# REGION-BASED MOTION-COMPENSATED FRAME RATE UP-CONVERSION BY HOMOGRAPHY PARAMETER INTERPOLATION

*Engin Türetken*[1], *and A. Aydın Alatan*[2]

[1] Ecole Polytechnique Fédérale de Lausanne, EPFL, CH-1015 Lausanne, Switzerland.
[2] Department of Electrical and Electronics Engineering, METU, 06531 Balgat, Ankara, Turkey.
e-mail: engin.turetken@epfl.ch, alatan@eee.metu.edu.tr

## ABSTRACT

A new region-based frame interpolation algorithm is proposed based on the segmented motion layers with planar perspective motion models. It is shown that performing the motion model interpolation in the homography parameter space is equivalent to interpolating the parameters of the real camera motion, which requires decomposition of the homography matrix, under several practically reasonable assumptions. Based on this reasoning, backward and forward motion models from the interpolation frame(s) to the original frames of the sequence are estimated for each motion layer. Layer support maps at the point(s) of interpolation in time are generated by using these interpolated motion models and the layer maps of the neighboring original frames. Finally, pixel intensities of the interpolation frame are determined by a series of conditions on the layer occlusion relations and the intensities at the transformed locations of the original frames. Experimental results show that the proposed algorithm achieves visually pleasing results without blur or halo effects on dynamic scenes with complex motion.

***Index Terms***— Frame rate up-conversion, frame interpolation, motion segmentation, motion model interpolation

## 1. INTRODUCTION

Frame rate up-conversion (FRUC), or scan/field rate up-conversion, is a technique of increasing the frame rate of a video signal by inserting interpolated frame(s) in-between the successive frames of the original signal. As a particular case, FRUC is often utilized to up-convert frame rate of a typical broadcast data for hold-type displays, such as liquid crystal displays (LCD) and electro-luminescence displays (ELD). Hold type display devices are known to have the characteristic of holding the displayed frame of a video until the next frame is written, which causes the moving objects to appear blurred, since the eyes predict the next frame in a time interval shorter than the inter-frame display period of an original sequence and try to track the moving object. Therefore, for a better perception quality, additional frames are required to be interpolated to increase the frame rate of the original signal. As another particular case, FRUC is utilized for conversion between widely used formats (e.g. from PAL or SECAM to NTSC).

The two most commonly used FRUC techniques are frame duplication/repetition and motion compensation based methods. Frame duplication/repetition generally yield inferior results, especially on complex (high depth variation and clutter) or dynamic (moving objects) scenes. Motion compensation based approaches yield relatively more elegant results on these scenes with the cost of a higher computational complexity and provided that accompanying motion estimation and interpolation schemes provide satisfactorily accurate estimates.

Most of the motion compensation based techniques rely on the idea of linearly interpolating motion vectors between the original successive frames to estimate the motion vectors between the frame to be interpolated (i.e. interpolation frame) and the original frames [1]-[5]. Alternatively, a higher order function can be used, such as a polynomial, as proposed in [6]. On the other hand, the interpolation problem becomes more complicated for parametric motion models, since a suitable parameter space for interpolation may need to be determined first.

The selection of pixel grouping scheme in the interpolation frame is another important consideration. In many methods, a regular grid of fixed-shape blocks is formed in the frame to be interpolated and the frame is interpolated at these block locations using the estimated motion parameters [7]-[9]. Although suffering from blocking artifacts, this approach has an advantage that every pixel is assigned to a block so that pixel association ambiguity in the middle frame is efficiently resolved. In a similar spirit, several region-based frame interpolation techniques utilize a segmentation step and obtain arbitrarily shaped region maps for the interpolation frame [10],[11].

In this paper, we follow a similar approach and propose a new region-based method for FRUC based on segmented motion layers with planar perspective motion models. Interpolation of the layer motion models is performed in the model parameter space without decomposing them into 3D structure and motion components. Interpolated motion models are then used to warp the layer support maps at the point of interpolation in time. Finally, the interpolation frame is generated from the two neighboring original frames of the sequence by taken into account layer occlusion relations and local intensity similarities.

