Online Video Registration of Dynamic Scenes using Frame Prediction

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 approach is proposed for Video registration of dynamic scenes, such as scenes with dynamic textures, moving objects, motion parallax, etc. This approach has three steps: (i) Assume that a few frames are already registered. (ii) Using the registered frames, the next frame is predicted. (iii) A new video frame is registered to the predicted frame.

Frame prediction overcomes the bias introduced by dynamics in the scene, even when dynamic objects cover the majority of the image. It can also overcome many systematic changes in intensity, and the "brightness constancy" is replaced with "dynamic constancy".

This predictive online approach can also be used with motion parallax, where non uniform image motion is caused by camera translation in a 3D scene with large depth variations. In this case a method to compute the camera ego motion is described.

Unlike video synthesis methods that generate dynamic video for display purposes, predictive alignment and registration can be done quickly and efficiently.

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 many natural scenes, which consist of moving objects and dynamic background, cases where most registration methods are likely to fail.

A pioneering attempt to deal with dynamic scenes was suggested in [8]. In his 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 deal with dynamics such as walking people. In addition, in [8] the motion parameters of all frames are computed simultaneously, resulting in a difficult non-linear optimization problem.

Unlike computer motion analysis, humans can distinguish easily 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 moving with the wind. The key to this human ability is an assumption regarding to the simplicity and predictability of a natural scene and of its 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 predicability 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 prediction generated from the preceding frames. The prediction is done using video synthesis techniques [15, 7, 11], and the alignment is done using common methods for parametric motion computation [2, 9]. Alternating between prediction and registration 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 prediction step in our approach and previous work on video completion or on dynamic textures. In these approaches the goal was to create a good looking video. Making a video to look good is not only difficult, but also makes the video less faithful to the original data. In our case we use the prediction only for motion computation. While this requires that many image regions will be correctly predicted, other regions may not be predicted accurately. In general the predicted image

does not have to look "perfect", and the prediction process allows us to use simpler and faster prediction schemes, as will be explained in more details in Sec. 2. Even when the frame prediction step does not give a perfect prediction of the next frame, the registration algorithm can still find the correct image motion since the error is mostly unbiased

Predictive alignment assumes that out of the entire video sequence, a few frames can be registered using existing methods. E.g. a few frames where the camera was static, or when enough static objects exist. The initial alignment is used as "synchronization", and the motion parameters of the remaining frames are computed using the proposed predictive alignment scheme.

An even more accurate video prediction can be made when a model for the scene dynamics is available. An example for such a model is motion parallax. In this case the video sequence will be represented in a space-time volume (or an *epipolar volume*), constructed by stacking all input images into an *x-y-t* volume (as was introduced by Bolles et. al. [3]). Frame prediction is possible in the space time volume, since when the camera moves at a constant velocity, image points move on straight lines in the space-time volume. Extending these straight lines according to the motion of the camera is a good prediction for the next frame.

The predictive approach to motion parallax can also be extended to handle $2D$ camera translations and also camera rotations. Setups describing camera motions which are applicable to this work are shown in Fig. 6. These cases can be used for view synthesis [14].

2 Video Alignment with Dynamic Scenes

Video motion analysis traditionally aligns two successive frames. This approach works well for static scenes, where one frame predicts the next frame up to their relative motion. But when the scenes are dynamic, the motion between the frames is not enough to predict the successive frame, and 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 the next frame can be predicted from prior images, and how this prediction can be used for image alignment.

2.1 Predictive Video 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 "consistent image dynamics" assumption implies that when the volume is aligned (e.g., when the camera is static), we can predict 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 "consistent image dynamics" assumption, we can find a prediction function over the preceding frames such that

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

$Predict$ is a non parametric extrapolation function, predicting the value of each pixel in the new image given the preceding space-time volume. This prediction should use the consistent image dynamics 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 predicted by

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

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

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

This relation is used in the predictive registration Scheme.

2.2 Next Frame Prediction

The prediction of the next frame given the aligned space-time volume of preceding frames is closely related to dynamic texture synthesis [6, 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 [15, 7, 11] for prediction. These methods assume that each small space-time block has likely appeared in the past, and thus a new image can be predicted by 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 for prediction, as our goal is to achieve a good alignment rather than a pleasing video.

