Interactive walkthroughs using "morphable 3D-mosaics"[*]

Nikos Komodakis[†], George Pagonis[†] and George Tziritas[†]
Computer Science Department[§], University of Crete

Abstract

This paper presents a hybrid (geometry- & image-based) technique suitable for providing interactive walkthroughs of large, complex outdoor scenes. Motion is restricted along a smooth predefined path and the input to the system is a sparse set of stereoscopic views at certain points (key-positions) along that path (one view per position). An approximate local 3D model is constructed from each view, capable of capturing photometric and geometric properties of the scene only locally. Then during the rendering process, a continuous morphing (both photometric & geometric) takes place between successive local 3D models, using what we call a "morphable 3D-model". The morphing proceeds in a physically-valid way. For this reason, a wide-baseline image matching technique is proposed, handling cases where the wide baseline between the two images is mainly due to a looming of the camera.

Our system can be extended in the event of multiple stereoscopic views (and therefore multiple local models) per key-position of the path (related by a camera rotation). In that case one local 3D-mosaic (per key-position) is constructed comprising all local 3D models therein and a "morphable 3D-mosaic" is used during the rendering process. A partial-differential equation is adopted to handle the problem of geometric consistency of each 3D-mosaic.

1. Introduction

One research problem of computer graphics that has attracted a lot of attention during the last years is the creation of modeling and rendering systems capable to provide photorealistic & interactive walkthroughs of complex, real-world environments. Two are the main approaches proposed so far in the computer vision literature for this purpose. On one hand (according to the purely geometric approach), a full 3D geometric model of the scene needs to be constructed. On the other hand, image-based rendering (IBR) methods skip the geometric modeling part and attempt to create novel views by appropriate resampling of a set of images.

[*]This work was supported by DHX project (IST-2001-33476)
[†]e-mail: {komod,pagonis,tziritas}@csd.uoc.gr
[§]P.O. Box 2208, 71409, Heraklion, Greece

While a lot of research has been done regarding small scale scenes, there are only few examples of works dealing with large scale environments. The current work is such an example and follows a hybrid (geometry- & image-based) approach, capable of providing interactive walkthroughs of large-scale, complex outdoor environments. For this purpose, a new data representation of a 3D scene is proposed consisting of a set of morphable (both geometrically & photometrically) 3D models.

The main assumption is that during the walkthrough, the user motion takes place along a (relatively) smooth, predefined path of the environment. The input to our system is then a sparse set of stereoscopic views captured at certain points (*"key-positions"*) along that path (one view per key-position). A series of *local 3D models* are then constructed, one for each stereoscopic view, capturing the photometric & geometric properties of the scene locally. These local models need to contain only an approximate representation of the scene geometry. During the transition between any two successive key-positions pos_1, pos_2 along the path (with corresponding local models L_1 and L_2), a *"morphable 3D-model"* L_{morph} is displayed by the rendering process. At point pos_1 this model coincides with L_1 and while we are approaching pos_2, it is gradually transformed (both photometrically and geometrically) into L_2 coinciding with the latter upon reaching key-position pos_2. The morphing proceeds in a physically-valid way and for this reason, a wide-baseline image matching technique is proposed handling instances where the dominant differences in the images appearance are due to a looming of the camera. Therefore, during the rendering process a continuous morphing (each time between successive local 3D models) takes place.

Our system can be extended to handle the case of having multiple stereoscopic views per key-position, which are related by a pure rotation of the stereoscopic camera at that position of the path. In that case, a *3D-mosaic* per key-position needs to be constructed in addition. This 3D-mosaic comprises the multiple local models coming from the corresponding stereoscopic views at that position.

The main advantages of our approach are:

- no global 3D model of the environment needs to be assembled (a cumbersome process for large scale scenes)

- scalability to large environments since at any time only one "morphable 3D-model" is displayed. In addition, we make use of a rendering path that is highly optimized in modern 3D graphics hardware
- being an image-based method, it can reproduce the photorealistic richness of a scene
- ease of data acquisition (e.g. collecting data for a path over 100 meters long took us only about 20 minutes) as well as an almost automated processing of these data

2. Related work

There have been numerous works on geometry-based modeling of real world scenes. For example Koch [8] uses stereoscopic image sequences for reconstructing a global 3D model of the environment. Another example is Pollefey et al.'s work on 3D reconstruction from hand-held cameras [12]. Debevec et. al [4] propose a hybrid (geometry- and image-based approach) for modeling architecture. However, they need a basic geometric model of the whole scene as well (recovered interactively). Then they use view dependent texture mapping to enhance appearance.

Lightfield [9] and Lumigraph [6] are two popular image-based rendering methods, but they require a large number of input images and so they are mainly used for small scale scenes. In "plenoptic modeling" [11], a warp operation is introduced that maps panoramic images (along with disparity) to any desired view. In [5], an image-based technique is proposed by which an end-user can create walkthroughs from a sequence of photographs. Finally, in the "sea of images" approach [1], a dense set of omnidirectional images (with image spacing $\approx$ 1.5 inches) are captured for creating interactive walkthroughs of large, indoor environments.