## 2. ESTABLISHING MOTION LAYER CORRESPONDANCES

The first step of the proposed method is to extract a set of segmentation maps for consecutive frames $F^{t-1}$ and $F^{t}$ of the sequence, between which a number of frames $\{F^{t-\Delta t_1}$, $F^{t-\Delta t_2}$, $\ldots, F^{t-\Delta t_n}\}$ $(0 < \Delta t_i < 1)$ will be interpolated. A variant of the motion segmentation algorithm proposed by Bleyer *et al*. [12] is utilized for extracting motion layers for both $F^{t-1}$ and $F^{t}$, where layer motion is modeled by planar perspective mapping (homography). An additional cost term for temporal coherence of motion layers is incorporated in the original cost function so as to make the layer extracting process robust to abrupt changes in layer appearances along the temporal axis.

As required by the following algorithm stages, the correspondences between the estimated motion layers at time instants *t-1* and *t* are established by mapping the layers and their pair-wise similarity scores to a bipartite graph. In a bipartite graph $G = (U, V, E)$ with two disjoint sets of vertices $U$ and $V$, and a set of links $E$, every link connects a vertex in $U$ and a vertex in $V$. In the proposed approach, the set of layers corresponding to $F^{t-1}$ and $F^t$ are mapped to the disjoint sets $U$ and $V$ with vertices representing the layers and weighted links representing the pair wise layer similarities. The normalized link weight between a layer $L_i^{t-1}$ of $F^{t-1}$, and a backward motion-compensated layer ${}^wL_j^t$ of $F^t$ is defined as

$$E(L_i^{t-1}, {}^wL_j^t) = \frac{\left|L_i^{t-1} \cap {}^wL_j^t\right|}{\min(\left|L_i^{t-1}\right|, \left|{}^wL_j^t\right|)} \quad (1)$$

where $L_i^{t-1} \cap {}^wL_j^t$ denotes the overlapping region of the layers and $|\,.\,|$ denotes the area of a region. The initial graph constructed with the similarity weights obtained from (1) has a link between every layer in $U$ and every layer in $V$. This redundancy is eliminated by deleting the links having weights below a prescribed threshold $T_E$. However, the links, whose source or target vertices has only one link left, are retained to ensure that every vertex is connected to the graph.

## 3. INTERPOLATING THE MOTION MODELS

Model parameter interpolation refers to estimating forward and backward motion models for a set of layers corresponding to the interpolation frames by using the backward models from $F^t$ to $F^{t-1}$. Suppose that only a single frame $F^{t-\Delta t}$, corresponding to time instant $t - \Delta t$ ($0 < \Delta t < 1$), is to be interpolated between the original frames. Given a set of backward layer motion models $\{P_1^t, P_2^t, ..., P_n^t\}$, representing the motion from $F^t$ to $F^{t-1}$, model parameter interpolation problem can be defined as estimating the parameters of forward models $\{P_{1,f}^{t-\Delta t}, P_{2,f}^{t-\Delta t}, ..., P_{n,f}^{t-\Delta t}\}$ from $F^{t-\Delta t}$ to $F^t$ and backward models $\{P_{1,b}^{t-\Delta t}, P_{2,b}^{t-\Delta t}, ..., P_{n,b}^{t-\Delta t}\}$ from $F^{t-\Delta t}$ to $F^{t-1}$. The problem is depicted in Figure 1 for a single layer.

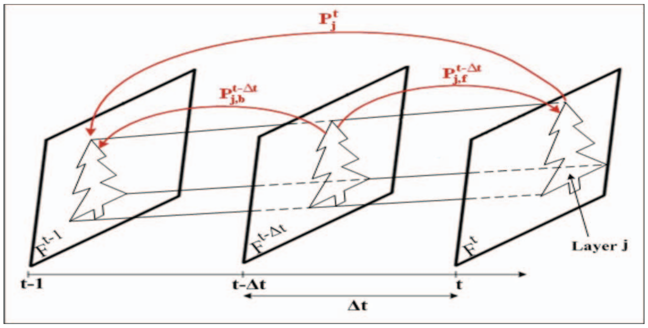


**Figure 1:** Motion model parameter interpolation problem.

Model parameter interpolation for the homography case involves in decomposition of the homography matrix into structure and motion elements of the captured scene. Suppose that the observed world points corresponding to an estimated motion layer lie on the homogeneous plane $\pi = (\mathbf{n}^T, d)$, where $\mathbf{n}$ is the unit plane normal and $d$ is the orthogonal distance of the plane from the camera center at time $t$. The projective homography matrix $P^t$ of backward motion field can then be expressed as follows [17]:

