EUROGRAPHICS 2008 / G. Drettakis and R. Scopigno
(Guest Editors)

Volume 27 (2008), Number 2

Augmented Panoramic Video

C. Hermans C. Vanaken T. Mertens F. Van Reeth P. Bekaert

Hasselt University - Expertise Center for Digital Media
transnationale Universiteit Limburg - School of Information Technology
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{firstname.lastname}@uhasselt.be

Abstract

Many video sequences consist of a locally dynamic background containing moving foreground subjects. In this paper we propose a novel way of re-displaying these sequences, by giving the user control over a virtual camera frame. Based on video mosaicing, we first compute a static high quality background panorama. After segmenting and removing the foreground subjects from the original video, the remaining elements are merged into a dynamic background panorama, which seamlessly extends the original video footage. We then re-display this augmented video by warping and cropping the panorama. The virtual camera can have an enlarged field-of-view and a controlled camera motion. Our technique is able to process videos with complex camera motions, reconstructing high quality panoramas without parallax artefacts, visible seams or blurring, while retaining repetitive dynamic elements.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - Texture, Video Analysis; I.4.3 [Image Processing and Computer Vision]: Enhancement - Registration; I.4.8 [Image Processing and Computer Vision]: Scene Analysis - Time-varying imagery;

1. Introduction

Videos are becoming an increasingly popular means of conveying information, and as a result the availability of video editing tools is growing every day. The addition of the temporal dimension to the spatial domain provides us with a large new range of potential problems and possibilities. Aside from the natural extensions from image editing, video sequences can be adjusted or improved in other ways [BM03].

One example of such video augmentations is video stabilization: the task of removing unintended and therefore unwanted shaky motion from a video. This is commonly achieved by computing and smoothing the motion path, either by making global adjustments commonly using a reference frame [LKK03] or by smoothing out local displacements [MOTS05]. Either way, the stabilized video will have gaps due to the warping of the original content. Instead of simply cropping the result, mosaicing [LKK03] or motion inpainting [MOTS05] can then be applied to fill in the missing information.

Another class of applications deals with restructuring the image or video dimensions, while preserving a maximum amount of salient information. A recent example in the image domain by Avadan *et al.* [AS07] allows for content-aware image resizing using seam carving. A work more closely related to our own, the video retargeting algorithm by Liu *et al.* [LG06], aims at adapting videos to fit a new display size. As this introduces virtual pans and cuts, their approach is designed to minimize the loss of important information.

Our work will focus on a common type of video sequence, in which the videographer shoots a scene by rotating the camera to capture the entire panorama, possibly zooming into areas of particular interest. Also typically, video sequences are focused on dynamic subjects, such as people or animals. We wish to re-display and manipulate such sequences in a meaningful way, presenting a technique that gives the user control over the camera's motion and field of view. As we are usually interested in increasing our field of view, our work can be seen as an inverse case of the video retargetting algorithm of Liu *et al.* [LG06].

Several assumptions are made. Most importantly, the video should be recorded from approximately a single location in the scene, i.e. the camera may only undergo a rotational motion. Significant translation would introduce severe parallax effects, which would require a more elaborate scene analysis with full 3D information. However, as it is practically impossible to avoid parallax, we compensate for it in our technique. Our contributions to the field are twofold: (i) the idea of editing a panning/rotating video sequence using a full panoramic representation; (ii) a robust video mosaicing algorithm that produces high quality panoramas without parallax artefacts, seams or blurring, while retaining repetitive dynamic elements.

2. Related Work

Our work is rooted in several subdomains of computer graphics and computer vision. Video registration, texture synthesis, and image & video completion are all related to the work presented in this paper.

2.1. Video Registration

The task of properly aligning partially overlapping images captured by a camera is commonly referred to as video registration. In case the camera follows a motion pattern more sophisticated than the common monodimensional panning sequence, extra measures need to be taken to assure that all available information is exploited. This information usually comes in the shape of an approximation of the frame topology. Generally some form of global optimization is utilized to ensure an overall consistent registration.

In the last decade, many approaches to global registration have been proposed. We will restrict ourselves to those most closely related to our own work, those that let topological information guide the registration process. A graph representation is commonly used to depict the topology, casting the problem as the identification of the shortest path [KCM00, MFM04, SHK98]. We opted for an alternative graph-based approach: instead of weighing the edges with some confidence measure of choice, our algorithm is designed to minimize the number of intermediary nodes between each frame and the reference frame. This is based on the notion that we do not necessarily need to know *how good* every single edge in the graph is, only that they are *good enough*. Our approach is aimed at reducing computation time, trying to minimize the number of homography computations.

2.2. Image/Video Completion & Texture Synthesis

Image completion poses the problem of filling in missing pixels in large unknown regions of an image in a visually consistent way. This is very similar to the objective of texture synthesis, in which a large area of texture information needs to be generated, based on the limited intensity information available in a smaller sample.

Historically, exemplar-based techniques have proven to be the most successful in dealing with this problem, copying pixels or source patches from the observed part of the image [EL99, CPT03, DCOY03, KEBK05]. The common drawback to these approaches is their greedy approach to filling the image, which can often lead to visual inconsistencies. Initial attempts to avoid this problem have taken a more global approach, using Expectation Maximization (EM)-like schemes for optimizing the process [KSE*03, WSI04]. However, EM is known to be particularly sensitive to initialization and can get trapped in poor local minima. Other recent approaches have applied dynamic programming or belief propagation [YFW01] to reach a more globally consistent image. Most of these algorithms guide the completion process by influencing the order by which the synthesis proceeds. This can either be done manually by user assistance, e.g. Jian Sun *et al.* [SYJS05] give priority to user-specified curves on which the most salient missing structures reside, or it can be deduced by the algorithm itself [KT06].

