Interpolation of Cinematic Sequences

J. Ribas-Corbera and J. Sklansky

Department of Electrical and Computer Engineering
University of California, Irvine
Irvine, CA 92717
U.S.A.

Abstract

We present a new algorithm for interframe interpolation of cinematic sequences. We demonstrate its applicability to video data compression of pedestrian traffic and data compression for video conferencing. In both of these applications it is assumed that the background is nearly stationary and that there are no inter-object occlusions. The interpolation algorithm makes use of estimates of optical flow to compensate for the motion of objects between two frames. We describe three major problems associated with motion compensated cinematic interpolation: interframe occlusion, interframe zooming and figure-ground ambiguity. Our algorithm suppresses artifacts caused by all three of these problems.

Index terms: computer vision, interframe interpolation, video compression.

1 INTRODUCTION

Interframe interpolation is the construction and insertion of additional frames in cinematic sequences. In video communications, skipping frames at the transmitter and interpolating them at the receiver is a video compression technique that reduces significantly the bit rate to transmit the video signal. Interframe interpolation can also be combined with other data compression techniques to further reduce the bit rate. For instance, subsampling a video sequence by a factor of 4 (i.e., out of every 4 frames only 1 is transmitted) and further compressing the transmitted frames by a factor of 25 using a standard technique such as JPEG [1], would achieve an overall compression ratio of 100:1. Interframe interpolation has potential additional applications in slow-motion cinematography, medical imaging, and computer-aided production of animated movies.

Interframe interpolation of good quality may be achievable by modeling the motion of objects in the scene [15],[16]. Our interpolation algorithm makes use of motion estimates and is based on the assumption that the scene consists of a few moving objects and a nearly stationary background.

Our interframe interpolation algorithm interpolates an arbitrary number of frames between two successive frames of a cinematic sequence. The algorithm consists of two stages: a) ray detection, and b) ray interpolation.

- In ray detection a forward optical flow, computed with the two frames in the natural order, and a backward optical flow, computed with the frames in the reverse order, are matched in order to associate pixels in one frame that correspond to pixels in the other one.
- In ray interpolation we construct the interpolated frames by estimating the gray levels between corresponding pixels and by filling gaps and resolving ambiguities that appear in the interpolation planes.

We present interpolation results on a video sequence of pedestrian traffic and on a video sequence of head-and-shoulders view of a salesman. We compare the resulting sequences to the sequences produced by linear interpolation and by a recent interframe interpolation scheme of Cafforio *et al.* [16]. We evaluated the distortion quantitatively by computing histograms of the relative absolute error. Our results are quantitatively and qualitatively superior to those produced by these other two algorithms. In particular, our algorithm filled gaps and resolved ambiguities caused by interframe occlusion, interframe zooming, and figure-ground ambiguity.

2 PRIOR WORK

Interframe interpolation is an attractive technique for video compression. It has been proven that sim-

0-8186-2840-5/92 $03.00 © 1992 IEEE

ple interpolation schemes such as zero-order interpolation [8] or linear interpolation [9] produce unsatisfactory results whenever motion is present in the scene. Therefore, it is necessary to compensate for the motion effects in the interpolation process. Encouraging results have been achieved in video conferencing and video telephony. However, cinematic sequences with large interframe displacements, occlusions, and nonrigid moving objects are still difficult to interpolate.

Several motion compensated image interpolation schemes have been proposed in the literature, [10], [11], [12], [13], [14], [15], [16]. Typically such an interpolation scheme consists of two steps: a) estimation of a map of all the displacements in the scene, and b) interpolation of frames using the map of displacements. Experience with these schemes indicates that there are two main problems associated with interframe interpolation:

- Displacement estimation algorithms must be improved to give more accurate estimates for single picture elements.
- Phenomena such as interframe occlusion, interframe zooming and figure-ground ambiguity produce in the interpolation plane gaps that must be filled and ambiguities that must be resolved to avoid artifacts in the interpolated images.