$$P^t = K^{t-1}\,(R - \mathbf{t}\,\mathbf{n}^T)\,(K^t)^{-1} \quad (2)$$

where $R$ is the rotation matrix, $\mathbf{t}$ is the translational vector of the relative camera motion normalized with the distance $d$, $K^{t-1}$ and $K^t$ are the camera calibration matrices corresponding to time instants $t-1$ and $t$, respectively.

Although several approaches exist to estimate $R$, $\mathbf{t}$, and $\mathbf{n}$ from a given homography matrix [13]–[16], the decomposition induces some computational complexity and requires the internal calibration matrices to be available. It will be shown in the following paragraphs that the decomposition can be avoided by a series of reasonable assumptions. For the time being, let the decomposed parameters be $\hat{R}$, $\hat{\mathbf{t}}$, and $\hat{\mathbf{n}}$ for the rotation matrix, the translation vector and the surface normal, respectively.

The rotation matrix can be expressed in the angle-axis representation with a rotation angle $\theta$ about a unit axis vector $\mathbf{a}$ [17]:

$$\hat{R} = I + \sin(\theta)[\mathbf{a}]_x + (1 - \cos(\theta))[\mathbf{a}]_x^2 \quad (3)$$

where $[\mathbf{a}]_x$ is the skew-symmetric matrix of $\mathbf{a}$. Under small angle approximation of rotation, the equation simplifies to

$$\hat{R} = I + \theta[\mathbf{a}]_x \quad (4)$$

Assuming that the distance between the camera centers in the direction of the plane normal $\mathbf{n}$ is much smaller than the distance $d$, the decomposed motion parameters can be interpolated at a time instant $t - \Delta t$ by a linear model:

$$\begin{aligned} \theta_b &= \theta\,\Delta t \\ \mathbf{t}_b &= \mathbf{t}\,\Delta t \end{aligned} \quad (5)$$

For the sake of simplicity, it is further assumed that the amount of change in focal length between $t-1$ and $t$ is negligible (i.e. $K^{t-1} \approx K^t$). The backward projective homography matrix $P_b^{t-\Delta t}$ of the interpolation frame can then be reconstructed as

$$P_b^{t-\Delta t} = K^t\,(I + \theta_b\,[\mathbf{a}]_x \text{ - } \mathbf{t}_b\,\mathbf{n}^T)\,(K^t)^{-1} = (1 - \Delta t)\,I + \Delta t\,P_b^t \quad (6)$$

which reveals that there is no need to decompose the homography matrix $P^t$ and estimate the camera calibration under the mentioned assumptions. Finally, the forward homography matrix $P_f^{t-\Delta t}$ can simply be computed from the available models as follows:

$$P_f^{t-\Delta t} = (P^t)^{-1} P_b^{t-\Delta t} \quad (7)$$

## 4. INTERPOLATING THE LAYER MAPS

The interpolation of the motion models is followed by estimation of the corresponding layer support maps at time instants $t-1$ and $t-\Delta t$. This is achieved essentially by backward warping the extracted layers (both support maps and intensities) of $F^t$ to the two previous time instants, and updating the overlapping and uncovered regions so as to ensure a single layer assignment for each pixel of $F^{t-\Delta t}$ and $F^t$. Figure 2 provides an overview of the layer map interpolation process.

Depth ordering relations of the overlapping layers is extracted by computing a visual similarity measure between each warped layer and the original image $F^{t-1}$ over the region of overlap. Each pixel of an overlapping region votes for the layer that gives the minimum sum of absolute intensity difference. Visual similarity of a warped layer is then modeled as the number of pixels that vote for the layer. The warped layers at $t-1$ and $t-\Delta t$ are updated by assigning the overlapping regions only to the top layers, which yield the maximum similarity score.