3. Overview of the modeling pipeline

We will first consider the simpler case of having one stereoscopic view per key-position of the path. Prior to capture, a calibration of the stereoscopic camera takes place [2]. The left & right camera calibration matrices (denoted hereafter by K_{left} & K_{right}) as well as the external geometry of the stereo-rig are estimated. Also lens distortion is removed. The main stages of the modeling pipeline are:

1. Local 3D models construction (section 4). A photometric & geometric representation of the scene near each key-position of the path is constructed. The geometric part of a local model must be just an approximation of the true scene geometry.
2. Relative pose estimation between successive local 3D models (section 5). Only a coarse estimate of the relative pose is needed since this will not be used for an exact registration of the local models but merely for the morphing procedure that takes place later.
3. Estimation of morphing between successive local 3D models along the path (section 6).

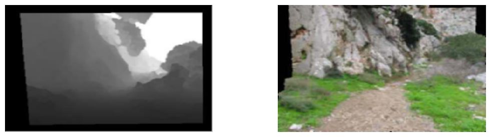

Fig. 1: (a) Depth map Z_0 of a local model (black pixels do not belong to its valid region dom_0). **(b)** A rendered view of the local model using an underlying triangle mesh

4. Local 3D models construction

For each stereoscopic image pair, a 3D model describing the scene locally (i.e. as seen from the camera viewpoint) must be produced during this stage. To this end, a stereo matching procedure is applied to the left & right images (I_{left} & I_{right}), so that disparity is estimated for all points inside a selected image region dom_0 of I_{left} (we refer the reader to [13] for a review on stereo matching). A Gaussian as well as a median filter is further applied to the disparity map for smoothing and removing spurious disparity values respectively. Using this disparity map (and the calibration matrices of the cameras) a 3D reconstruction takes place and thus the maps X_0, Y_0 and Z_0 are produced (see Fig. 1(a)), containing respectively x, y and z coordinates of the reconstructed points with respect to the 3D coordinate system of the left camera.

The set $L_0 = (X_0, Y_0, Z_0, I_{left}, dom_0)$ consisting of the images X_0, Y_0, Z_0 (the geometric-maps), the image region dom_0 (valid domain of geometric-maps) and the image I_{left} (the photometric map) makes up what we call a *"local model"* L_0. Hereafter that term will implicitly refer to such a set of elements. By applying a 2D triangulation on the image grid of a local model, a textured 3D triangle mesh can be produced. The 3D coordinates of triangle vertices are obtained from the underlying geometric maps while texture is obtained from I_{left} and mapped onto the mesh (see Fig. 1(b)). It should be noted that the geometric maps of a local model are expected to contain only an approximation of the scene's corresponding geometric model.

5. Relative pose estimation between successive local models

Let L_k=(X_k,Y_k,Z_k, I_k, dom_k) and L_{k+1}=(X_{k+1},Y_{k+1}, Z_{k+1}, I_{k+1}, dom_{k+1}) be 2 successive local models along the path. For their relative pose estimation, we need to extract a set of point matches (p_i, q_i) between the left images I_k, I_{k+1} of models L_k, L_{k+1} respectively (see section 5.1). Assuming that such a set of matches already exists, then the pose estimation can proceed as follows: the 3D points of L_k corresponding to p_i are $P_i = (X_k(p_i), Y_k(p_i), Z_k(p_i))$ and so the reprojections of p_i on image I_{k+1} are: $p'_i = K_{left}(R \cdot P_i + T) \in \mathbb{P}^2$, where R (a 3×3 orthonormal matrix) and T (a 3D vector) represent the unknown rotation and translation respectively.

So the pose estimation can be achieved by minimizing the following reprojection error: $\sum_i dist(q_i, p'_i)^2$, where $dist$ denotes euclidean image distance. For this purpose, an iterative constrained-minimization algorithm may be applied with rotation represented internally by a quaternion q ($\|q\|$=1). The *essential matrix* (also computable by the help of the matches (p_i, q_i) and K_{left}, K_{right}) can be used to provide an initial estimate [7] for the iterative algorithm.

5.1. Wide-baseline feature matching under camera looming

Therefore the pose estimation problem is reduced to that of extracting a sparse set of correspondences between I_k, I_{k+1}. A usual method for tackling the latter problem is the following: first a set of interest-points in I_k are extracted (using an interest-point detector). Then for each interest-point, say p, a set of candidate points CAND$_p$ inside a large rectangular region SEARCH$_p$ of I_{k+1} are examined and the best one is selected according to a similarity measure. Usually the candidate points are extracted by applying an interest-point detector to region SEARCH$_p$ as well.

However unlike left/right images of a stereoscopic view, I_k and I_{k+1} are separated by a wide baseline. Simple measures like correlation have been proved extremely inefficient in such cases. Assuming a smooth predefined path (and therefore a smooth change in orientation between I_k, I_{k+1}), it is safe to assume that the main difference at an object's appearance in images I_k and I_{k+1}, comes from the forward camera motion along the Z axis (looming). The idea for extracting valid correspondences is then based on the following observation: the dominant effect of an object being closer to the camera in image I_{k+1} is that its image region in I_{k+1} appears scaled by a certain scale factor s>1. That is, if $p \in I_k, q \in I_{k+1}$ are corresponding pixels: $I_{k+1}(sq) \approx I_k(p)$. So an image patch of I_k at p should look similar to an image patch of an appropriately rescaled (by s^{-1}) version of I_{k+1}.