Most interpolation algorithms do not fill those gaps and resolve those ambiguities adequately. Some interpolation schemes propose segmentation techniques in order to overcome the interframe occlusion of background [11], [15]. These techniques are complex and have several drawbacks. In Section 3.2.2 we describe the drawbacks of the most recent of these techniques, and we propose a new method to deal with this problem. In addition we describe our approach to overcoming interframe zooming and figure-ground ambiguity.

3 THE INTERFRAME INTERPOLATION SCHEME

3.1 Assumptions and Definitions

We refer to the given frames for interpolation of a cinematic sequence as *reference frames*. Our interpolation scheme reconstructs N missing frames between two consecutive reference frames from estimated trajectories and intensity variations of points in brightness patterns in the image-time domain. We refer to these estimated trajectories as *pixel trajectories*. If two succesive reference frames are separated by 1/6th ot a second or less, it is reasonable to assume the following:

- The pixel trajectories may be estimated adequately from the estimated optical flow in each of the reference frames [16].
- Between every pair of consecutive reference frames every pixel trajectory is approximately linear, and the velocity of the pixel along this segment of the pixel trajectory is approximately constant [10].

The straight line in the image-time domain that joins the center of the a pixel A in a reference frame to the center of its corresponding pixel B in the next preceding or succeeding reference frame is called a *correspondence ray*, or *ray* for short (Figure 1).

3.2 The Interpolation Algorithm

Our interpolation algorithm consists of two stages:

- Stage 1 (ray detection): Find the rays that join corresponding pixels in every two consecutive reference frames.
- Stage 2 (ray interpolation): Reconstruct the missing frames by linearly interpolating the pixel values at the two ends of each unambiguous ray, and then resolving ambiguities and filling gaps.

3.2.1 Ray Detection

A pixel A in the first reference frame R_1 *corresponds* to a pixel B in the second reference frame R_2 if the trajectory of A in the image-time domain from R_1 to R_2 coincides with the trajectory of B from R_2 to R_1. Corresponding pixels belong to the same brightness pattern rather than to the same object. An object may be static but the light source may move, thereby producing brightness changes in the scene. On the other hand an object may be moving but there may not be brightness changes in the image, e.g. a rotating homogeneous sphere with a temporally invariant light source, [2]. A correct interpolation requires the interpolation of moving brightness patterns with respect to time. Therefore, optical flow is suitable for interpolation since it estimates the motion of brightness patterns in a scene.

We make use of the optical flow estimation algorithm proposed by Agarwal *et al.* [3] due to the following reasons:

- It is able to accomodate large interframe displacements.

- It gives good optical flow estimates, especially in the presence of motion boundaries.

In general, any optical flow algorithm that provides these two capabilities is admissible in our interpolation scheme. In adition to [3], the optical flow techniques in [4], [5], [6], [7] may also be suitable for our scheme. The more accurate the optical flow, the better our interframe interpolation algorithm performs.

Our optical flow estimation algorithm applied on two consecutive reference frames, R_1 and R_2, estimates the average velocity vector for the pixels of R_1 in their trajectories from R_1 to R_2. We refer to this optical flow as the *forward flow*. The optical flow estimation for the pixels in the second reference frame, R_2, obtained from R_2 to R_1, is referred as *backward flow*.

If the forward flow of a pixel A in R_1 takes A to pixel B in R_2, and the backward flow of B takes B back to A these two pixels are declared *correspondent*, and a correspondence ray joining A and B is constructed. In practice this rule for forming correspondence rays is too restrictive, resulting in frequent missing of significant rays. To overcome this problem we use a simple method described by Cafforio and al. [16] which consists of two steps:

- At each pixel A in R_1 use the forward flow to identify a pixel A' in R_2.
- A small window centered on A' is searched for the pixel B where the backward flow identifies the pixel B' within R_1 that is at a minimum distance from A. If this distance is smaller than a given threshold, pixels A and B are declared correspondent (Figure 1).