Leaving out the spatio-temporal consistency requirement, we are left we the following simple video completion 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 predicted. We use the SSD (sum of square differences) as a

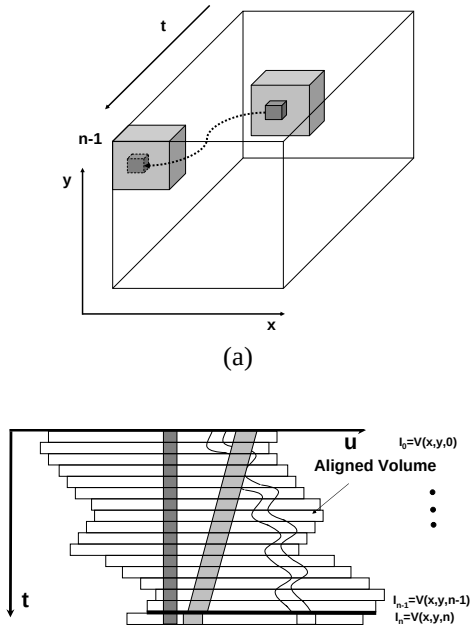


Figure 1. Frame Prediction using Space-Time Block Search
(a) For all blocks bordering with time $(n-1)$, a best matching block is searched in the space-time volume. Once such a block is found, the pixel in front of this block is copied to the corresponding position in the predicted frame $I_n^p(x,y)$ (b) The new frame I_n is not aligned to Frame I_{n-1}, but to a predicted frame that can be computed from the preceding space-time volume.

distance between space-time blocks. The distance d between each pair of space-time blocks W_p and W_q is given by,

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

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 predicted 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 prediction follows the "consistent image dynamics" assumption: given that the two space time blocks are similar, we assume that their continuations are also similar. While a naive search for each predicted pixel may be exhaustive, several accelerations can be used as described in Sec. 2.5.

2.3 The Predictive Registration Scheme

The online registration scheme for dynamic scenes uses the predictions described earlier. As already mentioned, we assume that the image motion of a few frames can be estimated with traditional robust image registration methods [13, 9]. Such initial alignment is used as "synchronization" for computing the motion parameters of the rest of the sequence. In the following we assume that the motion of the first K frames has already been computed. The predictive registration scheme can be described by the following steps:

1. Let $n = K + 1$.
2. Align all frames in the space time volume $V(x,y,\overrightarrow{(n-1)})$ to the coordinate system of the frame I_{n-1}.
3. Predict the next image of the sequence given the previous frames $I_n^p = Predict(V(x,y,\overrightarrow{(n-1)}))$.
4. Compute the motion parameters (The $2D$ image transformation T_n^{-1}) by aligning the new input image I_n to the prediction I_n^p.
5. Increase n by 1, and return to Step 2. Repeat until reaching the last frame of the sequence.

The 2D image alignment in Step 2 is performed using direct methods for parametric motion computation [2, 9]. 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 prediction 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: Given that the input image I_n and its prediction I_n^p are aligned, the difference between the two images is computed, and each pixel (x, y) receives a predictability score according to the frame differences around this pixel. From this we compute a binary predictability mask which measures the bias of the prediction,

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

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 Accelerating Computation of Frame Prediction

The most expensive stage of the predictive alignment is the prediction stage. In a naive implementation an exhaustive search is used, making this stage very slow. To enable fast prediction we have implemented several modifications which accelerate substantially this stage. Some of these accelerations are not valid for general video synthesis and completion techniques, as they may reduce the rendering quality of the prediction. But rendering quality can be sacrificed for registration.

Limited Search Range: Video segments may be very long, and searching the entire history is impractical. Moreover, the periodicity of most objects is usually of a short 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.

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 constructed a Gaussian pyramid for each image in the video, and used a multi-resolution search for each pixel. Given an estimate of the best matching cube from a lower resolution level, we search only a small spatial area in the higher resolution level. The multi-resolution framework allows to search a wide spatial range and to compare small space time cubes.

Summed Area Tables: Since the prediction uses a sum of squares of values in sub-blocks in both space and time (See Eq. 4), we can use summed-area tables [5] 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 highest resolution level, where the search range is the largest.

2.6 Handling Alignment Drift

Predictive alignment 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, predictive registration 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 'predictive video assumption', as there is no doubt that a constant moving scene is a predictable one. To reduce 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 predicted 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 predicted frame.