Pixels in the uncovered regions are assigned separately by using the extracted layers of previous time ($t-1$). For an uncovered pixel, the set of candidate current time ($t$) layer labels corresponding to the previous time layer label at that pixel are determined by using the estimated layer correspondences. Finally, the uncovered pixel is assigned to the spatially closest candidate to avoid the creation of disconnected support maps and to ensure a maximum level of compactness.

A similar strategy is followed for uncovered pixels of the interpolation frame. For each pixel of an uncovered region, the candidate layer set is initialized as the neighboring layers of the region. Each pixel of an uncovered region, is then warped to the previous time using the interpolated backward motion models $\{P_{k,b}^{t-\Delta t}, P_{l,b}^{t-\Delta t}, \ldots, P_{m,b}^{t-\Delta t}\}$ of the candidate layers. If the layer label at the warped location is different for a candidate layer, it is removed from the set. Finally, as before, the spatially closest layer is selected among the candidates for each uncovered pixel.

## 5. GENERATING THE INTERPOLATION FRAME

In the proposed approach, the intensity of each pixel in the interpolation frame is modeled as a function of the intensities at the corresponding locations in the original frames $F^{t-1}$ and $F^t$, which are computed by using the previously estimated backward and forward layer motion models. In the current implementation, bicubic interpolation is used to obtain the intensities at the sub-pixel locations.

Let the intensity vector (and layer label) at an integer pixel location $\mathbf{x}^{t-\Delta t}$ of the interpolation frame $F^{t-\Delta t}$ be denoted as $\mathbf{I}^{t-\Delta t}$ ($L^{t-\Delta t}$), and the intensity vectors (layer labels) at the corresponding sub-pixel locations $\mathbf{x}^{t-1}$ in $F^{t-1}$ and $\mathbf{x}^t$ in $F^t$ are computed to be $\mathbf{I}^{t-1}$ ($L^{t-1}$) and $\mathbf{I}^t$ ($L^t$), respectively. Then, the interpolated pixel intensity vector is expressed by the following simple weighted linear averaging scheme:

$$\mathbf{I}^{t-\Delta t} = \begin{cases} \mathbf{I}^{t-1} & : L^{t-\Delta t} = L^{t-1},\ L^{t-\Delta t} \neq L^t \\ \mathbf{I}^t & : L^{t-\Delta t} \neq L^{t-1},\ L^{t-\Delta t} = L^t \\ (1-\Delta t)\mathbf{I}^{t-1} + \Delta t \mathbf{I}^t & : \begin{array}{l} L^{t-\Delta t} = L^{t-1},\ L^{t-\Delta t} = L^t, \\ \left\| \mathbf{I}^{t-1} - \mathbf{I}^t \right\| < T_I \end{array} \end{cases} \tag{8}$$

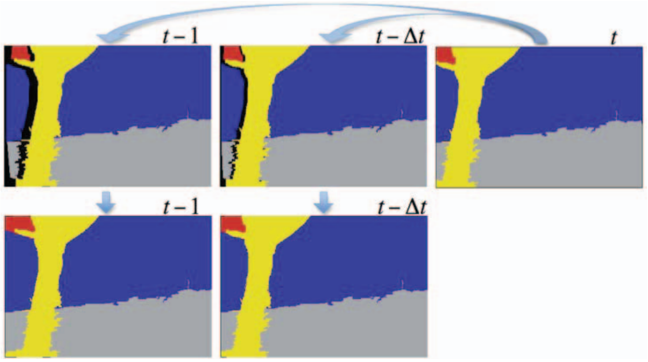


**Figure 2:** Overview of the layer map interpolation process. Top row: warping layer maps at time instant $t$ (right) to the two previous time instances $t-1$ (left) and $t-\Delta t$ (middle). Bottom row: updating the layer assignments on overlapping and uncovered regions of the warped layers at $t-1$ (left) and $t-\Delta t$ (right).