Recently some authors have extended the application range of their image completion and texture synthesis algorithms to the video domain. In one related work, Agarwala *et al.* [AZP*05] constructed 'panoramic video textures'. Starting from a video segment filmed by panning a camera across a dynamic scene, they combine looping segments of a constant duration in order to construct a single panoramic video texture. Even though this work naturally relates to dynamic panoramic backgrounds, there are several issues that prevent us from applying this technique to our situation. Our augmented video has a predetermined finite duration, and contains pixel intensities that should remain unchanged to preserve the original content. We cannot discard pixels from the input sequence to create a better fit for the required constant looping time. Finally, while we would like to use arbitrary input videos, their method is restricted to horizontal panning sequences.

2.3. Background Estimation

The problem of estimating a consistent background is commonly addressed by applying a temporal mean or median filter to the video at pixel level. However, in case of stationary occlusors that persist for more than half the sequence length, or when dealing with the presence of parallax effects, these simple approaches fail. Spatial support is required as an additional cue to improve pixel-level algorithms.

The work most closely related to our own is that of Columbari *et al.* [CFM06]. They present a region growing algorithm, which starts from patches that are always visible in the scene, gradually forming a consistent background. This approach shows similarities to the early examplar-based image completion algorithms, and potentialy inherits their common drawback: its greedy approach can lead to visual inconsistencies when two regions come together. While we use similar cues to guide our background synthesis, our

algorithm poses background estimation as an optimization problem with a well defined energy function. Our formulation also allows for sharper patches to be chosen over their blurred counterparts, reducing (if not completely removing) parallax effects if the intensity information is available in the original video.

3. Overview

Our video augmentation pipeline consists of five different processing steps. In the first of these steps, we properly warp and align all input images in a common reference frame. Using the result of this video registration step, we proceed on computing a globally consistent static background, containing sharp details and free of parallax artefacts. This background is used to make a distinction between the dynamic foreground elements (actors) and the static or dynamic background elements (repetitive/quasi-repetitive motions, or complex stochastic phenomena with an overall stationary structure). When the background elements are identified, the warped input video is extended with a dynamic background panorama. Finally, this dynamic background panorama is warped back to the original video, and modified according to the user-controlled virtual camera.

4. Our Approach

4.1. Notation

Two images of the same scene are related by a non-singular linear transformation of the projective plane in two cases: (a) if the scene is planar or (b) if the center of projection does not change, i.e. the only degrees of freedom are due to the orientation of the camera. In these cases we do not suffer from the effects of parallax, and the images can be composed together to form a mosaic.

Image points are represented by their homogeneous coordinates $\widetilde{\mathbf{x}} = (x, y, w)$, with $\mathbf{x} = (\frac{x}{w}, \frac{y}{w})$ being the corresponding Cartesian coordinates. A linear transformation of the projective plane, called a homography, is represented by a 3×3 matrix $\mathbf{H}$ when $\widetilde{\mathbf{x}}_j = \mathbf{H}_{i,j}\widetilde{\mathbf{x}}_i$, where $\widetilde{\mathbf{x}}_i$ and $\widetilde{\mathbf{x}}_j$ are corresponding points in frames i and j respectively.

4.2. Video Registration

Our video registration pipeline is essentially a two-step process, with an optional bundle adjustement step. During the initial estimation step, we have no knowledge of the frame topology (the relative spatial positioning of the frames), so we rely on temporal information only. Using the results of this initial estimation, we subsequently take a graph-based approach, using the newly acquired spatial information. Finally, we can employ an optional bundle adjustment step.

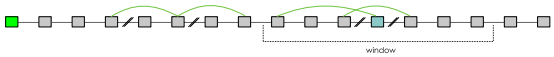

Figure 1: *(TiA): a sliding window of potential homography candidates is checked instead of only linking consecutive frames. (boxes: frames, edges: computed homographies with sufficient inlier support, black/green: neighbors/short-cuts)*

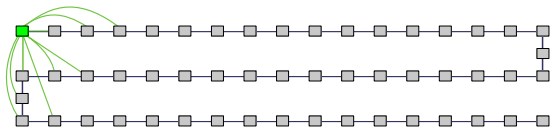

Figure 2: *(TdA, step 0): using topoligy information, all potential candidates for direct linking to the reference frame are computed. Those with sufficient support are added to the graph.*

4.2.1. Homography Computation

Feature detection and matching is done by employing the Kanade-Lucas-Tomasi tracker [LK81, TK91]. After proper normalization of the found correspondences [Har97], we employ a RANSAC-based [FB81] algorithm to compute homographies, using minimal sample sizes [BHN07]. When the RANSAC procedure has computed an initial homography and a matching set of initial inliers, we employ the method proposed by Kanatani *et al.* [iKO99]. This method is based on a statistical renormalization technique, and determines a statistically optimal homography. This non-linear estimation is repeated until a stable amount of inliers is achieved.

4.2.2. Topology-independent Alignment (TiA)

As a common first step in many graph-based registration algorithms, the inter-frame homographies between all neighboring frames are calculated. In order to establish an initial guess of the frame topology, we could recursively concatenate the homographies from each frame to a chosen reference frame.

$$\begin{cases} \mathbf{H}_{r,r} = I \\ \mathbf{H}_{i,r} = \mathbf{H}_{i+1,r}\mathbf{H}_{i,i+1} & \texttt{if } i < r \\ \mathbf{H}_{i,r} = \mathbf{H}_{i-1,r}\mathbf{H}_{i,i-1} & \texttt{if } i > r \end{cases} \tag{1}$$

These homographies are commonly computed from one frame of the input sequence to the next. However, for the purpose of image mosaicing, our experiments have indicated that computing these homographies on pre-warped images (using the target frame's homography as the warping function) results in more accurate estimates. The reasoning behind this is that the pixel-error is measured in the coordinate space of the final panorama directly.