Of course, the scale factor s varies across the image. Therefore the following strategy, for extracting reliable matches, can be applied:

1. Quantize the scale space of s to a discrete set of values $S = \{s_j\}_{j=0}^{n}$, where $1 = s_0 < s_1 < ... < s_n$
2. Rescale I_{k+1} by the inverse scale s_j^{-1} for all $s_j \in S$ to get rescaled images I_{k+1,s_j}

   For any $q \in I_{k+1}$, $p \in I_k$, let us denote by $\overline{I_{k+1,s_j}(q)}$ a (small) fixed-size patch around the projection of q on I_{k+1,s_j} and by $\overline{I_k(p)}$ an equal-size patch of I_k at p.
3. Given any point $p \in I_k$ and its set of candidate points CAND$_p = \{q_i\}$ in I_{k+1}, use correlation to find among the patches at any q_i and across any scale s_j, the one most similar to the patch of I_k at p:

   $$(q', s') = \arg\max_{q_i, s_j} corr(\ \overline{I_{k+1,s_j}(q_i)},\ \overline{I_k(p)}\ )$$

   This way, apart from a matching point $q' \in I_{k+1}$, a scale estimate s' is provided for point p as well .

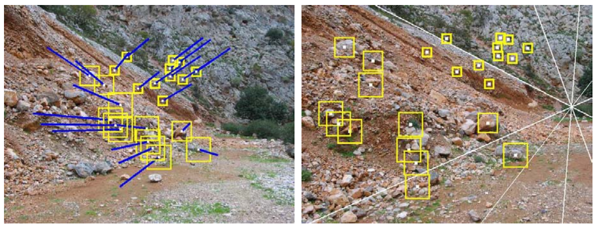

Fig. 2: **(a)** Image I_k along with computed optical flow vectors (blue segments) for all points marked white. **(b)** Image I_{k+1} along with matching points (also marked white) for all marked points of (a). A few epipolar lines are also shown. In both images, the yellow square around a point is analogous to the point's estimated scale factor (10 scales $S = \{1, 0.9^{-1}, ..., 0.1^{-1}\}$ have been used).

The above strategy has been proved very effective, giving a high percentage of exact matches even in cases with very large looming. Such an example can be seen in Fig. 2 wherein the images baseline is $\approx$ 15 meters, resulting in scale factors of size ≈ 2.5 for certain image regions. Even if we set as candidate points CAND$_p$ of a point p, all points inside SEARCH$_p$ in the other image (and not only detected interest-points therein), the above procedure still picks the right matches in most cases. The results in Fig. 2 have been produced this way.

6. Morphing estimation between successive local models along the path

At the current stage of the modeling pipeline, a series of approximate local 3D models (along with approximate estimates of the relative pose between every successive two) are available to us. Let $L_k = (X_k, Y_k, Z_k, I_k, dom_k)$, $L_{k+1} = (X_{k+1}, Y_{k+1}, Z_{k+1}, I_{k+1}, dom_{k+1})$ be such a pair of successive local models and pos_k, pos_{k+1} their corresponding key-positions on the path. By making use of the approximate pose estimate between L_k and L_{k+1}, we will assume hereafter that the 3D vertices of both models are expressed in a common 3D coordinate system.

Rather than trying to create a consistent global model by combining all local ones (a rather tedious task requiring among others high quality geometry and pose estimation) we will instead follow a different approach which is based on the following observation: near path point pos_k, model L_k is ideal for representing the surrounding scene. On the other hand, as we move forward along the path approaching key-position of the next model L_{k+1}, the photometric and geometric properties of the environment are much better captured by the latter model. (For example compare the fine details of the rocks that are revealed in Fig. 2(b) and are not visible in Fig. 2(a)). So during transition from pos_k to pos_{k+1}, we will try to gradually morph model L_k into a new destination model, which should coincide with L_{k+1} upon reaching point pos_{k+1}. (In fact, only part of this destination model can coincide with L_{k+1} since in general L_k, L_{k+1}

will not represent exactly the same part of the scene). This morphing should be geometric as well as photometric (the latter wherever possible) and should proceed in a physically valid way. For this reason, we will use what we call a *"morphable 3D-model"*:

$$L_{morph} = L_k \cup (X_{dst}, Y_{dst}, Z_{dst}, I_{dst})$$

In addition to including the elements of L_k, L_{morph} also consists of maps $X_{dst}, Y_{dst}, Z_{dst}$ and map I_{dst} containing respectively the destination 3D vertices and destination color values for all points of L_k. At any time during the rendering process, the 3D coordinates $vert_{ij}$ and color col_{ij} of the vertex of L_{morph} at point (i, j) will then be:

$$vert_{ij} = \begin{bmatrix} (1-m)X_k(i,j) + mX_{dst}(i,j) \\ (1-m)Y_k(i,j) + mY_{dst}(i,j) \\ (1-m)Z_k(i,j) + mZ_{dst}(i,j) \end{bmatrix} \quad (1)$$

$$col_{ij} = (1-m)I_k(i,j) + mI_{dst}(i,j) \quad (2)$$

where m is a parameter determining the amount of morphing ($m=0$ at pos_k, $m=1$ at pos_{k+1} and $0<m<1$ in between). Specifying therefore L_{morph} amounts to filling-in the values of the destination maps $\{X, Y, Z, I\}_{dst}$ for each point $p \in dom_k$.