3 Examples: Video Registration of Dynamic Scenes

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, 9]. The alignment was used for video stabilization, and the results are best seen in the enclosed video. To show stabilization results on paper, we have averaged the frames of the stabilized video. The average image of a stabilized video is sharp, while the average image of video which is not stabilized is blurred.

Figures 2 and 3 compare predictive registration to a traditional direct alignment [2, 9]. Both scenes include moving objects and flowing water, and a large portion of the image is dynamic. In spite of the dynamics, after prediction the entire image can be used for the alignment. In these examples we did not use any mask to remove unpredictable regions, and used the entire image for alignment.

Figures 4 and 5 show two more examples of apply-

Figure 2. The water flow in the input movie (up), as well as the moving pinguin, create a difficult scene for alignment. The video was registered using predictive alignment, 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 predictive alignment the registration is much better (lower right). Videos of the stabilized sequences, are included in the attached video.

ing predictive alignment to challenging scenes. In these scenes, the prediction of some of the regions was not good enough (Parts of the falls and the fumes in the 'waterfall' video, and some actions in the 'festival' video), so predictability masks (as described in Section 2.4) were used to exclude unpredictable regions from motion computation.

4 Video Alignment with Motion Parallax

When the camera's velocity and frame rate are constant, the time of frame capture is proportional to the location of the camera along the camera path. In this case, and for a static scene, the image features are arranged in an EPI plane (an *x*-*t* slice of the *x*-*y*-*t* volume) along straight lines, since the projections of each 3D point are only along a straight line in this plane [3]. Each straight line represents a different image feature corresponding to a point in the 3D world, and the slope of this line is inversely proportional to the depth of that point. Points at infinity, for example, will create straight lines parallel to the t axis, since their projection into the image is constant, and does not change with camera translation. Closer points move faster in the image, and the straight line representing them will have a small angle with the x axis.

The space time volume was used in [4] to differentiate between different depth layers in a video. Object were even removed from the scene, and the vacated space has been filled in using the straight-line property of the EPI lines.

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 the predictive alignment (lower right). The sharper average shows the superiority of predictive alignment. Videos of the stabilized sequences, are given in the attached video.

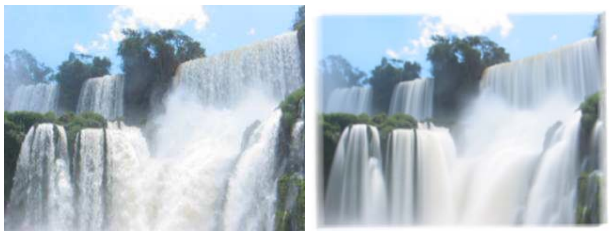


Figure 4. This waterfall sequence poses a challenging task for registration, as most of the scene is covered with falling water. The video was stabilized with predictive alignment (using a rotation and translation motion model). An average of 40 frames in the stabilized video is shown to evaluate the quality of the stabilization. The dynamic regions are blurred only in the flow direction, while the static region remain relatively sharp after averaging.

Figure 5. While the dynamic crowd in this festival makes alignment a real nightmare, predictive alignment had no problems. Three original frames are shown at the top. The panorama is stitched from the video after alignment by frame averaging. The scene dynamics is visible by ghosting, and the static background is clearly well registered.

We propose to use this straight-line property not only for filling-in video, but also for enabling image alignment even in presence of strong parallax.

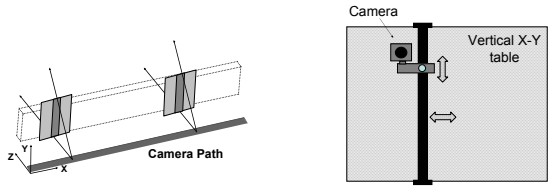
4.1 Prediction with Parallax

When the velocity of the camera varies, the time of frame capture is no longer proportional to the location of the camera. Image features are no longer arranged along straight lines in the EPI plane. The predictive approach to the computation of the camera motion assumes that a few frames are captured with a constant velocity. Only the correct camera motion can predict the next frame from the straight space-time lines computed for the preceding frames.

The Space-Time approach can also be extended to handle $2D$ camera translations and also camera rotations. Setups describing camera motions which are applicable the proposed analysis are shown in Fig. 6.