Due to the recursive nature of the computation process, estimation errors will propagate down the homography chain.

A grove misalignment would immediately break the chain. Therefore, we propose a slightly different scheme, in which we use a sliding window of potential homography candidates instead of simply linking consecutive frames (fig.1). As we have mentioned before, when we compute a homography we require the target frame's warping function to be known. As such, we keep track of the frames that are already linked to the reference frame, and label them as committed. Starting from the reference frame, we try to connect adjacent uncommitted frames to committed frames. Initially, only the reference frame itself is labeled as committed, as it is the only one whose warping function to the reference mosaic is already known. For each uncommited frame i, we attempt to compute the homography $\mathbf{H}_{i,j}$ to each commited node $j \in [i-d, i+d]$, starting with the closest neighbors. As soon as we find a homography with a confidence value above a predefined threshold, it is stored and the source frame is labeled as committed. We repeat this procedure until no more uncommitted frames remain.

There are two possible reasons why no more frames are committed: (a) either all the frames have a parent frame (a frame through which they are linked to the reference frame) and an associated homography to this frame, or (b) for each of the uncommitted frames i, no potential candidate j within the given window size has been found. In the latter case, we use the previously stored confidence values to find the homography with the highest confidence, add the source frame to the list of committed frames, and resume the previous procedure.

This produces a homography tree with constraints on the confidence values associated between the different nodes. Unfortunately, considering the depth of the tree, the propagated estimation errorss will still result in a considerable misalignement at the end of the sequence. However, we now have a first estimate of the frame topology in the reference mosaic, which we can use to provide us with a more accurate registration algorithm.

4.2.3. Topology-dependent Alignment (TdA)

In this stage the homography tree that resulted from the previous stage will be transformed into a new instance, taking into account the estimated topology information.

As we have stated before, our algorithm is designed to minimize the number of intermediary nodes between each frame and the reference frame, based on the notion that we do not necessarily need to know how good every single edge in the graph is, only that that they are good enough (fig.2). If we can guarantee a minimum level of confidence, the results will be usable for future computations. During our experiments, we have used the number of correspondences within predefined error bounds as our confidence metric of choice. We provide an outline of our algorithm in alg.1.

Note the similarities with the previous stage: we utilize a confidence threshold to decide if we add the child node i of the homography $\mathbf{H}_{i,j}$ to the set of connected (committed) frames. Also, once a node is connected to the rest of the graph, its warping function is known. This way, we can always use pre-warped images to perform the homography estimation.

Algorithm 1 Topology-dependent Alignment

1. $\mathcal{U} := \{i \mid i \neq r\}$; % unconnected frames ($\neq$ reference frame)
2. $\mathcal{C} := \{r\}$; % previously connected frames
3. $\mathcal{N} := \emptyset$; % newly connected frames
4. `while` $\mathcal{U} \neq \emptyset$
   a. $\forall i \in \mathcal{U}$:
      i. Sort $j \in \mathcal{C}$, according to $\|i-j\|$, closest first;
      ii. Find first $j \in C$, where $\#inliers(\mathbf{H}_{i,j}) \geq threshold$
      iii. If j found: $\mathcal{N} = \mathcal{N} \cup \{i\} \ \wedge \ \mathcal{U} = \mathcal{U} - \{i\}$
   b. $\begin{cases} \texttt{if } \mathcal{N} \neq \emptyset : \mathcal{C} = \mathcal{N} \\ \texttt{else} : \mathcal{C} = \{k \mid \arg\max(\#inliers(\mathbf{H}_{k,j})), k \in \mathcal{U}, j \notin \mathcal{U}\} \end{cases}$
5. `end while`

In principle, when iterating through the set of unconnected frames $\mathcal{U}$, we could compute the homography between each pair $(i, j) \in \mathcal{U} \times \mathcal{C}$ to look for new additions to the graph. Homography computation however is an expensive operation, and should be avoided if a lack of inlier support is expected beforehand. Therefore, we will only consider frames with a significant degree of overlap as potential candidates for addition.

We use the available topology information to reduce the search space of potential edge candidates: in order to establish the degree of overlap, homographies $\mathbf{H}_{i,r}$ from the topology-independent alignment step are used as an approximation to the true registration matrices. As an overlap measure, we use the normalized distance between centroids:

$$\delta_{ij} = \frac{max(0, |\mathbf{c}_i - \mathbf{c}_j| - |d_i - d_j|/2)}{min(d_i, d_j)} \qquad (2)$$

where $\mathbf{c}_i$, $\mathbf{c}_j$, d_i and d_j are the centroids and the diameter of the projection onto the mosaic of frames i and j, respectively.

4.2.4. Bundle Adjustement

As a final step we apply the bundle adjustment step proposed by Marzotto *et al.* [MFM04], which finds the solution $\{\mathbf{H}_i\}$ that minimizes the total misalignment of a predefined set of m grid points on the mosaic. Let x_k be a grid-point and let $\mathcal{E}_k$ be the set of edges $(i, j) \in \mathcal{E}$ so that x_k belongs to the overlap region between frame i and frame j. The error at the grid-point $\mathbf{x}_k$ is defined as:

$$E_k = \frac{1}{|\mathcal{E}_k|} \sum_{(i,j) \in \mathcal{E}_k} \left\| \mathbf{x}_k - \pi \left(\mathbf{H}_{i,r} \mathbf{H}_{i,j} \mathbf{H}_{j,r}^{-1} \tilde{\mathbf{x}}_k \right) \right\|^2 \qquad (3)$$

where π transforms homogeneous coordinates into Cartesian (pixel) coordinates.