The image-plane coordinates $(x_r,\ y_r)$ of a ray that joins a pixel A in R_1 to a pixel B in R_2, at each interpolated frame n follow the equations:

$$x_r(n) = \frac{(N+1-n)x_A + nx_B}{N+1}$$

$$y_r(n) = \frac{(N+1-n)y_A + ny_B}{N+1}$$

$$1 \le n \le N \qquad (1)$$

where $(x_A,\ y_A)$ are the image-plane coordinates of pixel A, $(x_B,\ y_B)$ are the image-plane coordinates of pixel B, and N are the number of frames to interpolate between R_1 and R_2.

3.2.2 Ray Interpolation

In this stage we reconstruct the N missing frames between R_1 and R_2 by analyzing the evolution along time of the detected rays. We can distinguish two steps in this procedure: a) direct interpolation and b) suppression of gaps and ambiguities.

a) Direct interpolation:

Most of the pixels in the interpolated frames can be reconstructed directly by linearly interpolating the pixel values at the two ends of a ray. The intensity associated with a ray r that joins a pixel A at R_1 to a pixel B at R_2 at each interpolated frame n is found by:

$$I_r(x_r(n),\ y_r(n)) =$$

$$Nint[\frac{(N+1-n)I_{R_1}(x_A,\ y_A) + nI_{R_2}(x_B,\ y_B)}{N+1}]$$

$$1 \le n \le N \qquad (2)$$

where $I_{R_1}(x_A,\ y_A)$ is the intensity of pixel A, which is at coordinates $(x_A,\ y_A)$, $I_{R_2}(x_B,\ y_B)$ is the intensity of pixel B at $(x_B,\ y_B)$, $Nint[x]$ denotes the nearest integer to x, and $(x_r(n),\ y_r(n))$ are the image-time domain coordinates of r found in Equations (1).

Since a ray r joins two pixels at their central points, we imagine that r carries a square pixel of L^2 area along its trajectory and that the intensity of this pixel is $I_r(x_r(n),\ y_r(n))$, which is defined in Equation (2).

A pixel in the interpolation plane may be crossed nearby by several rays (Figure 2). The pixels carried by the rays contribute to the intensity value of the pixel at (x, y) in a interpolated frame n whenever their area overlap the pixel area with center at (x, y) (Figure 2). That means that a ray r_i with coordinates $(x_{r_i}(n),\ y_{r_i}(n))$ contributes to interpolate the intensity value of the pixel (x, y) at the interpolated frame n, only if:

$$| x - x_{r_i}(n) | < L$$

$$| y - y_{r_i}(n) | < L \qquad (3)$$

Let $\{r_i\}$ be the set of rays that satisfy both inequalities. If $\{r_i\}$ are not parallel there is an ambiguity, which is solved in step b). Otherwise, the area contribution of the pixel carried by a ray r_i to a pixel (x, y) at n is:

$$A_{r_i} = (L - | x - x_{r_i}(n) |)(L - | y - y_{r_i}(n) |) \qquad (4)$$

The intensity of the pixel (x, y) at the *nth* interpolated frame, $I(x, y, n)$, can be determined by a weighted average of the rays' intensities $\{I_{r_i}\}$, where each weight is the area contribution of each ray of the set. The final intensity value for a directly interpolated pixel is computed by the following equation:

$$I(x, y, n) = \frac{\sum_i I_{r_i} A_{r_i}}{A_{r_i}} \quad (5)$$

b) Suppression of gaps and ambiguities:

We next show that even in the ideal case of having detected exact correspondence rays there appear ambiguities and gaps in the interpolation plane. Most designers supressed the ambiguities by averaging and the gaps and by spatial or zero order interpolation [10], [12], [14], [16]. These solutions degrade the reconstructed frames when the interframe displacement is large, and the moving objects are nonrigid. We suppress these ambiguities and gaps by analyzing their causes. There are three main causes of these gaps and ambiguities: interframe occlusion, interframe zooming and figure-ground ambiguity. For simplicity, we limit the assumed ambiguities and gaps to scenes where the background is nearly stationary and there are no interobject occlusions.