Alignment with parallax uses both motion parameters and shape parameters. The motion parameters are the translation and rotation of the camera, which vary for each frame. The shape parameters are represented by the slopes of the lines in the EPI plane for a camera translating along a straight line, or the slopes of the planes in the light field space [12] for a camera translating in a plane. The slopes of the lines and the planes in the EPI domain are inversely proportional to the depth of the corresponding 3D points, and thus they remain constant for each scene point at all frames.

To compute the locations of the optical centers of the



(a) (b)

Figure 6. Common setups for 1D and 2D camera motions.
(a) 1D motion - The camera moves along a straight line.
(b) 2D motion - Traditional light field capturing device. The camera can move to arbitrary locations along the u-v table.

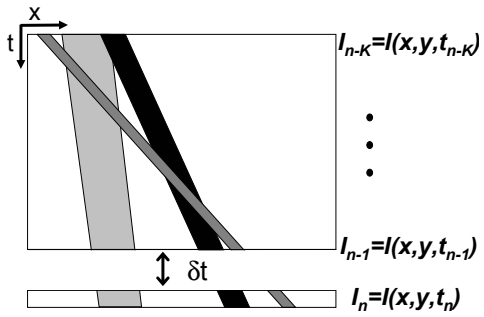


Figure 7. Given the shape parameters (EPI slopes), only the correct motion parameter t_n can predict the next frame I_n from the space-time volume.

input cameras, such that image features will reside on straight lines (or on planes), we use the following steps (to be detailed later):

1. Choose a frame I, and initialize from it a set $S = \{I\}$. Assume that the shape parameters corresponding to this image are spatially uniform (scene in infinity).
2. Compute motion parameters (translation components and optionally rotation components) by aligning a new frame to the existing set S.
3. Add the registered frame to the set S.
4. Estimate the shape parameters (the slopes of EPI lines or the slope of EPI planes) for this set S.
5. Return to 2. Repeat until reaching the last frame of the sequence.

Fig. 7 demonstrates this scheme for the case of a camera translating along a straight line.

4.2 Estimating the Shape Parameters (EPI slopes)

The shape parameters are needed only for a subset of image points, as they are used to compute only a few motion parameters. The process can be formulated in the following way: Let k be the index of the frame for which we estimate the shape and let $T_{n,k} = (u_n - u_k, v_n - v_k)^t$ be the translation of the optical center of the camera between the n^{th} and the k^{th} frames.

Following [10], The shape parameter $d = d(x, y, k)$ in the image point (x, y) minimizes the error function:

$$Err(d) = \sum_{n \neq k} w_n^d \cdot \sum_{x,y \in W} (d \cdot \bigtriangledown I^t \cdot T_{n,k} + I_n - I_k)^2, \quad (6)$$

Where $\bigtriangledown I$ is the gradient of the image I_k in the point (x, y), and W is a small window around (x, y). (A 5x5 window was used). The minimum of this quadratic equation is obtained by:

$$d = -\frac{\sum_{n \neq k} w_n^d \cdot \sum_{x,y} \bigtriangledown I^t \cdot T_{n,k} \cdot (I_n(x, y) - I_k(x, y))}{\sum_{n \neq k} w_n^d \cdot \sum_{x,y} (\bigtriangledown I^t \cdot T)^2} \quad (7)$$

The weights w_n^d determine the influence of each frame on the shape estimation. Most of the weights are set to zero, except for frames which are close in time or in space (currently we use the five closest frames).

For each window in I_k, the computation described above is repeated iteratively until convergence, where in each iteration, the relevant regions in all the frames $\{I_n\}$ with $w_n^d \neq 0$ are warped back towards I_k according to $T_{n,k}$ and the current estimate of d.

As we do not need to estimate the shape parameters for every pixel, only the best points are used:

1. We do not use points with a small gradient in the direction of motion. The threshold is selected according to the desired number of points to use.
2. We do not use points for which the iterative shape computation algorithm fails to converge.

4.3 Predictive Alignment with Parallax

The alignment concept is demonstrated in Fig. 7. Given the shape parameters (EPI slopes) computed from the previously aligned frames, the motion parameters should be those that best predict the next frame. This is computed using a slight modification of the Lucas-Kanade direct $2D$ alignment as described in [2].