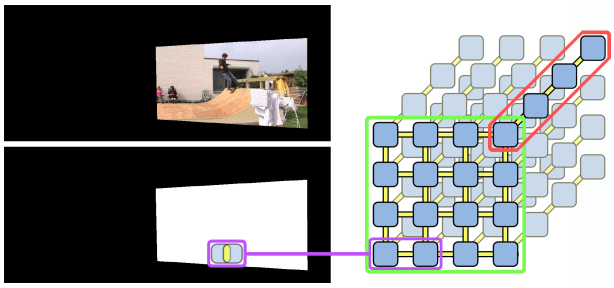

Figure 3: *(left) An example of a warped input frame, and the associated binary mask. (right) Background Estimation MRF: a 2D grid of nodes (green), with label patches in the temporal dimension (red). Every pair of connected nodes represents overlapping image patches (purple).*

Since we want to minimize the error at all grid points simultaneously, we end up with a system of non-linear equations that can be cast as a least-squares problem.

$$\min_{\{\mathbf{H}_{i,r}\}} \sum_{k=1}^{m} E_k^2 \tag{4}$$

The Levenberg-Marquardt algorithm is used to solve Eq. (4), using the previous set of $\{\mathbf{H}_{i,r}\}$ as the starting solution. Data standardization is carried out to improve the conditioning of the problem [Har97].

4.3. Static Background Estimation

After we have properly registered all frames, the next step in our pipeline consists of computing a consistent static background. In an ideal situation, registration would be perfect and every background pixel would be visible for a sufficient period of time. Unfortunately, real-world footage is rarely this perfect, and as a result we will have to deal with the effects of motion blur and parallax. For good measure, we will also be dealing with the possibility of actors that stay in a single place for a significant period of time, only exposing the true background for a few seconds.

As we have stated before, our background estimation algorithm shows some similarities to the region growing algorithm of Columbari *et al.* [CFM06]. However, unlike the greedy approach taken in their work, we have opted to pose the background estimation as a discrete global optimization problem.

4.3.1. Problem Statement

Given a set of warped input images $\mathcal{I}_i$ and their binary masks $\mathcal{B}_i$ (fig.3), the goal of our algorithm is to compute a visually plausible background by merging spatially consistent, but temporally varying patches into a consistent background image. To this end, we propose the use of a discrete Markov Random Field (MRF).

The nodes $\mathcal{N}$ of the MRF are defined by placing an image lattice over the total space occupied by the mosaic, with a horizontal and vertical spacing of $step_x$ and $step_y$ respectively. Each node is uniquely defined by their (x,y) coordinate on the mosaic, and the edges $\mathcal{E}$ of the MRF are defined by looking at the 4-neighborhood of each individual node. The total label set $\mathcal{L}$ consists of all possible $w \times h$ patches around every node $n_i \in \mathcal{N}$. Thus, the labels $l \in \mathcal{L}$ are uniquely defined by the spatial coordinates (x,y) of their center pixel, and their frame number $t \in [1,N]$. Note that $step_x$ and $step_y$ are set so that a region of overlap between neighboring patches of size $w \times h$ always exists. Every node $n_i(x_i,y_i) \in \mathcal{N}$ has a maximum of N possible label candidates $l(x_l,y_l,t)$ where $(x_l,y_l) = (x_i,y_i)$. Also, a label (x,y,t) will only be considered a valid background candidate if the full patch window $\mathcal{W} = [x-\frac{w}{2},x+\frac{w}{2}] \times [y-\frac{h}{2},y+\frac{h}{2}]$ is marked in the binary mask $\mathcal{B}_t$ (fig.3), or if it is only partially marked but located on the border of the mosaic.

The single node potential $V_i(l)$ for placing label l over node n_i will describe the likelihood of patch l being part of the background. This likelihood can be expressed in terms of the number of frames in which the patch is visible during the entire sequence. However, a single patch has no inherent information about the duration of its visibility, so we are required to perform an a priori clustering step. As in the work of Columbari *et al.* [CFM06], we apply single linkage agglomerative clustering [JMF99] to group our labels l_t into clusters $\mathcal{C}_t \subset \mathcal{L}_i$. Every cluster $\mathcal{C}_t$ will be defined by choosing one of its labels l_t as the primary label, and each node will be assigned a set of clusters instead of a set of individual labels. Based on the size of these clusters, we can now define our single node potentials as:

$$V_i(\mathcal{C}) = \alpha \left[1 - \left(\frac{|\mathcal{C}|}{N} \right)^2 \right] \tag{5}$$

Lastly, the pairwise potential $V_{ij}(\mathcal{C},\mathcal{C}')$ will measure how well these clusters agree on their region of overlap. We will define the pairwise potential by the sum of squared differences (SSD) of the mean labels from the respective clusters in this area of overlap $\mathcal{A}$, divided by the amount of overlap pixels $|\mathcal{A}|$:

$$V_{ij}(\mathcal{C},\mathcal{C}') = \beta \left[\frac{1}{|\mathcal{A}|} \sum_{(x,y)\in\mathcal{A}} \left(\bar{\mathcal{I}}(x,y) - \bar{\mathcal{I}}'(x,y) \right)^2 \right] \tag{6}$$

Based on this formulation, where α and β are user-specified weights, our goal will now be to assign a cluster $\hat{\mathcal{C}}_i \subset \mathcal{L}$ to each node n_i, so that the total energy cost $E(\{\hat{\mathcal{C}}_i\})$ of the MRF is minimized, where:

$$E(\{\hat{\mathcal{C}}_i\}) = \sum_{i=1}^{|\mathcal{N}|} V_i(\hat{\mathcal{C}}_i) + \sum_{(i,j)\in\mathcal{E}} V_{ij}(\hat{\mathcal{C}}_i,\hat{\mathcal{C}}_j) \tag{7}$$

Figure 4: *We use the median of absolute deviations to: (a) segment out foreground elements, and (b) classify background regions as static or dynamic (see labels)).*

4.3.2. Energy Minimization by Belief Propagation

As an advantage of formulating background estimation as an energy minimization problem, we can now apply belief propagation to our energy function.