- **Interframe occlusion:**

Pixels that do not have an associated correspondence ray (i.e., they do not have a corresponding pixel) are often occluded in the other reference frame. Background occlusion, or covered-uncovered background, is the case of interframe occlusion that has been more examined in the literature [11], [15]. If an object moves from the first reference frame to the second one, some background around the object will get occluded in the next reference frame, producing gaps in the interpolated frames. The same effect occurs backwards from the second reference frame to the first. Observe this effect in the lateral view of a moving square at Figure 3.

Two earlier techniques that compensate for this effect propose a pre-detection and segmentation of the occluded areas [11],[15]. The more recent and more advanced technique was proposed by Thoma *et al.* [15]. This technique detects the interframe occlusion of the background based on the errors that an optical flow commits at the motion edges. The optical flow proposed in [15] is a hierarchical block matching. This technique will fail if the estimated optical flow is accurate at the motion boundaries. In addition it suffers from the complexity of a threshold and change detector computation for each reference frame, and by multiple executions of segmentation procedures in the reference frames and the interpolated frames. Our technique suppresses the gaps associated with interframe occlusion of the background while avoiding the complexities in the Thoma technique. The more accurate the optical flow, the better our technique performs.

From Figure 3 observe that if:

- a) there is a gap at a pixel P in a interpolated frame,
- b) the pixel at the same coordinate at one of the reference frames, A, does not have an associated ray, and
- c) the pixel at the same coordinate at the other reference frame, B, has an associated ray,

then there is an occlusion at pixel P. Our technique confirms the occlusion by repeating the same procedure in a small neighborhood of P, and finally estimates the intensity of P as the intensity at A.

- **Interframe zooming:**

If the image of an object expands from R_1 to R_2 the correspondence rays diverge (Figure 4), thereby producing gaps in the interpolation plane. These gaps can be suppressed by repeating the ray detection process in with R_1 and R_2 in the reverse order and filling in the missing rays. This technique is also applicable to cinematic sequences with nonstationary background and interobject occlusion. Our results show that even if there is no zooming of images, gaps caused by accidental divergence may occur, for which case this technique continues to be effective.

- **Figure-ground ambiguity:**

The most common cause of figure-ground ambiguity in an interpolated frame is the coincidence of several nonparallel rays in a pixel of an interpolation plane. We refer to this phenomenon as *ray coincidence*. For a pixel in the interpolation plane experiencing ray coincidence, Equation (5) is not applicable since the rays crossing this pixel link pixels from different scene elements in the reference frames. Consider the case of a square moving from the first to the second reference frame (lateral view is shown at Figure 5). In an interpolation plane (dashed line) there are coincidences of rays belonging to the square and rays belonging to the background. Note in Figure 5 that

the rays AA' and BB' coincide at P. We resolve this ambiguity in favor of rays that cross the pixel area in the interpolation plane with the largest slope. Then we apply formula (5) to find out the intensity value for that pixel.

In more general scenes several rays belonging to various moving objects and to a moving background may all cross a pixel area P in the interpolation plane. In those cases a sophisticated decision is required to identify the occluding objects.

4 EXPERIMENTAL RESULTS

We tested our algorithm, as well as linear interpolation and the algorithm of Cafforio *et al.* [16], on two real cinematic sequences. The first sequence is a head-and-shoulders view of a salesman ("salesman"), and the second is a scene with pedestrian traffic ("people").

