mesh); integration of multiple mSM (from each multiscopic unit) with the global VH; and reducing mSM computation time as this global optimization process is yet far from real-time. As the mSM computing structure is compatible with massively parallel programming, GPGPU seems promising for this end.

6. Model tracking

After the reconstruction stage described in section 4, we obtain a sequence of discrete volumes which represent the character's pose at each video frame. In traditional multi-view reconstruction pipelines, these volumes are transformed in a sequence of 3D textured meshes which are successively loaded for the rendering of each frame. Our goal is to introduce a dynamic representation of the character to free ourselves from this static description of the scene. We expect our approach to be as generic as possible, allowing us to deal with various types of scenes (actors wearing loose clothes or close-up shots). We are looking for a single, temporally consistent, animated model according to the character's motion. Our method starts with computing a 3D motion flow between two consecutive frames. At this stage we work on the reconstructed volumes. In the next step we use these flows to animate a dynamic mesh model. The reconstructed mesh at the first frame is used as the initial template model. By deforming it at each frame according to the estimated flows, we deduce a character's animation.

6.1. Motion Flow Estimation

To compute the motion flow between two consecutive volumes V^n and V^{n+1}, we focus on the surface voxels (which belong to the surface of the object). They are characterized by an RGB color (coming from the video images) and a normal vector. For each surface voxel v_i^n in V^n, we search the surface voxel v_j^{n+1} of V^{n+1}, in a fixed neighborhood, which minimizes the sum of three terms:

- $\delta_{i,j}$: the Euclidean distance (which penalizes the matching of two voxels which could satisfy the other terms but could be considered too far from each other),
- $\varphi_{i,j}$: the angular difference between normals (which penalizes the matching of two voxels which belong to opposite surfaces),
- $\sigma_{i,j}$: the colorimetric difference (which compares the colors of two voxels and their neighborhood and favors the blocks which have close colors).

The distance function we want to minimize is:

$$D\left(v_i^n, v_j^{n+1}\right) = \omega_1 \delta_{i,j} + \omega_2 \varphi_{i,j} + \omega_2 \sigma_{i,j}$$

ω_1, ω_2 and ω_3 are weighting terms.

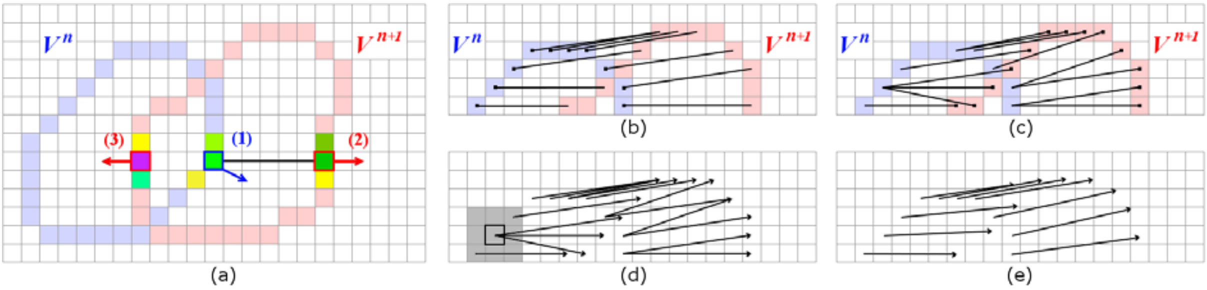


Figure 4. (a) Matching of two voxels, the voxel (1) matches better with the voxel (2) than with the voxel (3). (b) (c) forward and backward matching between the two volumes. (d) Gaussian filter (in grey) applied to the raw vectors fields. (e) final displacement field.

Each pair of matched voxels defines a vector that we add to a 3D vector field. This vector field is a data structure where each surface voxel is associated to one or several displacement vectors. We repeat this operation in a backward way (from V^{n+1} to V^n) to ensure that all surface voxels in each volume are matched with a surface voxel in the other frame. We obtain a raw vectors field which does not properly describe the motion between the two frames. Some voxels can also be associated with more than one motion vector. This is why we then apply a 3D Gaussian filter on this field to obtain a regular 3D motion field. For each surface voxel, we compute a single vector which is an average, weighted by Gaussian coefficients, of all the vectors in a fixed neighborhood. Thus, we obtain a smooth 3D motion field where each surface voxel is associated with a single motion vector.

