# Online Registration of Dynamic Scenes using Video Extrapolation

Alex Rav-Acha Yael Pritch Shmuel Peleg
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
E-Mail: {alexis,yaelpri,peleg}@cs.huji.ac.il

## Abstract

*An online process is proposed for video registration of dynamic scenes, such as scenes with dynamic textures or with moving objects. This process has three steps: (i) A few frames are assumed to be already registered. (ii) Using the registered frames, the next new frame is extrapolated. (iii) The actual new frame is registered to the extrapolated frame.*

*Video extrapolation overcomes the bias introduced by dynamics in the scene, even when the dynamic regions cover almost the entire image. It can also overcome not only motion, but also many fluctuations in intensity. The traditional "brightness constancy" is now replaced with "dynamics constancy".*

## 1 Introduction

When a video sequence is captured by a moving camera, motion analysis is required for many video editing and video analysis applications. Most methods for image alignment assume that a dominant part of the scene is static, and also assume brightness constancy. These assumptions are violated in scenes with moving objects or with dynamic background, cases where most registration methods will likely fail.

A pioneering attempt to perform motion analysis in dynamic scenes was suggested in [6]. In this work, the entropy of an auto regressive process was minimized with respect to the motion parameters of all frames. But the implementation of this approach may be impractical for many real scenes. First, the auto regressive model is restricted to scenes which can be approximated by a stochastic process, and it can not handle dynamics such as walking people. In addition, in [6] the motion parameters of all frames are computed simultaneously, resulting in a difficult non-linear optimization problem. Moreover, extending this method to cases with multiple dynamic textures requires segmenting the scene into its different dynamic textures [10]. With our proposed approach, no segmentation is needed.

Unlike computer motion analysis, humans can easily distinguish between the motion of the camera and the internal dynamics in the scene. For example, we can virtually align an un-stabilized video of a sea, even when the waves are constantly moving. The key to this human ability is an assumption regarding the simplicity and consistency of scenes and of their dynamics: It is assumed that when a video is aligned, the dynamics in the scene become smoother and more predictable. This allows humans to track the motion of the camera even when no apparent registration information exists. We therefore try to replace the "brightness constancy assumption" with a "dynamics constancy assumption".

This dynamic constancy assumption is used as a basis for our online registration algorithm: given a new frame of the sequence, it is aligned to best fit the extrapolation of the preceding frames. The extrapolation is done using video synthesis techniques [11, 5, 8], and the alignment is done using traditional methods for parametric motion computation [2, 7]. Alternating between video extrapolation and image alignment results in a robust online registration algorithm which can handle complex scenes, having both dynamic textures and moving objects.

There is a major difference between the video extrapolation step in our approach and previous results on video completion or on dynamic texture synthesis. Creating a good looking video, as is the goal in video completion or synthesis, is not only difficult, but also creates a video which deviates from the original data. In our case we use the video extrapolation only for motion computation. While this requires that many image regions will be correctly extrapolated, other regions may not be extrapolated at all.

## 2 Video Alignment with Dynamic Scenes

Video motion analysis traditionally aligns two successive frames. This approach may work well for static scenes, where one frame can predict the next frame up to their global relative motion. But when the scenes are dynamic, the global motion between the frames is not enough to predict the successive frame, and global motion analysis between such two frames is likely to fail. We propose to replace the assumptions of static scenes and brightness constancy with a much more general assumption of consistent image dynamics: "What happened in the past is likely to happen in the future". In this section we will describe how a video be extrapolated using this assumption, and how this extrapolation can be used for image alignment.

### 2.1 Dynamics Constancy Assumption

Let a video sequence consist of frames $I_1 \dots I_N$. A space-time volume $V$ is constructed from this video sequence by stacking all the frames along the time axis, $V(x, y, t) = I_t(x, y)$. The "dynamics constancy" assumption implies that when the volume is aligned (e.g., when the camera is static), we can estimate a large portion of each image $I_n = V(x, y, n)$ from the preceding frames $I_1 \dots I_{n-1}$. We will denote the space-time volume constructed by all the frames up to the $k^{th}$ frame by $V(x, y, \overrightarrow{k})$. According to the "dynamics constancy" assumption, we can find an extrapolation function over the preceding frames such that

$$I_n(x, y) = V(x, y, n) \approx Extrapolate(V(x, y, \overrightarrow{n-1})). \quad (1)$$

$Extrapolate$ is a non parametric extrapolation function, estimating the value of each pixel in the new image given the preceding space-time volume. This extrapolation should use the dynamics constancy assumption, and will be described in the next section.