where $T_I$ is a prescribed intensity difference threshold for detecting the disagreement between intensities of the corresponding original frame locations, and hence, avoiding the formation of blur and "halo" (i.e. ghosting artifacts around motion boundaries) effects. The above equation states that if the layer label at a transformed location $\mathbf{x}^{t-1}$ or $\mathbf{x}^t$ is different than the layer label at the source location $\mathbf{x}^{t-\Delta t}$, then the intensity at the transformed location should not be taken into account, since the layer is occluded at that point. For all the remaining cases that are not covered by (8), only the intensity that is closer to the average intensity $\mathbf{N}^{t-\Delta t}$, in the neighborhood of the interpolation frame location $\mathbf{x}^{t-\Delta t}$, is used as follows:

$$\mathbf{I}^{t-\Delta t} = \begin{cases} \mathbf{I}^{t-1} & : \left\| \mathbf{N}^{t-\Delta t} - \mathbf{I}^{t-1} \right\| \leq \left\| \mathbf{N}^{t-\Delta t} - \mathbf{I}^t \right\| \\ \mathbf{I}^t & : \left\| \mathbf{N}^{t-\Delta t} - \mathbf{I}^{t-1} \right\| > \left\| \mathbf{N}^{t-\Delta t} - \mathbf{I}^t \right\| \end{cases} \tag{9}$$

## 6. EXPERIMENTAL RESULTS

The proposed algorithm is tested on several well-known video sequences. In order to assess the performance of the proposed method, only odd frames of the inputted sequences are processed and a single frame corresponding to $\Delta t = 0.5$ is interpolated between each pair of odd frames. The interpolated frames are then compared with the even frames of the original sequence both objectively and subjectively.

Some of the interpolation results are illustrated in Figure 3 and Figure 4 for *Flower Garden* and *Mobile & Calendar* sequences (352x240), respectively. The algorithm achieves visually pleasing results without apparent blur or halo effects and with sharpness at object boundaries preserved. For objective evaluation, peak signal-to-noise ratio (PSNR) between the even frames of the original sequence and the interpolated frames are computed. Figure 5 provides a comparison between the proposed method and simple frame averaging for the first 150 frames of *Flower Garden* and *Mobile & Calendar* sequences. The plots show a significant improvement in PSNR (up to 8dB for *Flower Garden*) as well as an enhanced robustness to changes in motion complexity.

## 7. CONCLUSIONS

FRUC is an important problem for consumer electronics, especially for improving the performance of hold-type displays. Increasing frame-rates (especially more than a factor of two) requires modeling and utilization of motion field more precisely. Based on the experimental results, the proposed FRUC algorithm, which exploits affine motion models in arbitrarily shaped regions, yields quite promising results.

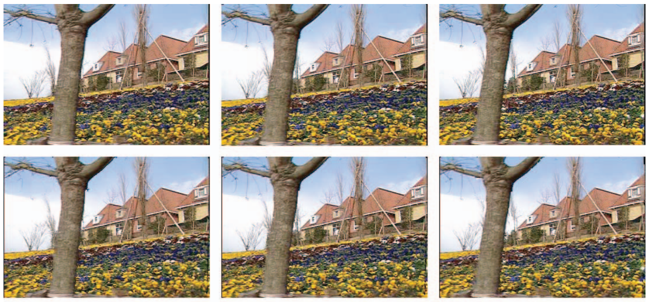


**Figure 3:** Interpolation results for *Flower Garden* sequence. Top row, left to right: frames 324, 326 and 328 of the original sequence. Bottom row, left to right: corresponding interpolated frames.

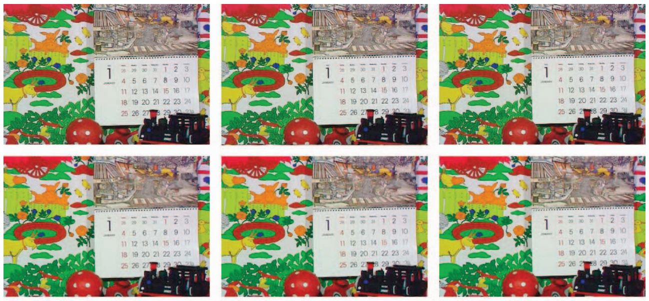
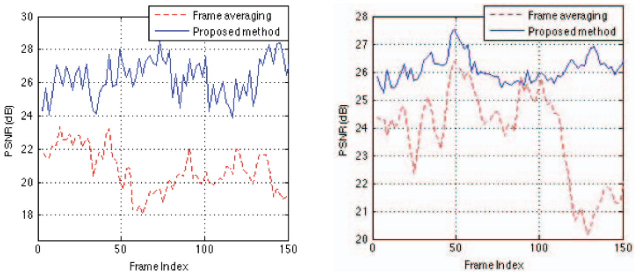


**Figure 4:** Interpolation results for *Mobile & Calendar* sequence. Top row, left to right: frames 38, 40 and 42 of the original sequence. Bottom row, left to right: corresponding interpolated frames.