For this purpose, a 2-step procedure will be followed that depends on whether point p has a physically corresponding point in L_{k+1} or not:

1. Let Ψ be that subset of region $dom_k \subseteq I_k$, consisting only of those L_k points that have physically corresponding points in model L_{k+1} and let $u_{k\to k+1}$ be a function which maps these points to their counterparts in the I_{k+1} image. (Region Ψ represents that part of the scene which is common to both models L_k, L_{k+1}). Since model L_k (after morphing) should coincide with L_{k+1}, it must then holds:

$$\begin{bmatrix} X_{dst}(p) \\ Y_{dst}(p) \\ Z_{dst}(p) \\ I_{dst}(p) \end{bmatrix} = \begin{bmatrix} X_{k+1}(u_{k\to k+1}(p)) \\ Y_{k+1}(u_{k\to k+1}(p)) \\ Z_{k+1}(u_{k\to k+1}(p)) \\ I_{k+1}(u_{k\to k+1}(p)) \end{bmatrix} \forall p \in \Psi \quad (3)$$

   Points of region Ψ are therefore transformed both photometrically and geometrically.

2. The rest of the points (that is points in $\bar{\Psi}=dom_k \backslash \Psi$) do not have counterparts in model L_{k+1}. So these points will retain their color value (from model L_k) at the destination maps and no photometric morphing will take place:

$$I_{dst}(p) = I_k(p), \; \forall p \in \bar{\Psi} \quad (4)$$

   But we still need to apply geometric morphing to those points so that no distortion/discontinuity in the 3D structure is observed during transition from pos_k to pos_{k+1}. Therefore we still need to fill-in the destination 3D coordinates for all points in $\bar{\Psi}$.

The 2 important remaining issues (which also constitute the core of the morphing procedure) are:

1. How to compute the mapping $u_{k\to k+1}$. This is equivalent to estimating a 2D optical flow field between the left images I_k and I_{k+1}.

2. And how to obtain the values of the destination geometric-maps at the points inside region $\bar{\Psi}$, needed for the geometric morphing therein. Both issues will be the subject of the two sections that follow.

6.1. Estimating optical flow between wide-baseline images I_k and I_{k+1}

In general, obtaining a reliable, relatively-dense optical flow field between wide-baseline images like I_k and I_{k+1} is a particularly difficult problem. Without additional input, usually only a sparse set of optical flow vectors can be obtained in the best case. The basic problems being that:

1. For every point in I_k , a large region of I_{k+1} image has to be searched for obtaining a corresponding point. This way the chance of an erroneous optical flow vector increases significantly (as well as the computational cost)

2. Simple measures (like correlation) are very inefficient for comparing pixel blocks between wide-baseline images

3. Even if both of the above problems are solved, optical flow estimation is inherently an ill-posed problem and additional assumptions are needed

For dealing with the first problem, we will make use of the underlying geometric maps X_k,Y_k,Z_k of model L_k as well as the relative pose between I_k and I_{k+1}. By using these quantites, we can theoretically reproject any point, say p, of I_k onto image I_{k+1}. In practice since all of the above quantities are estimated only approximately, this permits us just to restrict the searching over a smaller region R_p around the reprojection point. The search region can be restricted further by taking the intersection of R_p with a small zone around the epipolar line corresponding to p. In addition, since we are interested in searching only for points of I_{k+1} that belong to dom_{k+1} (this is where L_{k+1} is defined), the final search region SEARCH$_p$ of p will be $R_p \cap dom_{k+1}$. If SEARCH$_p$ is empty, then no optical flow vector will be estimated and point p will be considered as not belonging to region Ψ.

For dealing with the second problem, we will use a technique similar to the one described in section 5.1 for getting a sparse set of correspondences. As already stated therein, the dominant effect due to a looming of the camera is that pixel neighborhoods in image I_{k+1} are scaled by a factor varying across the image. The solution proposed therein

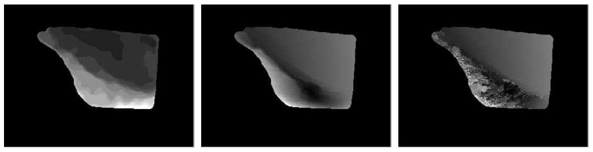
(a) (b) (c)

Fig. 3: Maps[1] of: **(a)** *scale factors* and **(b)** *optical flow magnitudes* for all points in Ψ, as estimated after applying the optical flow algorithm to the images of Fig. 2 and while using 10 possible scales $S = \{1, 0.9^{-1}, ..., 0.1^{-1}\}$. **(c)** Corresponding optical flow magnitudes when only one scale $S = \{1\}$ has been used. As expected, in this case the algorithm fails to produce exact optical flow for points that actually have larger scale factors.

(and which we will also follow here) was to compare I_k image patches with patches not only from I_{k+1} but also from rescaled versions of the latter image. We will again use a discrete set of scale factors $S = \{1{=}s_0{<}s_1{<}...{<}s_n\}$ and $I_{k+1,s}$ will denote I_{k+1} rescaled by s^{-1} with $s \in S$. Also, for any $q \in I_{k+1}, p \in I_k$ we will denote by $\overline{I_{k+1,s}(q)}$ a (small) fixed-size patch around the projection of q in $I_{k+1,s}$ and by $\overline{I_k(p)}$ an equal size patch of I_k at p.

Finally, to deal with the ill-posed character of the problem, we will first reduce the optical flow estimation to a discrete labeling problem and then formulate it in terms of energy minimization of a first order Markov Random Field [10]. The labels will consist of vectors $l = (d_x, d_y, s) \in \mathbb{R}^2{\times}S$, where the first 2 coordinates denote the components of the optical flow vector while the third one denotes the scale factor. This means that after labeling, not only an optical flow but also a scale estimation will be provided for each point (see Fig. 3(a)). Given a label l, we will denote its optical flow vector by $flow_l = (d_x, d_y)$ and its scale by $scale_l = s$.