Belief propagation (BP) is an iterative inference algorithm that works by propagating local messages along the nodes of an MRF [YFW01]. Messages sent from node n_i to n_j form a set $\{m_{ij}(l)\}_{l\in\mathcal{L}}$, where element $m_{ij}(l)$ indicates how likely node n_i thinks that node n_j should be assigned label l. Furthermore, messages are updated (i.e. sent) until convergence as follows:

$$m_{ij}(l) = \min_{l_i\in\mathcal{L}}\{V_i(l_i) + V_{ij}(l_i,l_j) + \sum_{k:k\neq j,(k,i)\in\mathcal{E}} m_{ki}(l_i)\} \quad (8)$$

This update rule is associated with the min-sum version of BP, where the potentials are described in the -*log* domain. After convergence, a set of beliefs $\{b_i(l)\}_{l\in\mathcal{L}}$ is computed for each node, where belief $b_i(l)$ is defined as follows:

$$b_i(l) = -V_i(l) - \sum_{k:(k,i)\in\mathcal{E}} m_{ki}(l) \quad (9)$$

These beliefs approximate the max-marginal of the posterior at node n_i, and thus describes the likelihood that the label l should be assigned to that node. Based on this fact, a node is then assigned the label with the maximum belief, i.e. $\hat{l}_i = \arg\max_{l\in\mathcal{L}} b_i(l)$. It is known that, for tree structured graphs, BP will always converge to the optimal solution, while for graphs with loops, it can only guarantee to find a local optimum.

4.3.3. Dual-step Energy Minimization

In order to reduce the computational time of our algorithm, we have opted to perform our background estimation in two seperate steps. During the initial step, we will try to assign a cluster $\hat{\mathcal{C}}_i \subset \mathcal{L}$ to each node n_i, minimizing the total energy cost $E(\{\hat{\mathcal{C}}_i\})$ of the MRF (eq.7). Here, the clusters take the role of labels in the BP algorithm.

In a subsequential step, we will unpack these clusters and assign a label $\hat{l}_i \in \hat{\mathcal{C}}_i$ to each node n_i, minimizing another energy cost $E(\{\hat{l}_i\})$ associated with individual labels rather than clusters:

$$E(\{\hat{l}_i\}) = \sum_{i=1}^{|\mathcal{N}|} V_i(\hat{l}_i) + \sum_{(i,j)\in\mathcal{E}} V_{ij}(\hat{l}_i,\hat{l}_j) \quad (10)$$

The single node potential function $V_i(l)$ of label l estimates the level of blur of the corresponding window $\mathcal{W}$:

$$V_i(l) = -\frac{1}{|\mathcal{W}|}\sum_{(x,y)\in\mathcal{W}} \left(\mathcal{I}(x,y) - \bar{\mathcal{I}}(\mathcal{W})\right)^2 \quad (11)$$

where $\bar{\mathcal{I}}(\mathcal{W})$ symbolizes the mean of window $\mathcal{W}$. By defining this single node potential, we encourage the use of the sharper labels in each cluster over their more blurry counterparts. The pairwise potential $V_{ij}(l,l')$ computes the SSD of the labels in their respective area of overlap $\mathcal{A}$, divided by its size $|\mathcal{A}|$.

Choosing this two-step approach over a single minimization step decreases computation times due to the reduced number of labels in each step. In addition, it will also increase the robustness of the algorithm, as false positives are most likely to have been removed from the label set after the clustering step.

4.4. Foreground Segmentation

The purpose of the foreground segmentation component is to identify the actors (dynamic foreground elements) in our scene. For every warped input frame we need to decide which pixels belong to the (static or dynamic) background and which pixels belong to the foreground.

To do this, we use a classifier that is based on the X84 outlier rejection rule [HRRS86]. Every pixel of each warped frame is compared with its associated pixel in the static background image calculated in section 4.3. To make a robust classification based on the difference between these pixel values, we incorporate the median of absolute deviations (MAD) into the computations (fig.4a). The MAD is a statistical measure that is commonly used to describe the variability of data with outliers.

$$MAD(x,y) = med_i\{|\mathcal{I}_i(x,y) - bg(x,y)|\} \quad (12)$$

An input pixel (x,y) belongs to a foreground element if

$$\frac{|\mathcal{I}_i(x,y) - bg(x,y)|}{MAD(x,y)^2} > \chi_3^{-1}(\alpha) \quad (13)$$

where $\chi_3^{-1}(\alpha)$ is the inverse-chi-square distribution with 3 degrees of freedom and a confidence value of α. The resulting segmentation images $\mathcal{S}_i$ are cleaned up by using standard morphological filtering operations.

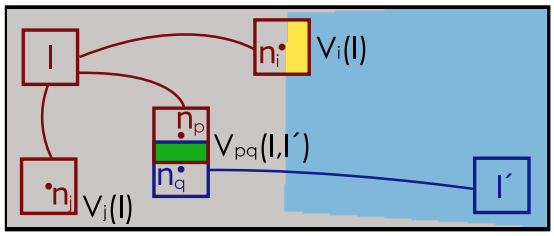

Figure 5: *Dynamic Background Potentials: for boundary node n_i its single node potential $V_i(l)$ will be an SSD over the yellow region, while for nodes n_p, n_q their pairwise potential $V_{pq}(l,l')$ will be an SSD over the green region. Non-boundary node n_j has zero single node potential.*

4.5. Dynamic Background Estimation

Our process of creating dynamic background content is structured in a fashion similar to our background generation component.