(a) (b)

**Figure 5:** PSNR of the proposed method compared to frame averaging for the first 150 frames of (a) *Flower Garden* and (b) *Mobile & Calendar* sequences.

## 8. REFERENCES

[1] L.U. Tiehan, "Motion Compensated Frame Rate Conversion with Protection Against Compensation Artifacts," *WIPO*, Patent No. 2007123759, Nov. 1, 2007.

[2] K. Ohwaki, Y. Takeyama, G. Itoh, and N. Mishima, "Apparatus, Method, and Computer Program Product for Detecting Motion Vector and for Creating Interpolation Frame," *US Patent Office (US PTO)*, Patent No. 2008069221, Mar. 20, 2008.

[3] K. Sato, M. Yamasaki, K. Hirayama, H. Yoshimura, Y. Hamakawa, K. Douniwa, and Y. Ogawa, "Interpolation Frame Generating Method and Interpolation Frame Generating Apparatus," *US PTO*, Patent No. 2008031338, Feb. 7, 2008.

[4] H.F. Chen, S.S. Kim, and J.H. Sung, "Frame Interpolator, Frame Interpolation Method and Motion Reliability Evaluator," *US PTO*, Patent No. 2007140346, Jun. 21, 2007.

[5] K. Ohwaki, G. Itoh, and N. Mishima, "Method, Apparatus and Computer Program Product for Generating Interpolation Frame," *US PTO*, Patent No. 2006222077, Oct. 5, 2006.

[6] K. Bugwadia, E.D. Petajan, and N.N Puri, "Motion Compensation Image Interpolation – Frame Rate Conversion for HDTV," *US PTO*, Patent No. 6229570, May 8, 2001.

[7] B.D. Choi, J.W. Han, C.S. Kim, and S.J. Ko, "Motion Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Trans, Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407-416, Apr. 2007.

[8] N. Mishima, and G. Itoh, "Novel Frame Interpolation Method For Hold-Type Displays," *IEEE International Conf. on Image Processing (ICIP)*, Singapore, TA-P3.9, pp. 1473-1476, 2004.

[9] B.T. Choi, S.H. Lee, and S.J. Ko, "New Frame Rate Up-conversion Using Bi-directional Motion Estimation," *IEEE Trans. on Consumer Electron.*, vol. 46, (3), pp. 603-609, Aug. 2000.

[10] J. Benois-Pineau, and H. Nicolas, "A New Method for Region-based Depth Ordering in a Video Sequence: Application to Frame Interpolation," *Journal of Visual Communication and Image Representation*, vol. 13, no. 3, pp. 363–385, 2002.

[11] J. Wang, "Video temporal reconstruction and frame rate conversion," *ETD Collection for Wayne State University*, Paper AAI3243070, Jan. 2006.

[12] M. Bleyer, M. Gelautz, and C. Rhemann, "Region-based Optical Flow Estimation with Treatment of Occlusions", *Joint Hungarian-Austrian Conference on Image Processing and Pattern Recognition (HACIPPR)*, pp. 235 – 242, 2005.

[13] R. Y. Tsai, and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 6, pp. 1147-1152, 1981.

[14] O. Faugeras, and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3): pp. 485–508, 1988.

[15] Z. Zhang, and A.R. Hanson, "3D Reconstruction based on homography mapping", *Proc. ARPA96*, pp. 1007-1012, 1996.

[16] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control", INRIA, ISSN 0249-6399 ISRN INRIA/RR-6303-FR+ENG, 2007.

[17] R. I. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge Univ. Press, Second Edition, 2004.