For each point p in I_k, its possible labels will be: $\text{LABELS}_p = \{q - p : q \in \text{SEARCH}_p\} \times S$. That is we are searching for points only inside region SEARCH_p but across all scales in S. Getting an optical flow field is then equivalent to picking one element from the cartesian product $\text{LABELS} = \prod_{p\in\Psi} \text{LABELS}_p$. In our case, that element f of LABELS which minimizes the following energy will be chosen:

$$E(f) = \sum_{(p,p')\in\aleph} V_{p,p'}(f_p, f_{p'}) + \sum_{p\in\Psi} D(f_p)$$

The symbol $\aleph$ denotes a set of interacting pairs of pixels inside Ψ. The first term of $E(f)$ represents the sum of potentials of interacting pixels where each potential is given by the so-called Potts model (in which case the function $V(\mu, \nu)$ is equal to a non-zero constant if its arguments are equal and 0 otherwise).

Regarding the terms $D(f_p)$, these measure the correlation between corresponding image patches as determined by labeling f. According to labeling f, for a point p in I_k, its corresponding point is the projection of $p + flow_{f_p}$ in image $I_{k+1,scale_{f_p}}$. So:

$$D(f_p) = corr(\ \overline{I_k(p)}\ ,\ \overline{I_{k+1,scale_{f_p}}(p + flow_{f_p})}\ )$$

Energy $E(f)$ can be minimized using either the iterated conditional modes (ICM) algorithm [10] or recently introduced algorithms based on graph cuts [3]. The results after applying the ICM method to the images of Fig. 2 appear in Fig. 3.

6.2. Geometric morphing in region $\bar{\Psi}$

After estimation of optical flow $u_{k\to k+1}$, we may apply equation (3) to all points in Ψ and thus fill-in the $X_{dst}, Y_{dst}, Z_{dst}$ arrays therein (see Fig. 4(a)). To completely specify morphing, we still need to fill-in the values at the rest of the points, that is at points in $\bar{\Psi} = dom_k \backslash \Psi$. In other words, we need to specify the destination vertices for all points of L_k in $\bar{\Psi}$. Since these points do not have a physically corresponding point in L_{k+1}, we cannot apply (3) to get a destination 3D coordinate from model L_{k+1}. The simplest solution would be that no geometric morphing is applied to these points and that their destination vertices just coincide with their L_k vertices. However, in that case:

- points in Ψ will have destination vertices from L_{k+1}
- while points in $\bar{\Psi}$ will have destination vertices from L_k

The problem resulting out of this situation is that the produced destination maps $X_{dst}, Y_{dst}, Z_{dst}$ (see Figs. 4(b), 5(a)) will contain discontinuities along the boundary (say $\partial\Psi$) between regions Ψ and $\bar{\Psi}$, causing this way annoying discontinuity artifacts (holes) in the geometry of the "morphable 3D-model" during the morphing procedure. This will happen because the geometries of L_k and L_{k+1} (as well as their relative pose) have been estimated only approximately and will not therefore match perfectly.

The right way to fill-in the destination vertices at the points in $\bar{\Psi}$ is based on the observation that a physically valid destination 3D model should satisfy the following 2 conditions:

1. On the boundary of $\bar{\Psi}$, no discontinuity in 3D structure should exist

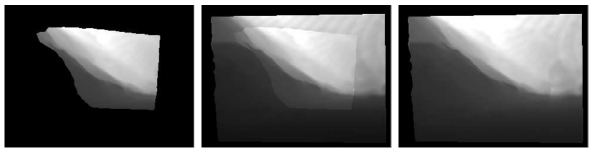
(a) (b) (c)

Fig. 4: **(a)** Destination depth map Z_{dst} for points in Ψ after using optical flow of Fig. 3(b) and applying eq. (3). **(b)** Depth map Z_{dst} of (a) extended to points in $\bar{\Psi}$ without applying geometric morphing. Observe the discontinuities along $\partial\Psi$. **(c)** Depth map Z_{dst} of (a) extended to points in $\bar{\Psi}$ after applying geometric morphing.

[1]Darker pixels (of a grayscale image) correspond to smaller values

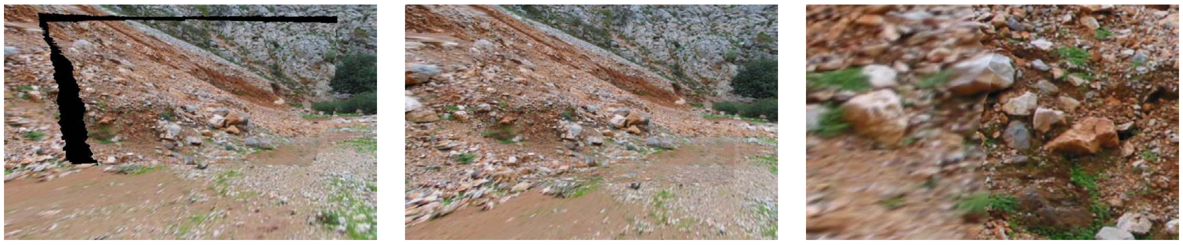
(a) (b) (c)

Fig. 5: Rendered views of the morphable 3D-model during transition from the key-position corresponding to image 2(a) to the key-position of image 2(b): **(a)** when no geometric morphing is applied to points in $\bar{\Psi}$ and **(b)** when geometric morphing is applied to points in $\bar{\Psi}$. **(c)** A close-up view of the rendered image in (b). Although there is no geometric discontinuity, there is a difference in texture resolution between the left part of the image (points in $\bar{\Psi}$) and the right part (points in Ψ) because only points of the latter part are morphed photometrically.