4.5.1. Single-step Energy Minimization

Given a set of warped input images $\mathcal{I}_i$, their binary masks $\mathcal{B}_i$ and a foreground/background segmentation $\mathcal{S}_i$, the goal of this component is to compute a visually plausible panoramic video by merging spatially consistent, but temporally varying patches into a consistent video panorama. To this end, we propose the use of a new MRF, expanding our static background estimation MRF into the temporal dimension.

The nodes $\mathcal{N}$ are once again defined by placing an image lattice over the total space occupied by the mosaic, adding a temporal dimension t. Each node is thus uniquely defined by its (x,y,t) coordinates, and the edges $\mathcal{E}$ are defined by looking at the 6-neighborhood of each individual node. The label set $\mathcal{L}$ is a subset of the one we used for static background estimation. We remove the labels containing the foreground object (encoded in the segmentation images $\mathcal{S}$), which leaves us with both the static and dynamic background labels.

Nodes n_i located on the border of a warped input image will already contain some initial content. Therefore, any label $\hat{l}_i$ assigned to these nodes should retain as much intensity information present as possible. As such, the single node potential $V_i(l)$ of assigning label l to node n_i represents how well the intensity information of label l agrees with the intensities present in the window $\mathcal{W}$ around the center of node n_i (fig.5):

$$V_i(l_t) = \alpha \left[\frac{1}{|\mathcal{W}|} \sum_{(x,y)\in\mathcal{W}} \mathcal{B}_i(x,y) \left(\mathcal{I}_i(x,y) - \mathcal{I}_t(x,y)\right)^2 \right] \tag{14}$$

Lastly, the pairwise potential $V_{ij}(l,l')$ must be defined in a way that provides us with both spatial and temporal consistency. In case nodes i and j are spatial neighbors, the pairwise potential is defined by the normalized SSD over the area of overlap $\mathcal{A}$:

$$V_{ij}^S(l,l') = \beta \left[\frac{1}{|\mathcal{A}|} \sum_{(x,y)\in\mathcal{A}} \left(\mathcal{I}(x,y) - \mathcal{I}'(x,y)\right)^2 \right] \tag{15}$$

When dealing with temporal neighbors, we want to encourage temporal continuity for dynamic elements. However, not all dynamic background movement is caused by actual scene movement, as some of it originates from parallax artefacts. Because incorporating this unwanted movement in our results leads to visual artefacts, we need to subdivide the label set $\mathcal{L}$ into a subset of static ($\mathcal{L}_S$) and dynamic labels ($\mathcal{L}_D$). This subdivision is based on thresholding, using the MAD values calculated in the foreground segmentation step.

$$\kappa(l) = \left(\frac{1}{|\mathcal{W}_l|} \sum_{(x,y)\in\mathcal{W}_l} MAD(x,y) \right)^2 \tag{16}$$

If $\kappa(l)$ exceeds a predefined threshold, label l will be considered dynamic (see fig.4b).

Depending on which subset two labels l and l' are in, temporal costs are chosen to either encourage temporal continuity (for the dynamic elements), or to increase temporal coherence (for the static elements). In the first case, we will assign a penalty to subsequential labels in the output video that are not subsequential in the original sequence:

$$V_{ij}^{TD}(l,l') = \gamma \quad \texttt{if} \quad \left[t(n_i) - t(n_j)\right] \neq \left[t(l) - t(l')\right] \tag{17}$$

In case of static background elements, the pairwise potential is defined by the normalized SSD over their common spatial window $\mathcal{W}$:

$$V_{ij}^{TS}(l,l') = \lambda \left[\frac{1}{|\mathcal{W}|} \sum_{(x,y)\in\mathcal{W}} \left(\mathcal{I}(x,y) - \mathcal{I}'(x,y)\right)^2 \right] \tag{18}$$

Based on these formulations, where α, β, γ and λ are user-specified weights and $t(l)$ returns the label's frame number, a label $\hat{l}_i \in \mathcal{L}$ should be assigned to each node n_i, so that the total energy cost $E(\{\hat{l}_i\})$ of the MRF is minimized, where:

$$E(\{\hat{l}_i\}) = \sum_{i=1}^{|\mathcal{N}|} V_i(\hat{l}_i) + \sum_{(i,j)\in\mathcal{E}} \left[V_{ij}^S(\hat{l}_i,\hat{l}_j) + V_{ij}^{T^*}(\hat{l}_i,\hat{l}_j) \right] \tag{19}$$

4.6. Visualization

In the end, the goal of our system is to re-dispay video sequences with a controlled camera motion, field of view and zoom.

4.6.1. Camera motion

Warping the dynamic video panorama back to the coordinate system of the original input footage can be done by simply applying the inverse of the homographies $\{\mathbf{H}_{r,i}\} = \{\mathbf{H}_{i,r}^{-1}\}$, computed during the registration step, to each respective

frame of the dynamic panorama. If we want to control the rotational motion performed by the virtual camera, we first need to recover the original camera motion. To achieve this, we will need to calibrate the camera, seperating the intrinsic and extrinsic camera parameters. Assuming that not all rotations are about the same axis, we can linearly decompose the homographies $\mathbf{H}_{r,i}$ as described by Agapito *et al.* [AHH99]:

$$\mathbf{H}_{r,i} = \mathbf{K}_i \mathbf{R}_{r,i} \mathbf{K}_r^{-1} \tag{20}$$

Pre-multiplying or replacing rotation matrix $\mathbf{R}_{r,i}$ with a user-controlled rotation **R'** will allow direct access to the virtual camera.

$$\mathbf{H'}_{r,i} = \mathbf{K}_i \mathbf{R'} \mathbf{R}_{r,i} \mathbf{K}_r^{-1} \tag{21}$$

Also, as we know the translations of all pixels for each pair of neighboring frames, we have the option of adding blur to pixels that were not part of the original input video. This can easily be achieved by convoluting the selected pixels with an appropriate kernel.