Assume that all the images $I_0 \ldots I_{k-1}$ have already been aligned, and let the k^{th} frame be the new video frame. We also know of the shape parameters $d(x, y, n)$ for $n < k$. To compute the motion parameters of the new frame, we minimize the following prediction error function: (Sometimes the term I_t is used to denote the difference between images).

$$Err(p, q) = \sum_{n \neq k} w_n^a \cdot \sum_{x,y} (p\frac{\partial I_n}{\partial x} + q\frac{\partial I_n}{\partial y} + I_n - I_k)^2, \quad (8)$$

where the displacement p, q of each point is given by:

$$\begin{aligned} p(x, y, n) &= (u_n - u_k) \cdot d(x, y, n) \\ q(x, y, n) &= (v_n - v_k) \cdot d(x, y, n). \end{aligned} \quad (9)$$

Note the use of the derivatives $\frac{\partial I_n}{\partial x}$ and $\frac{\partial I_n}{\partial y}$ which are estimated from I_n rather then from I_k, since we haven't computed $d(x, y, k)$ yet, and therefore we must align frame I_k to the rest of the images.

The coefficients w_n^a are also used to weight the importance of each frame in the alignment. For example, frames which are far off, or contain fewer information should receive smaller weights. For each image whose location u_n, v_n is unknown we set $w_n^a = 0$.

Currently we use about three preceding frames to predict the next frame. When the camera is translating on a plane we use several additional frames which are not neighbors in time but whose optical centers are close. In this way we reduce the drift in the motion computations.

4.3.1 Handling rotations

When the camera can also rotate, image displacements are a combination of the translational component, which is depth dependent, and the rotational component which is depth independent. Assuming small camera rotations and using the approximation $cos(\alpha) \approx 1$ and $sin(\alpha) \approx \alpha$ the following motion model is obtained:

$$\begin{aligned} p(x,y,n) &= (u_n - u_k) \cdot d(x,y,n) + a - \alpha \cdot y \\ q(x,y,n) &= (v_n - v_k) \cdot d(x,y,n) + b + \alpha \cdot x. \end{aligned} \quad (10)$$

a and b denote the small pan and tilt which induce an approximately uniform displacement in the image. α denotes small camera rotation about the z axis. For larger rotations, or when the focal length is small, full rectification can be used.

Using Eq. 10 with the error function in Eq. 8, and setting to zero the derivative with respect to the motion parameters (camera shift u, v and rotational components α, a, b), gives a set of five linear equations with five unknowns.

If the camera is restricted to translate along a straight line (without the loss of generality this line is horizontal), then $v_n = v_k = 0$, and we are left with fewer unknowns - one unknown for translation only, and four unknowns for translation plus rotation.

4.4 Examples: Predictive Alignment with Parallax

The example in Fig. 8 is from a video that was taken from a moving car having substantial motion parallax. The differences between the mosaic images obtained by $2D$ image alignment and the mosaic images obtained by predictive alignment is evident.

5 Concluding Remarks

An approach for video registration of dynamic images has been presented. The image dynamics can be a result of dynamics in the scene, or a result of motion parallax. The frames in such video sequences can be aligned by predicting the next frame from the preceding frames.

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

The predictive alignment was also shown to be applicable to motion parallax, when a camera is moving in a static scene. The stronger assumptions that can me made for motion parallax result in more accurate alignment.

A possible future challenge can be the development of predictive alignment when motion parallax and scene dynamic are combined. 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] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision (IJCV'87)*, 1(1):7–56, 1987.

[4] A. Criminisi, S. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *CVIU*, 97(1):51–85, January 2005.

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

[6] G. Doretto, A. Chiuso, S. Soatto, and Y. Wu. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, February 2003.

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

[8] 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.

[9] 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.

[10] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. *PAMI*, 24(11):1528–1534, November 2002.

[11] 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.

[12] M. Levoy and P. Hanrahan. Light field rendering. *SIGGRAPH*, 30:31–42, 1996.

[13] 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.

(a)


(b)


Figure 8. Mosaicing from a translating camera with motion parallax.
(a) Using regular $2D$ parametric image alignment. Distortions occur when image motion alternates between far and near objects.
(b) Using predictive alignment all cars are properly scaled.

[14] A. Rav-Acha and S. Peleg. A unified approach for motion analysis and view synthesis. In *Second IEEE International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Thessaloniki, Greece, September 2004.

[15] 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.