When the camera is moving, the image transformation induced by the camera motion should be added to this equation. Assuming that all frames in the space time volume $V(x, y, \overrightarrow{n-1})$ are aligned to the coordinate system of the $(n-1)^{th}$ frame, the new image $I_n(x, y)$ can be approximated by

$$I_n \approx T_n(Extrapolate(V(x, y, \overrightarrow{n-1}))). \quad (2)$$

$T_n$ is a $2D$ image transformation between frames $I_{n-1}$ and $I_n$, and is applied on the extrapolated image. Applying the inverse transformation on both sides of the equation gives

$$T^{-1}(I_n) \approx Extrapolate(V(x, y, \overrightarrow{n-1})). \quad (3)$$

This relation is used in the registration scheme.

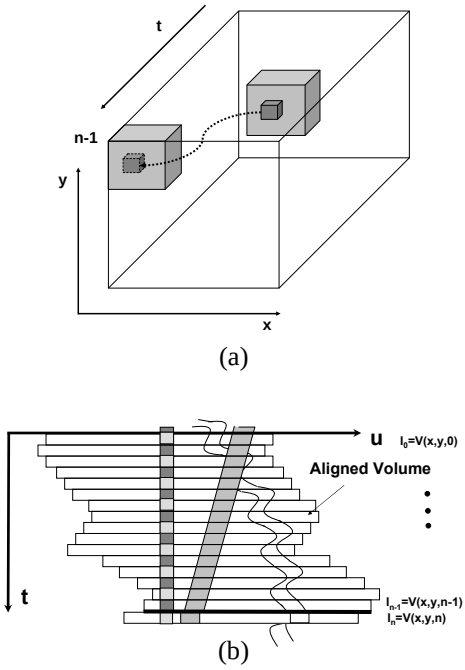


**Figure 1.** Video Extrapolation using a Space-Time Block Search. Both motion and intensity variation are accounted for.
(a) For all blocks bordering with time $(n-1)$, a best matching block is searched in the space-time volume. Once such a block is found, the pixel in front of this block is copied to the corresponding position in the extrapolated frame $I_n^p(x, y)$.
(b) The new frame $I_n$ is not aligned to Frame $I_{n-1}$, but to the frame that has been extrapolated from the preceding space-time volume.

### 2.2 Video Extrapolation

Our video extrapolation is closely related to dynamic texture synthesis [4, 1]. However, dynamic textures are characterized by repetitive stochastic processes, and do not apply to more structured dynamic scenes, such as walking people. We therefore prefer to use non-parametric video extrapolation methods [11, 5, 8]. These methods assume that each small space-time block has likely appeared in the past, and thus the video can be extrapolated using similar blocks from earlier video portions. This is demonstrated in Fig. 1. Various video interpolation or extrapolation methods differ in the way they enforce spatio-temporal consistency of all blocks in the synthesized video. However, this problem is not important in our case, as our goal is to achieve a good alignment rather than a pleasing video.

Leaving out the spatio-temporal consistency requirement, we are left with the following simple video extrapolation scheme: Assume that the aligned space time volume

$V(x, y, \overrightarrow{n-1})$ is given, and a new image $I_n^p$ is to be estimated. For each pair of space-time blocks $W_p$ and $W_q$ we define the SSD (sum of square differences) to be:

$$d(W_p, W_q) = \sum_{(x,y,t)} (W_p(x, y, t) - W_q(x, y, t))^2. \quad (4)$$

As shown in Fig. 1, for each pixel $(x, y)$ in image $I_{n-1}$ we define a space-time block $W_{x,y,n-1}$ whose spatial center is at pixel $(x, y)$ and whose temporal boundary is at time $n-1$ (future frames can not be used in an online approach). We then search in the space time volume $V(x, y, \overrightarrow{n-2})$ for a space-time block with the minimal SSD to block $W_{x,y,n-1}$. Let $W_p = W(x_p, y_p, t_p)$ be the most similar block, spatially centered at pixel $(x_p, y_p)$ and temporally bounded by $t_p$. The value of the extrapolated pixel $I_n^p(x, y)$ will be taken from $V(x_p, y_p, t_p+1)$, the pixel that appeared immediately after the most similar block. This scheme follows the "dynamics constancy" assumption: given that two different space time blocks are similar, we assume that their continuations are also similar. While a naive search for each pixel may be exhaustive, several accelerations can be used as described in Sec. 2.6.