2. In the interior of $\bar{\Psi}$, the relative 3D structure of the initial L_k model should be preserved

Intuitively this means that as a result of morhing, vertices of L_k inside $\bar{\Psi}$ must be deformed (without distorting their relative 3D structure) so as to seamlessly match the 3D vertices of L_{k+1} along the boundary of $\bar{\Psi}$. In mathematical terms, preserving the relative 3D structure of L_k implies:

$$\begin{bmatrix} X_{dst}(p) - X_{dst}(p') \\ Y_{dst}(p) - Y_{dst}(p') \\ Z_{dst}(p) - Z_{dst}(p') \end{bmatrix} = \begin{bmatrix} X_k(p) - X_k(p') \\ Y_k(p) - Y_k(p') \\ Z_k(p) - Z_k(p') \end{bmatrix}, \forall p, p' \in \bar{\Psi}$$

which is easily seen to be equivalent to:

$$\begin{bmatrix} \nabla X_{dst}(p) \\ \nabla Y_{dst}(p) \\ \nabla Z_{dst}(p) \end{bmatrix} = \begin{bmatrix} \nabla X_k(p) \\ \nabla Y_k(p) \\ \nabla Z_k(p) \end{bmatrix}, \forall p \in \bar{\Psi}$$

We may then extract the destination vertices by solving 3 independent minimization problems (one for each of $X_{dst}, Y_{dst}, Z_{dst}$), all of the same type. It is therefore enough to consider only the Z_{dst} case. Z_{dst} is then the solution to the following optimization problem:

$$\min_{Z_{dst}} \iint_{\bar{\Psi}} \|\nabla Z_{dst} - \nabla Z_k\|^2, \ \ given\ Z_{dst}|_{\partial\bar{\Psi}} \quad (5)$$

The finite-difference discretization of (5) (using the underlying pixel grid) yields a quadratic optimization problem which can be reduced to a sparse (banded) system of linear equations, solved easily.

An alternative solution to the problem can be given by observing that a function minimizing (5) is also a solution to the following Poisson equation with Dirichlet boundary conditions [16]:

$$\triangle Z_{dst} = \text{div}(\nabla Z_k), \ \ given\ Z_{dst}|_{\partial\bar{\Psi}} \quad (6)$$

Therefore in this case the solution can be given by solving 3 independent Poisson equations of the above type. See Figures 4(c), 5(b) for a result produced with this method.

7. Rendering pipeline

At any time, only one "morphable 3D-model" L_{morph} needs to be displayed. Therefore, during the rendering process just the geometric & photometric morphing of L_k (as described in section 6) needs to take place .

For this purpose, two 3D triangle meshes $tri_{\Psi}, tri_{\bar{\Psi}}$ are first generated by computing 2D triangulations of regions Ψ, $\bar{\Psi}$ and by making use of the underlying geometric maps of L_k. Then, application of the geometric morphing (as defined by (1)) to the vertices of these meshes, amounts to exploiting simple "vertex-shader" capabilities [14] of modern graphics cards (along with the $\{X, Y, Z\}_{dst}$ maps of course).

Regarding the photometric morphing of mesh tri_{Ψ}, "multitexturing" [14] can be utilized at a first step. In this case, both images I_k, I_{k+1} will be applied as textures to mesh tri_{Ψ} and each vertex of tri_{Ψ} will be assigned texture coordinates from a point $p \in I_k$ as well as its corresponding point $u_{k\to k+1}(p) \in I_{k+1}$ (see (3)). Then, a straightforward "pixel-shader" [14] can be used to implement photometric morphing (as defined by (2)) for mesh tri_{Ψ}. Regarding $tri_{\bar{\Psi}}$, no actual photometric morphing takes place in there (see (4)) and so only image I_k needs to be mapped as a texture onto this surface.

Therefore, by use of pixel- and vertex-shaders, just 2 textured triangle meshes are given as input to the graphics pipeline at any time (a rendering path which is highly optimized in modern 3D graphics hardware).

8. Extending the modeling pipeline

Up to this point we have been assuming that during the image acquisition process, we have been capturing one stereoscopic image-pair per key-position along the path. We will now consider the case in which multiple stereoscopic views per key-position are captured and these stereoscopic views are related to each other by a simple rotation of the stereoscopic camera. This scenario is very useful in cases where we need to have an extended field of view (like in large VR screens) and/or when we want to be able to look around the environment. In this new case, multiple local 3D models per key-position will exist and they will be related to each other by a pure rotation in 3D space.

In order to reduce this case to the one already examined, it suffices that a single local model per key-position (called *3D-mosaic* hereafter) is constructed. This 3D model should replace all local models at that position. Then at any time during the rendering process, a morphing between a successive pair of these new local models (3D-mosaics) needs to take place as before. For this reason, the term *"morphable 3D-mosaics"* is being used in this case.