6.2. Mesh Animation

In the second step, we immerse the template mesh in the motion flow and we apply to each vertex the translation defined by the corresponding vector. The result is too irregular to be used. So we then apply a regularization algorithm to obtain a regular mesh which corresponds to the pose defined by the visual hull. We consider the mesh as a mechanical mass-spring system. Each vertex is submitted to a set of forces including: the spring force applied by the incident edges (tends to regularize the vertices distribution), the smoothing force (which tends to smooth the surface of the mesh) and the matching force (uses a signed distance field computed from the volume to push the vertices in the direction of the object surface). We use a modified Euler integration method to solve this system: for each vertex, we use a semi-implicit integration method using a fixed neighborhood. This operation is performed on each vertex, corresponding to a single global iteration. We apply as many global iterations as necessary.

7. Results & discussion

As previously stated, this project is ongoing and the below presented reconstructions are preliminary results of methods discussed in sections 5 and 6. Unfortunately, to this date, these studies have been following parallel ways using available suitable media. Section 6 modules have thus benefited of an early shooting of a boy “Kevin”, with 16 monoscopic units, which had been processed by “visual hull” only. Later on, (rather recently in fact) the layout and calibration of multiscopic units has been achieved, yielding another sequence “Ludo” captured with 16 monoscopic units and 1 multiscopic unit composed of 4 cameras. This sequence is mainly used for section 5 processes.

Let’s now present some 3D reconstruction results for one time stamp of “Ludo” rush. Figure 5 shows media used for multi-stereo matching avoiding “out of multiscopic masks” assumptions. Images and associated masks are rectified. Figure 5 expose their parts used in results exhibited in Figure 6. These results clearly show that mSM yields outliers in front of the model (from the multiscopic viewpoint but outside the visual hull (see bottom right of Figure 6 in front of the head and left of the right wrist). Those will soon be avoided by our ongoing hybridization process (see 5.2). Nevertheless, these preliminary results demonstrate the usefulness of mSM in VH correction with more detailed depth in the face and neck areas of the model.

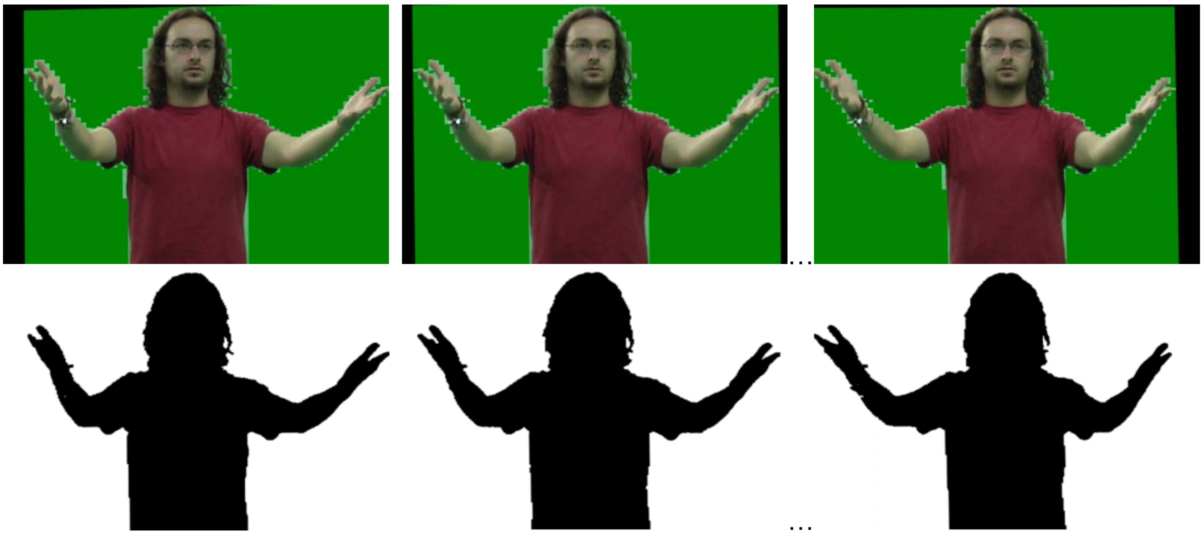


Figure 5. Upper parts of 3 images from the rectified multiscopic set of same time stamp in “Ludo” rush (top) and associated masks (bottom)