We used the SSD (sum of square differences) as a distance measure between two space-time blocks, but other distance measures can be used such as the sum of absolute differences or more sophisticated measures ([11]). We did not notice a substantial difference in registration results.

### 2.3 Alignment with Video Extrapolation

The online registration scheme for dynamic scenes uses the video extrapolation described earlier. As already mentioned, we assume that the image motion of a few frames can be estimated with traditional robust image registration methods [9, 7]. Such initial alignment is used as "synchronization" for computing the motion parameters of the rest of the sequence. Alignment with Video Extrapolation can be described by the following steps:

1. Assume that the motion of the first $K$ frames has already been computed, and let $n = K + 1$.
2. Align all frames in the space time volume $V(x, y, \overrightarrow{(n-1)})$ to the coordinate system of Frame $I_{n-1}$.
3. Estimate the next new image by extrapolation from the previous frames $I_n^p = Extrapolate(V(x, y, \overrightarrow{(n-1)}))$.
4. Compute the motion parameters (The global $2D$ image transformation $T_n^{-1}$) by aligning the new input image $I_n$ to the extrapolated image $I_n^p$.
5. Increase $n$ by 1, and return to Step 2. Repeat until reaching the last frame of the sequence.

The global 2D image alignment in Step 2 is performed using direct methods for parametric motion computation [2, 7]. Outliers are marked during this alignment as described in the next section.

### 2.4 Masking Unpredictable Regions

Real scenes always have a few regions that can not be predicted. For example, people walking in the street often change their behavior in an unpredictable way, e.g. raising their hands or changing their direction. In these cases the video extrapolation will fail, resulting in outliers. The alignment can be improved by estimating the predictability of each region, where unpredictable regions get lower weights during the alignment stage. To do so, we incorporate a predictability score $M(x, y, t)$ which is estimated during the alignment process, and is later used for future alignment.

The predictability score $M$ is computed is the following way: After the new input image $I_n$ is aligned with the extrapolated image $I_n^p$ which estimated it, the difference between the two images is computed. Each pixel $(x, y)$ receives a predictability score according to the color differences in its neighborhood. Low color differences indicates that the pixel has been estimated accurately, while large differences indicate poor estimation. From these differences a binary predictability mask is computed, indicating the accuracy of the extrapolation,

$$M(x, y, n) = \begin{cases} 1 & if \frac{\sum (I_n - I_n^p)^2}{\sum I_x^2 + I_y^2} < r \\ 0 & otherwise, \end{cases} \quad (5)$$

where the summation is over a window around $(x, y)$, and $r$ is a threshold (We usually used $r = 1$). This is a conservative scheme to mask out pixels in which the residual energy will likely bias the registration. The predictability mask $M_n(x, y) = M(x, y, n)$ is used in the alignment of frame $I_{n+1}$ to frame $I_{n+1}^p$.

### 2.5 Fuzzy Estimation

Applications such as video completion or video compression also use extensively frame predictions. Unlike these applications, video registration is not limited to use a single prediction. Instead, better alignment can be obtained when a fuzzy prediction is used. The fuzzy prediction can be obtained by keeping not only the best candidate for each pixel, but the best $S$ candidates (We used up to five candidates for each pixel). The multiple predictions for each pixel can easily be combined using a summation of the error terms:

$$T_n = \arg\min_T \{ \sum_{x,y,s} \lambda_{x,y,s} (T^{-1}(I_n)(x, y) - I_n^p(x, y, s))^2 \} \quad (6)$$

where $I_n^p(x,y,s)$ is the $s^{th}$ candidate for the value of the pixel $I_n(x,y)$. The weight $\lambda_{x,y,s}$ of each candidate is based on the difference of its corresponding space-time cube from the current one as defined in Eq. 4, and is given by:

$$\lambda_{x,y,s} = e^{\frac{-d(W_p,W_q)^2}{2\sigma^2}}.$$

We used $\sigma = 1/255$ to reflect the noise in the image gray-levels. Note that the weights for each pixel do not necessarily sum to one, and therefore the registration mostly relies on the most predictable regions.