4.6.2. Field of View

Besides the ability to control camera motion, we also allow the user to adjust the field of view. This brings up several complications: (a) by expanding the field of view, empty parts of the panorama can become visible when the camera reaches the edge of the panorama, and (b) when the rotation of virtual camera is adjusted, parts of the foreground element(s) may no longer be visible. To cope with these situations, we iteratively adjust the rotation **R'** and if needed the focal length of $\mathbf{K}_i$. During this computation, we treat the absence of gaps in our output frame as a hard, and the visibilty of the actors as a soft constraint.

5. Results

We have applied our algorithm to a variety of input sequences, chosen specifically to test individual components of our algorithm. For example, in order to test our registration algorithm, a skateboarding sequence with recurring loops in the frame topology was computed. Our waterfall scene contains both structured (miniature water wheel) and unstructured (waterfall) dynamic background elements.

In general, the augmentation of the original video sequences generates convincing results (depicted in fig.7). Careful examination however will reveal occasional artefacts, in the form of 'popping' effects. These artefacts are usually the product of aperiodic background elements, or background elements without a full visible cycle, labeled as dynamic background. Their temporal continuity ends abruptly, resulting in the popping artefact.

Figure 6: *Comparison of (a) temporal mean filtering, (b) temporal median filtering, and (c) our background estimation algorithm.*

5.1. Discussion

The automatic registration of the video frames consistently provides us with accurate results, unless the underlying inter-frame warping procedure breaks down. This happens in two cases: (a) when comparing a severely blurred image with an undistorted one, and (b) when dealing with stochastic dynamic regions filling nearly the entire input image. We will look into the recent work of Yuan *et al.* [YSQS07] to deal with the first issue, but we are unaware of any methods that can deal with the second.

Our static background estimation component produces high-quality static panoramas, under the assumptions that parallax effects stay within reasonable bounds and that all sharp background pixels are visible at least once within the entire sequence. A comparison of our technique to standard background subtraction methods is shown in fig.6. It should be noted that this stage can be used as a stand-alone application for background estimation in cluttered scenes.

Our dynamic background estimation component generates convincing results, with the exception of the popping artefacts which we mentioned earlier. However, there are some limitations that need to be taken into account when applying our technique, e.g. BP algorithms tend to use large amounts of memory. This requires us to take several measures to make sure our algorithm does not unnecessarily squander its resources. Whereas precomputing the single-node potentials relieves us from storing binary masks and segmentation information, label clustering and label pruning [KT06] reduce the amount of pairwise potentials that needs to be computed. It should be noted that the pairwise potentials only depend on intensity information stored in the labels. As a result, in case the amount of labels is sufficiently reduced in number, it is possible to pre-compute and store

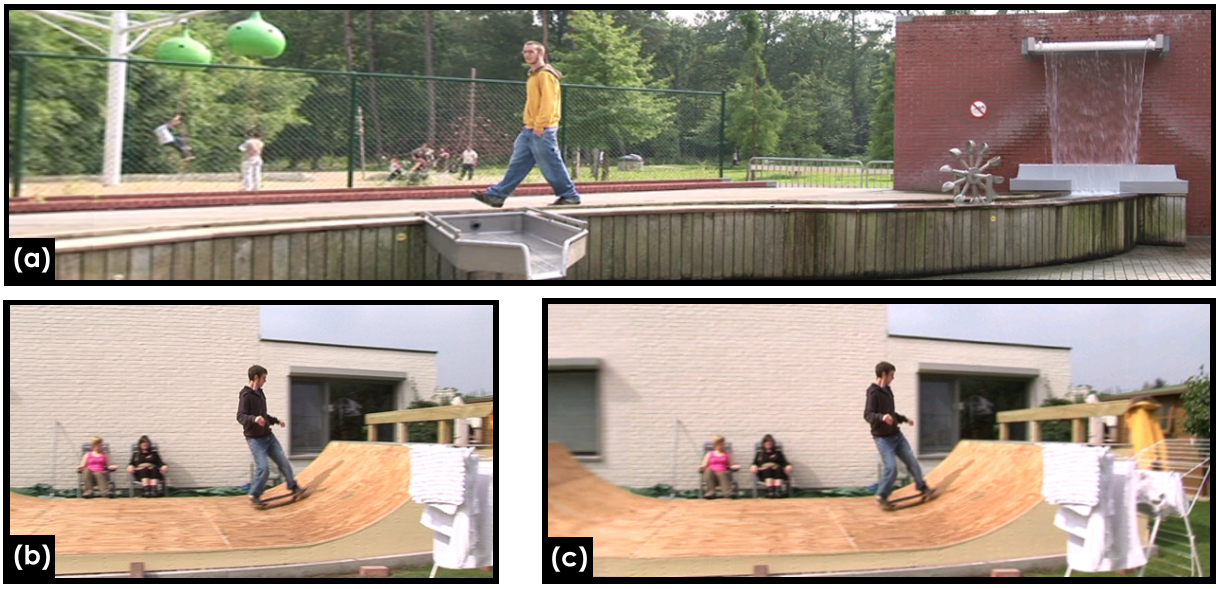

Figure 7: *(a) A cropped panorama frame from our waterfall scene. (b-c) An input frame of our skateboarding sequence, and the processed frame with an expanded field of view. Motion blur has been added in an additional post-processing step.*

all pairwise potentials in memory, without the need to retain the intensity images themselves. Storing the potentials in memory also reduces the required computation times from a number of days to a few hours, depending on the scene.