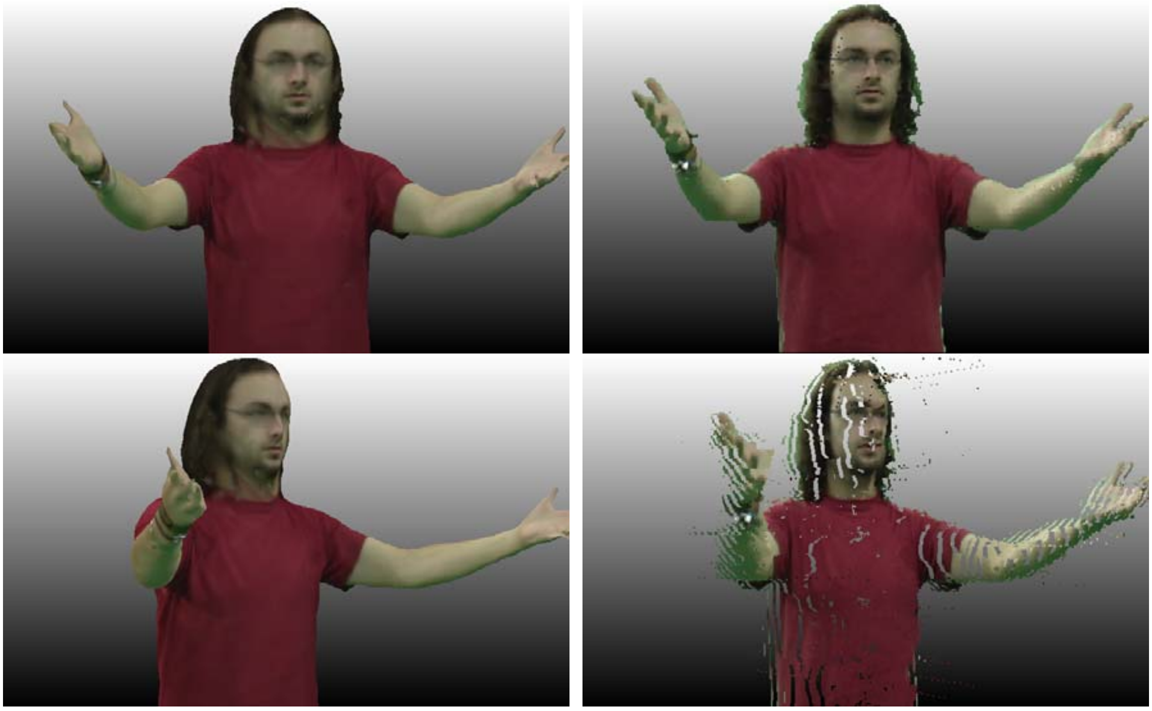


Figure 6. Comparison of textured visual hull (left) and multi-stereo matching (right) results for one time stamp of “Ludo” rush ; front (top) and lateral (bottom) views of the models ; VH mesh is expressed in metric global scene space while mSM points are expressed in local image (pixel, disparity) space with, thus, artificial depth distortion. One can see at bottom right that reconstructed points lie on constant disparity planes.

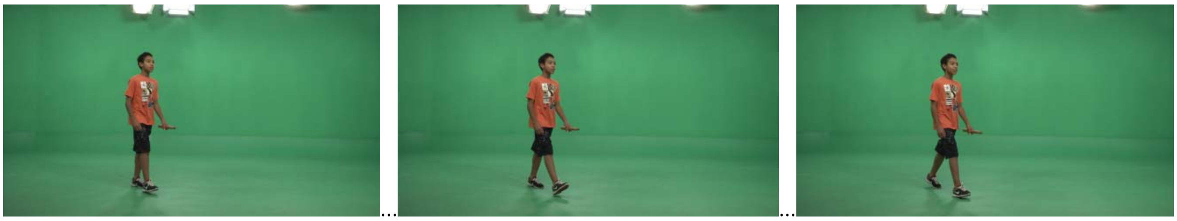


Figure 7. Three different images of same time stamp in “Kevin” rush

Our model tracking method allows us to transform a reconstructed volume sequence into an animated, temporally coherent, mesh. Several reconstructed volume sequences (see Figure 7) containing about thirty frames have been used to test the functional motion flow block. Due to the final regularization step, we obtain a satisfying motion field where each surface voxel is associated to a displacement vector. This method works on generic datasets, whatever the shape of the reconstructed object or character. However, the changes in the topology of objects that could appear during the sequences are not well supported and may result in inconsistent motions.