### 2.6 Accelerating the Video Extrapolation

The most expensive stage of the dynamic registration is finding the best candidates in the video extrapolation stage. An exhaustive search makes this stage very slow. To enable fast extrapolation we have implemented several modifications which accelerate substantially this stage. Some of these accelerations may not be valid for general video synthesis and completion techniques, as they can reduce the rendering quality of the resulting video. But high rendering quality is not essential for accurate registration.

**Limited Search Range:** Video sequences can be very long, and searching the entire history may not be practical. Moreover, the periodicity of most objects is usually of a short time period. We have therefore limited the search for similar space-time cubes to a small volume in both time and space around each pixel. Typically, we searched up to 10-20 frames backwards (periods of approximately one second).

**Using Pyramids:** We assume that the spatio-temporal behavior of objects in the video can be recognized even in a lower resolution. Under this assumption, we construct a Gaussian pyramid for each image in the video, and use a multi-resolution search for each pixel. Given an estimate of a matching cube from a lower resolution level, we search only in a small spatial area in the higher resolution level. The multi-resolution framework allows to search in a wide spatial range and to compare small space-time cubes.

**Summed Area Tables:** Since the video extrapolation uses a sum of squares of values in sub-blocks in both space and time (See Eq. 4), we can use summed-area tables [3] to compute all the distances for all the pixels in the image in $O(N \cdot S_x \cdot S_y \cdot S_t)$ where $N$ is the number of pixels in the image, and $S_x$, $S_y$ and $S_t$ are the search ranges in the $x$,$y$ and $t$ directions respectively. This saves the factor of the window size (Typically $5 \times 5 \times 5$) over a direct implementation. This step cannot be used together with the multi-resolution search, as the lookup table changes from pixel to pixel, but it can still be used in the lowest resolution level, where the search range is the largest.

**Figure 2.** The water flow in the input movie (top), as well as the moving pinguin, create a difficult scene for alignment. The video was registered using extrapolation, an was compared to regular alignment. An average of 40 frames in the stabilized sequence is shown. Using a traditional 2D parametric alignment the sequence is very unstable, and the average image is very blurry (lower left). With video extrapolation the registration is much better (lower right).

### 2.7 Handling Alignment Drift

Alignment based on Video Extrapolation follows Newton's First Law: An object in uniform motion tends to remain in that state. If we initialize our registration algorithm with a small motion relative to the real camera motion, our method will continue this motion for the entire video. In this case the background will be handled as a slowly moving object. This is not a bug in the algorithm, but rather a degree of freedom resulting from the "dynamics constancy" assumption.

To eliminate this degree of freedom we incorporate a prior bias, and assume that some of the scene is static. This is done by aligning the new image to both the extrapolated image and the previous image, giving the previous image a low weight. In our experiments we gave a weight of 0.1 to the previous frame and a weight of 0.9 to the extrapolated frame. This prior prevented the possible drift, while not reducing the accuracy of motion computation.

## 3 Examples

In this section we show various examples of video alignment for dynamic scenes. A few examples are also compared to regular direct alignment as in [2, 7]. The computed alignment was used for video stabilization, and the stabilized sequences are best seen in the web site: *http://www.vision.huji.ac.il/dynreg*. To show stabilization results in print, we have averaged the frames of the stabilized video. When the video is stabilized accurately, static

**Figure 3.** In the original video (top) the water and the bear are dynamic, while the rocks are static. Average images of 40 frames are shown, with traditional 2D parametric alignment (lower left) and with our proposed method (lower right). The sharper average shows the superiority of our method.

regions appear sharp while dynamic objects are ghosted. When stabilization is erroneous, both static and dynamic regions are blurred.

Figures 2 and 3 compare the registration using video extrapolation with traditional direct alignment [2, 7]. Both scenes include moving objects and flowing water, and a large portion of the image is dynamic. In spite of the dynamics, after video extrapolation the entire image can be used for the alignment. For this comparison, in these examples we did not use any mask to remove unpredictable regions nor did we use a fuzzy estimation, but rather used the entire image for the alignment.