9. 3D-mosaic construction

As already explained, a 3D-mosaic at a certain position along the path should replace/comprise all local models coming from captured stereoscopic views at that position. Let $L_i = (X_i, Y_i, Z_i, I_i, dom_i)$ with $i \in \{1, \ldots, n\}$ be such a set of local models. Then a new local model $L_{mosaic} = (X_{mosaic}, Y_{mosaic}, Z_{mosaic}, I_{mosaic}, dom_{mosaic})$ needs to be constructed which amounts to filling its geometric and photometric maps. Intuitively, L_{mosaic} should correspond to a local model produced from a stereoscopic camera with a wider field of view placed at the same path position.

It is safe to assume that the images I_i (which are the left-camera images), correspond to views related to each other by a pure rotation. (Actually, the relative pose between 2 such images will not be pure rotation but will also contain a small translational part due to the fact that the stereoscopic camera rotates around the tripod and not the optical center of the left camera. However this translation is negligible in practice). We may therefore assume that the local 3D models are related to each other by a pure rotation as well. An overview of the steps that needs to be taken for the construction of L_{mosaic} now follows:

- As a first step the rotation between local models needs to be estimated. This will help us in registering the local models in 3D space.
- Then a geometric rectification of each L_i must take place so that the resulting local models are geometrically consistent with each other. This is a necessary step since the geometry of each L_i has been estimated only approximately and thus contains errors.
- Eventually, the maps of the refined and consistent local models will be merged so that the final map of the 3D-mosaic is produced

The most interesting problem that needs to be handled during the 3D-mosaic construction is that of making all models geometrically consistent so that a seamless (without discontinuities) geometric-map of L_{mosaic} is produced. Each of the above steps will be explained in the following sections.

9.1. Rotation (R_{ij}) estimation between views I_i, I_j

First the homography H_{ij} between images I_i, I_j will be computed. (Since the views I_i, I_j are related by a rotation, H_{ij} will be the infinite homography induced by the plane at infinity.) For the H_{ij} estimation [7], a sparse set of (at least 4) point matches between I_i, I_j is first extracted and then a robust estimation procedure (e.g. RANSAC) is applied to cope with outliers. Inlier matches can then be used to refine the H_{ij} estimate by minimizing a suitable error function.

If $R_{ij} \in SO(3)$ is the 3×3 orthonormal matrix representing rotation, then: $H_{ij} = K_{left} R_{ij} K_{left}^{-1} \Leftrightarrow R_{ij} = K_{left}^{-1} H_{ij} K_{left}$. In practice due to errors in the computed H_{ij}, the above matrix will not be orthonormal. So for the estimation of R_{ij} [15], an iterative minimization procedure will be applied to: $\sum_k dist(p'_k, K_{left} R_{ij} K_{left}^{-1} p_k)^2$, where (p_k, p'_k) are the inlier matches that resulted after estimation of H_{ij} while $dist$ denotes euclidean image distance. The projection of $K_{left}^{-1} H_{ij} K_{left}$ to the space $SO(3)$ of 3D rotation matrices will be given as initial value to the iterative procedure.

9.2. Geometric rectification of local models

Since at this stage the rotation between any two local models is known, hereafter we may assume that the 3D vertices of all L_i are expressed in a common 3D coordinate system. Unfortunately L_i are not geometrically consistent with each other, so the model resulting from combining these local models directly, would contain a lot of discontinuities at the boundary between any 2 neighboring L_i (see Fig. 6(c)). This is true because L_i have been created independently and their geometry has been estimated only approximately.

Let $\text{RECTIFY}_{L_i}(L_j)$ denote an operator which takes as input 2 local models, L_i, L_j, and modifies the geometric-maps only of the latter so that they are consistent with the geometric-maps of the former (the geometric maps of L_i do not change during RECTIFY). Assuming that such an operator exists , then ensuring consistency between all models can be achieved by merely applying $\text{RECTIFY}_{L_i}(L_j)$ for all pairs L_i,L_j with $i < j$.

So it suffices that we define $\text{RECTIFY}_{L_i}(L_j)$ for any 2 models, say L_i, L_j. Let X, Y, Z be the new rectified geometric-maps of L_j that we want to estimate so that they are geometrically consistent with those of L_i. Since we know homography H_{ij}, we may assume that image points of L_i have been aligned to the image plane of L_j. Let $\Psi = dom_i \cap dom_j$ be the overlap region of the 2 models. To be geometrically consistent, the new rectified maps of L_j should coincide with those of L_i at points inside Ψ:

$$\big[X(p)\ Y(p)\ Z(p)\big] = \big[X_i(p)\ Y_i(p)\ Z_i(p)\big]\ ,\ \forall p \in \Psi \quad (7)$$

We still need to define the rectified maps on $\bar{\Psi} = dom_j \backslash \Psi$. On one hand, this must be done so that no discontinuity appears along $\partial\bar{\Psi}$ (and thus seamless rectified maps are produced). On the other hand, we must try to preserve the relative 3D structure of the existing geometric-maps (of L_j) in the interior of $\bar{\Psi}$. The last statement amounts to:

$$\begin{bmatrix} X(p) - X(q) \\ Y(p) - Y(q) \\ Z(p) - Z(q) \end{bmatrix} = \begin{bmatrix} X_j(p) - X_j(q) \\ Y_j(p) - Y_j(q) \\ Z_j(p) - Z_j(q) \end{bmatrix} \quad \forall p, q \in \bar{\Psi}$$

or equivalently:

$$\begin{bmatrix} \nabla X(p) \\ \nabla Y(p) \\ \nabla Z(p) \end{bmatrix} = \begin{bmatrix} \nabla X_j(p) \\ \nabla Y_j(p) \\ \nabla Z_j(p) \end{bmatrix}\ ,\ \forall p \in \bar{\Psi} \quad (8)$$