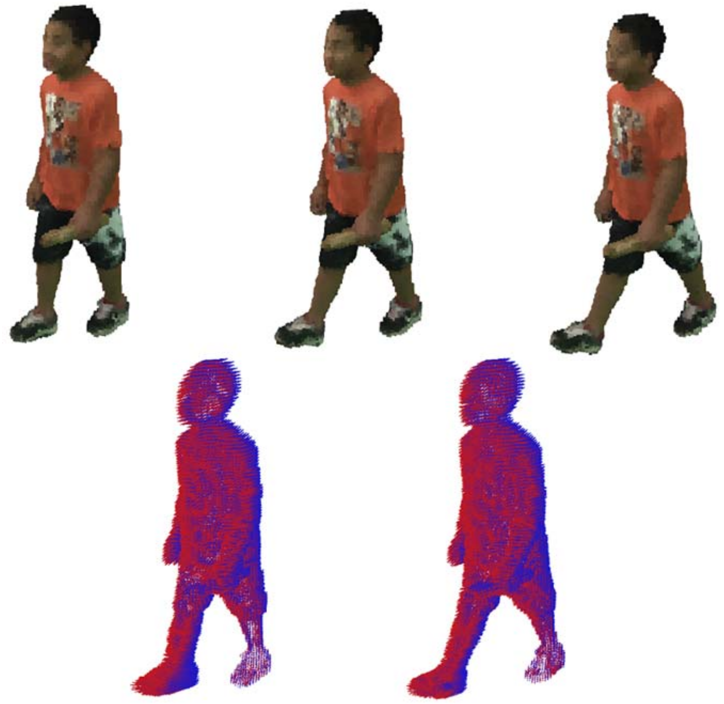


Figure 8. Model tracking results for "Kevin" rush
Top: three successive frames of the volumes sequence. Bottom: two intermediate motions flows.

The mesh animation process is also very sensitive to the changes in topology. It also became unstable when the motions between two frames are too wide. These problems can be partly overcome by increasing the number of iterations during the mesh regularization step.

Another issue is the number of parameters which have to be fixed by the user (weighting coefficients for the terms of voxel matching and mesh regularization, Gaussian filter radius, and number of iterations) and that may not be robust for all the sequence. These problems prevent us from computing efficiently an animation from long and complex sequences. The last limitation is the computing time, which could be reduced by the use of GPGPU technologies.

8. Conclusion

In this article, we presented the state of play of the project RECOVER3D. Based on a system coupling mono and multiscopic video camera units, RECOVER3D should ultimately allow us both to reconstruct and animate a 3D scene with a unique dynamic mesh. The initial results demonstrate of the validity of the approach even if there is still a lot of work to improve it. To date and for some specific situations (only with the VH support), the intensive use of hardware acceleration satisfies the real time constraints for reasonable sizes of grids of voxels and a reasonable number of viewpoints. On such context, the RECOVER3D consortium has achieved real time (25 images per second) for voxel grids of 1283 covered by 24 RGB high definition cameras (1920x1080 pixels). This GPGPU intensive use could also greatly reduce computation time of stereo matching and model tracking processes, yielding, finally, an efficient shooting facility for economically realistic industrial production of qualitative 4D avatars of moving actors.

Acknowledgments

This work is supported by the French National Fund for a Digital Society as part of the call “Technologies for scanning and recovery of cultural, scientific and educational contents”, in the services, uses and innovating contents.