5.2. Future work

During our experiments, we have warped all video frames to the image plane of the references frame. This effectively reduces the resolution of the background pixels in the outer regions of our background panorama. In the future, we would like to test the effectiveness of our approach on other parametrizations of the scene intensities, such as cylindrical or spherical pixel coordinates.

Another interesting area for future work could be devising a hierarchical approach to our dynamic background estimation procedure. Building on the results from a lower resolution level, we might be able to narrow down the number of candidate labels for each new iteration.

6. Conclusion

Besides presenting the idea of editing panning/rotating video sequences using a full panoramic representation, we present a robust video mosaicing algorithm that produces high quality panoramas without parallax artefacts, seams or blurring, while retaining repetitive dynamic elements. Our technique allows the user to control the camera of a panning/rotating video in a post-processing step, allowing for a seamless change of aspect ratio or camera motion path. It also facilitates other post-processing steps such as adding motion blur or video stabilization.

7. Acknowledgements

The authors acknowledge financial support on a structural basis from the European Regional Development Fund (ERDF) and the Flemish Government. Part of this research was funded by the BOF-projectfund of Hasselt University. Part of the work is also funded by the European research project IST-2-511316-IP : IP-RACINE (Integrated Project Research Area CINE). Furthermore we would like to thank our colleagues for their help and inspiration, in particular Tom Haber for his helpful set of C++ libraries. Finally, we would like to thank Bart Bulen for shooting our video sequences.

References

[AHH99] AGAPITO L., HARTLEY R., HAYMAN E.: Linear calibration of a rotating and zooming camera. In *CVPR '99: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1999), pp. 15–21.

[AS07] AVIDAN S., SHAMIR A.: Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10.

[AZP*05] AGARWALA A., ZHENG K. C., PAL C., AGRAWALA M., COHEN M., CURLESS B., SALESIN D., SZELISKI R.: Panoramic video textures. *ACM Trans. Graph. 24*, 3 (2005), 821–827.

[BHN07] BROWN M., HARTLEY R. I., NISTER D.: Minimal solutions for panoramic stitching. In *CVPR '07: Proceedings of the 2007 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (june 2007), pp. 1–8.

[BM03] BENNETT E. P., MCMILLAN L.: Proscenium: a framework for spatio-temporal video editing. pp. 177–184.

[CFM06] COLOMBARI A., FUSIELLO A., MURINO V.: Background initialization in cluttered sequences. *5th Workshop on Perceptual Organization in Computer Vision* (2006), 197.

[CPT03] CRIMINISI A., PEREZ P., TOYAMA K.: Object removal by exemplar-based inpainting. In *CVPR '03: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2003), pp. II: 721–728.

[DCOY03] DRORI I., COHEN-OR D., YESHURUN H.: Fragment-based image completion. *ACM Trans. Graph. 22*, 3 (2003), 303–312.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision - Volume 2* (1999), p. 1033.

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM archive 24* (1981), 381 – 395.

[Har97] HARTLEY R. I.: In defense of the eight-point algorithm. *IEEE Transactions On Pattern Analysis And Machine Intelligence 19*, 6 (1997), 580–593.

[HRRS86] HAMPEL F., RONCHETTI E., ROUSSEEUW P., STAHEL W.: *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 1986.

[iKO99] ICHI KANATANI K., OHTA N.: Accuracy bounds and optimal computation of homography for image mosaicing applications. In *ICCV (1)* (1999), pp. 73–78.

[JMF99] JAIN A. K., MURTY M. N., FLYNN P. J.: Data clustering: a review. *ACM Computing Surveys 31*, 3 (1999), 264–323.

[KCM00] KANG E., COHEN I., MEDIONI G.: A graph-based global registration for 2D mosaics. In *Proceedings of lnternational Conference on Pattern Recognition 2000* (2000), pp. 257–260.

[KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. *ACM Trans. Graph. 24*, 3 (2005), 795–802.

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (2003), 277–286.

[KT06] KOMODAKIS N., TZIRITAS G.: Image completion using global optimization. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), pp. 442–452.

[LG06] LIU F., GLEICHER M.: Video retargeting: automating pan and scan. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia* (2006), pp. 241–250.

[LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *IJCAI '81: Proceedings of the 7th International Joint Conference on Artificial Intelligence* (April 1981), pp. 674–679.

[LKK03] LITVIN A., KONRAD J., KARL W. C.: Probabilistic video stabilization using kalman filtering and mosaicking. In *IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Proc.* (September 2003).

[MFM04] MARZOTTO R., FUSIELLO A., MURINO V.: High resolution video mosaicing with global alignment. 692–698.

[MOTS05] MATSUSHITA Y., OFEK E., TANG X., SHUM H.-Y.: Full-frame video stabilization. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 50–57.

[SHK98] SAWHNEY H. S., HSU S., KUMAR R.: Robust video mosaicing through topology inference and local to global alignment. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II* (1998), pp. 103–119.

[SYJS05] SUN J., YUAN L., JIA J., SHUM H.-Y.: Image completion with structure propagation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (2005), pp. 861–868.

[TK91] TOMASI C., KANADE T.: *Detection and Tracking of Point Features*. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.

[WSI04] WEXLER Y., SHECHTMAN E., IRANI M.: Space-time video completion. In *CVPR '04: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2004), vol. 1, pp. 120–127.

[YFW01] YEDIDIA J. S., FREEMAN W. T., WEISS Y.: Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence 2001 Distinguished Lecture Track* (2001).

[YSQS07] YUAN L., SUN J., QUAN L., SHUM H.-Y.: Blurred/non-blurred image alignment using sparseness prior. In *ICCV '07: Proceedings of the International Conference on Computer Vision* (2007).