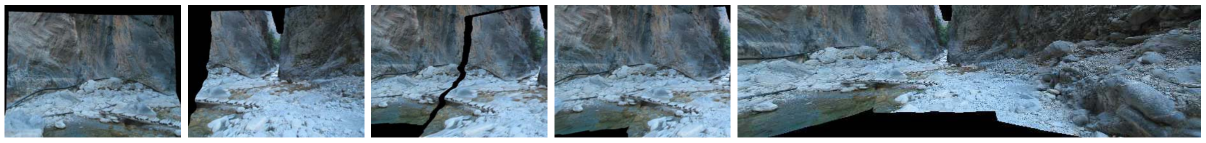
(a) (b) (c) (d) (e)

Fig. 6: Rendered views of: **(a)** a local model L_j **(b)** a local model L_i **(c)** 3D-mosaic of L_i, L_j without geometry rectification (holes are due to discontinuities in geometry and not due to misregistration in 3D space) **(d)** 3D-mosaic of L_i, L_j after $\mathrm{RECTIFY}_{L_i}(L_j)$ has been applied **(e)** a 3D-mosaic created from L_i, L_j plus another local model (not shown)

Then, based on (8), (7), we can extract the Z rectified map (X, Y maps are treated analogously) by solving the following optimization problem:

$$\min_Z \iint_{\bar{\Psi}} \|\nabla Z - \nabla Z_j\|^2, \; Z|_{\partial\bar{\Psi}} = Z_i|_{\partial\bar{\Psi}} \qquad (9)$$

The above problem, like the one defined by (5), can be reduced either to a banded linear system or to a Poisson differential equation, as explained in section 6.2. See Fig. 6(d) for a result produced with the latter method.

Another option for the merging of the geometric maps of L_i, L_j could have been the use of a feathering-like approach. The advantage of our approach (against feathering) is the preservation of the model's 3D structure. This can be illustrated with a very simple example. Let a rectangular planar object be at constant depth Z_{true}. Suppose that depth map Z_i, corresponding to most of the left part of the object, has been estimated correctly ($Z_i \equiv Z_{true}$) but depth map Z_j, corresponding to most of the right part of the object, has been estimated as $Z_j \equiv Z_{true} + error$. When using a feathering-like approach, the resulting object will appear distorted in the center (its depth will vary from Z_{true} to $Z_{true} + error$ therein). On the contrary using our method, an object still having a planar structure will be produced. This is important since such errors may exist in models produced from disparity estimation. In fact the magnitude of these errors will be analogous to depth and can thus be quite large for distant objects.

9.3. Merging the rectified local models

Since H_{ij} is known for any i, j, we may assume that all local models are defined on a common image plane. Therefore, due to the fact that the rectified geometric-maps are consistent with each other, we can directly merge them so that the $\{X, Y, Z\}_{mosaic}$ maps are produced. For the creation of the I_{mosaic} photometric map, a standard image-mosaicing procedure [15] can be applied independently. The valid region of the 3D-mosaic will be: $dom_{mosaic} = \cup_i dom_i$. One 3D-mosaic so generated appears in Fig. 6(e).

10. Conclusions

A new approach for interactive walkthroughs of large, outdoor scenes has been proposed. No global model of the scene needs to be constructed and at any time during the rendering process, only one "morphable 3D-mosaic" is displayed. Our method uses a rendering path which is highly optimized in modern 3D graphics hardware and thus can produce photorealistic renderings at interactive frame rates. Finally, it is worth mentioning that our system has been also tested successfully for the case of a scene inside Samaria Gorge, using as input 45 stereoscopic views acquired along a path over 100 meters long. 15 key-positions have been selected along the path and 3 stereoscopic views per key-position have been captured (covering this way approximately a 120° field of view at any time).

References

[1] D. G. Aliaga, T. Funkhouser, D. Yanovsky, and I. Carlbom. Sea of images. In *VIS 2002*, pages 331–338, 2002.

[2] J.-Y. Bouguet. Camera Calibration Toolbox for MATLAB. http://www.vision.caltech.edu/bouguetj/calib_doc.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, November 2001.

[4] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH 96*, pages 11–20, 1996.

[5] S. Fleishman, B. Chen, A. Kaufman, and D. Cohen-Or. Navigating through sparse views. In *VRST99*, pp. 82-87, 1999.

[6] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH 96*, pages 43–54, 1996.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.

[8] R. Koch. 3d surface reconstruction from stereoscopic image sequences. In *ICCV95*, pages 109–114, 1995.

[9] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH 96*, pages 31–42, 1996.

[10] S. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

[11] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering approach. In *Siggraph95*, pp. 39-46, 1995.

[12] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3d models with a video camera. In *Proc. 3DIM'99*, pages 14–23, 1999.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, April 2002.

[14] M. Segal and K. Akeley. The OpenGL Graphics System: A Specification (Version 1.5). http://www.opengl.org.

[15] R. Szeliski. Video mosaics for virtual environments. *IEEE CGA*, 16(2):22–30, March 1996.

[16] D. Zwilliger. *Handbook of Differential Equations*. Boston, MA: Academic Press, 1997.