References

1. MIT Computer Graphics Group, (accessed 2013): “Dynamic Shape Capture and Articulated Shape Animation”, http://people.csail.mit.edu/drdaniel/
2. MPI Informatik, Graphics Vision and Video Group, (accessed 2013): http://gvv.mpi-inf.mpg.de/GVV_Projects.html
3. INRIA Morpheo Team, (accessed 2013): ”Capture and Analysis of Shapes in Motion”, http://morpheo.inrialpes.fr/
4. University of Surrey, Centre for Vision, Speech and Signal Processing, (accessed 2013): ”SurfCap: Surface Motion Capture", http://kahlan.eps.surrey.ac.uk/Personal/AdrianHilton/Research.html
5. University of Tsinghua, Broadband Network & Digital Media Lab, (accessed 2013): ”Multi-camera Multi-lighting Dome”, http://media.au.tsinghua.edu.cn/dome.jsp
6. VizRT, (accessed 2013): “Libero”, http://www.vizrt.com/products/viz_libero/
7. Digital Air, (accessed 2013): http://www.digitalair.com/
8. Moeslund T. B., Hilton A., Kruger V. and Sigal L. (2011): “Visual Analysis of Humans - Looking at People”, Springer.
9. Szeliski R, (2010): “Computer Vision: Algorithms and Applications”, Springer,1st edition.
10. Birchfield S. and Tomasi C.(1998): “Depth discontinuities by pixel-to-pixel stereo”, proceedings ICCV 1998, Bombay, India, pp 1073-1080.
11. Kang S. B., Szeliski R., Chai J., (2001): “Handling Occlusions in Dense Multi-view Stereo”, proceedings CVPR 2001, vol. 1, Kauai, USA, pp 103-110.
12. Chien, C.H., and Aggarwal,J,K, (1986): “Volume/surface Octrees for the Representation of Three-Dimensional Objects”, CVGIP, vol. 36(1), pp.100-113.
13. Steinbach, E. G.; Girod, B.; Eisert, P. & Betz A., (2000): “3-D Reconstruction of Real-World Objects Using Extended Voxels”, proceedings ICIP 2000, vol. 1, Vancouver, Canada, pp. 569-572.
14. Cheung G., Kanade T., Bouguet J.Y., and Holler M., (2000): “A real time system for robust 3D voxel reconstruction of human motions”, proceedings CVPR 2000, vol. 2, pp 714-720.
15. Laurentini, A. (1994): “The visual hull concept for silhouette-based image understanding”, IEEE Transaction on PAMI. Vol. 16(2), pp 150-162.
16. Seitz S. M. and Dyer C. R., (1999): “Photorealistic scene reconstruction by voxel coloring”, International Journal of Computer Vision, 35(2), pp 151–173.
17. Kutulakos K. N. and Seitz S. M., (2000): “A theory of shape by space carving”, International Journal of Computer Vision, 38(3), pp 199–218.
18. Ming Li, Schirmacher H., Magnor M., and Siedel H.-P., (2002): “Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes”, proceedings of IEEE Workshop on Multimedia Signal Processing 2002, St. Thomas, US Virgin Islands, pp 9-12.
19. Hilton A. and Starck J., (2004): “Multiple view reconstruction of people”, proceedings of 3DPVT 2004, Thessaloniki, Greece, pp 357-364.
20. Vlasic D., Baran I., Matusik W., Popović J., (2008): “Articulated mesh animation from multi-view silhouettes”, proceedings SIGGRAPH’08, Los Angeles, USA, pp 97:1--97:9.
21. Liu Y., Gall J., Stoll C., Dai Q., Seidel H-P., Theobalt C., (2013): “Markerless Motion Capture of Multiple Characters Using Multi-view Image Segmentation”, IEEE Transaction on PAMI. Vol. 34(11), pp 2720-2735..
22. Vedula S., Rander P., Collins R., Kanade T., (1999), “Three-dimensional scene flow”, proceedings ICCV 1999, Kerkyra, Corfu, Greece, vol. 2, pp. 722-729.
23. Varanasi K., Zaharescu A., Boyer E., Horaud, R., (2008): “Temporal surface tracking using mesh evolution”, proceedings ECCV '08, Marseille, France, LNCS, vol 5303, part 2, pp. 30-43.

24. Petit B., Letouzey A., Boyer E., Franco J.-S., (2001): “Surface flow from visual cues”, proceedings Vision, Modeling and Visualization Workshop - VMV 2011, Berlin, Germany, pp. 1-8.
25. Tung T., Matsuyama T., (2010): “Dynamic surface matching by geodesic mapping for 3D animation transfer”, proceedings CVPR 2010, San Francisco, USA, pp. 1402-1409.
26. Starck J., Hilton A., (2007): “Correspondence labelling for wide-timeframe free-form surface matching”, Proceedings ICCV 2007, Rio de Janeiro, Brazil, pp 1-8.
27. Nobuhara S.; Matsuyama T., (2003): "Dynamic 3D shape from multi-viewpoint images using deformable mesh model", proceedings ISPA 2003, vol.1, Roma, Italy, pp.192-197.
28. Lorensen W.E. and Cline H.E., (1987): “Marching cubes: A high resolution 3D surface construction algorithm”, proceedings SIGGRAPH 1987, volume 21, Anaheim, USA, pp 163–169.
29. Svoboda T., Martinec D., and Pajdla T., (2005): “A convenient multi-camera self-calibration for virtual environments”, *PRESENCE: Teleoperators and Virtual Environments*, 14(4), pp 407-422.
30. Lourakis M, Argyros A (2008, accessed 2013): “SBA: A generic sparse bundle adjustment C/C++ package based on the Levenberg Marquardt algorithm”, http://users.ics.forth.gr/~lourakis/sba/
31. Niquin C., Prévost S., Remion Y., (2010): “An occlusion approach with consistency constraint for multiscopic depth extraction”, International Journal of Digital Multimedia Broadcasting, vol. 2010, Article ID 857160, 8 pages.
32. Hartley R., Zisserman A., (2003): Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press.