The sequence shown in Figure 4 was used by [10] and by [6] as an example for their registration of dynamic textures. The global motion in this sequence is a horizontal translation, and the true displacement can be computed from the motion of one of the flowers. The displacement error reported by [10] was $29.4\%$ of the total displacement between the first and last frames, while the error of our methods was only $1.7\%$.

Figures 5 and 6 show two more examples of video registration using extrapolation in challenging scenes. In these scenes, the estimation of some of the regions was not good enough (Parts of the falls and the fumes in the 'waterfall'

**Figure 4.** A sequence of moving flowers taken by a panning camera. See *http://www.robots.ox.ac.uk/∼awf/iccv01/*. Our motion computation with video extrapolation gave an accumulated translation error of 1.7% between the first and last frames, while [10] reported an accumulated error of $29.4\%$.

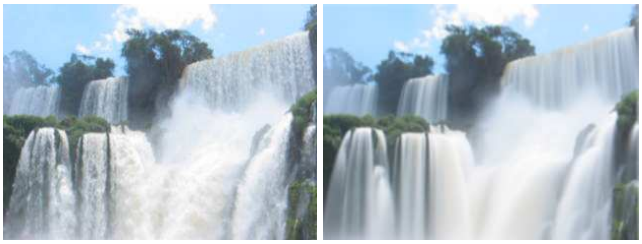

**Figure 5.** This waterfall sequence (left) poses a challenging task for registration, as most of the scene is covered with falling water. The video was stabilized using video extrapolation (using a rotation and translation motion model). An average of 40 frames in the stabilized video (right) is shown to evaluate the quality of the stabilization. The dynamic regions are blurred only in the flow direction, while the static regions remain relatively sharp after averaging.

video, and some actions in the 'festival' video), so predictability masks (as described in Section 2.4) were used to exclude unpredictable regions from the motion computations.

## 4 Concluding Remarks

An approach for video registration of dynamic scenes has been presented. The dynamics in the scene can be either stochastic as in dynamic textures, or structured as in moving people. Intensity changes such as flickering can also be addresses. The frames in such video sequences are aligned by estimating the next frame using video extrapolation from the preceding frames.

Video extrapolation for alignment can be done much faster than other video completion approaches, resulting in a robust and efficient registration. The examples show excellent registration for very challenging dynamic images that were previously considered impossible to align.

Most methods which address videos with multiple dy-

**Figure 6.** While the dynamic crowd in the Edinburgh festival makes alignment a real nightmare, alignment using video extrapolation had no problems. Three original frames are shown at the top. The panorama is stitched from the video after the alignment by frame averaging. The scene dynamics is visible by ghosting, and the static background is clearly well registered.

namic patterns use a segmentation of the scene. Due to its non parametric nature, the proposed approach can find the motion parameters without any segmentation.

The proposed video extrapolation is different from Image prediction used for video compression in the following aspects:

- The main objective of the video extrapolation in our case is to minimize the motion bias rather than the prediction error.
- An estimation of the gray-values at a sparse set of image locations is sufficient for accurate registration, while it is not applicable for compression.
- Unlike video compression methods which compute the optical flow between current and previous frames, our video extrapolation does not use the current frame. This is due to the fact that such an optical flow would mix between the camera motion and the scene dynamics.

A possible future challenge can be the development of a registration scheme in the presence of both motion parallax and scene dynamics. This combination is not simple, as motion parallax depends on the dynamic of the camera, which has no relation to the dynamic of the scene.

## References


[1] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M.Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans. Visualization and Computer Graphics*, 7(2):120–135, 2001.

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV'92)*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.

[3] F. C. Crow. Summed-area tables for texture mapping. In *SIGGRAPH '84*, pages 207–212, 1984.

[4] G. Doretto, A. Chiuso, S. Soatto, and Y. Wu. Dynamic textures. *IJCV*, 51(2):91–109, February 2003.

[5] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038, Corfu, 1999.

[6] A. Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. In *International Conference on Computer Vision (ICCV'01)*, volume I, pages 662–669, Vancouver, Canada, July 2001.

[7] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *International Conference on Computer Vision (ICCV'88)*, pages 959–966, Bombay, India, January 1998.

[8] V. Kwatra, A. Schdl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.

[9] P. Meer, D. Mintz, D. Kim, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[10] R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamic textures. In *CVPR*, pages 516–521, San Diego, USA, June 2005.

[11] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 120–127, Washington, DC, June 2004.