All algorithms were implemented in C on a SPARC station 1+. Cafforio's interpolation scheme is also based on matching the forward and backward optical flow to find corresponding pixels between the two reference frames (Section 3.2.1). It differs from our scheme mainly in the correction of gaps and ambiguities that appear in the interpolation planes. Cafforio's interpolation technique does not resolve adequately the gaps and ambiguities caused by interframe occlusion, interframe zooming and figure-ground ambiguity. The intensity of the pixels at gaps in an interpolation plane are estimated from the pixels on that reference frame which is closer to the interpolation plane. Cafforio *et al.* do not explain how their algorithm accounts for ray coincidence ambiguities in pixels in the interpolation plane. In our implementation of Cafforio's scheme, we assumed that they resolved every ambiguity at a pixel P in the interpolation plane by averaging the gray levels of the pixels of the ambiguous rays associated with P.

We evaluated the performance of our interpolation scheme, Cafforio's scheme and linear interpolation qualitatively by examining the picture quality of the frames reconstructed in our experiments. We also evaluated our results quantitatively by computing the relative absolute error for each pixel value in the interpolated frames. For a pixel P with intensity I_P in a interpolated frame, and with intensity O_P in the original frame, the relative absolute error ϵ is:

$$\epsilon = \frac{| I_P - O_P |}{O_P} \qquad (6)$$

This quantitative analysis was performed only in regions of the images where there is significant motion.

Sequence "salesman":

The first sequence consists of 5 consecutive frames of a head-and-shoulders view of a salesman. This sequence is typical of video telephony and video conferencing. The original images were of size 360 × 288. The cinematic sequence was subsampled by a factor of 4: frames 2, 3, and 4 were interpolated from reference frames 1 and 5. All of the interframe displacements between reference frames were 7 pixels or less.

The top row of Figure 6 shows the original frames 2, 3, and 4 from left to right. The frames interpolated by our new technique are in the bottom row, under the respective original ones. Figure 7(a) shows reference frame 1 with the estimated forward optical flow, and Figure 7(b) shows reference frame 5 with the estimated backward optical flow. The amount of motion in this five-frame sequence can be appreciated in Figure 7(c), which shows the frame difference between the reference frames. Figure 7(d) shows the reconstructed frame 3. Figure 8 shows a close-up of the box and the right hand of the salesman in frames 2, 3 and 4. This region contains the fastest motion of the scene. The three reconstructed frames 2, 3, and 4 are again displayed in the bottom row. In Figures 6 and 8 the interpolated frames seem to be of acceptable quality for video conferencing, even in the neigborhood of the box, where there is fast motion.

Figure 9(a) shows a close-up of frame 3 reconstructed by our technique. Figure 9(b) shows a "direct interpolation" (see Section 3.2.2 a) in which pixels at gaps are black. In this figure the gaps around the moving box are caused by the interframe occlusion problem. There are also gaps elsewhere in the image produced by accidental ray divergences. Figure 9(c) shows the same close-up of frame 3 reconstructed by Cafforio's interpolation scheme. Observe the distortion in Figure 9(c), especially in the regions that suffer from the interframe occlusion problem. Figure 9(d) shows a linear interpolation of frame 3. Notice that our algorithm produced images that are qualitatively superior to both the algorithm of Cafforio *et al.* and linear interpolation. In Table 1, we summarize quantitative results of the performance of the three interpolation algorithms in a close-up restricted to the box and its neighborhood, shown in Figure 9. For our interpolation technique, the percents of pixels with ϵ smaller than 0.05, 0.10 and 0.20 are higher than the other two techniques, and the percent of pixels with ϵ greater than 0.5 for our technique is remarkably lower than the other two. The quantitative results corroborate the superiority of the results obtained by our interpolation technique.

Sequence "people":

The second sequence consists of 4 consecutive frames of a scene with pedestrians crossing a street. The frames were of size 256 × 240. The sequence was subsampled by a factor of 3: frames 2 and 3 were interpolated from reference frames 1 and 4. All of the displacements between reference frames were 12 pixels or less. This scene contains multiple moving objects and a stationary background. The motion is primarily translational but the objects (people) are nonrigid, and show nonrigid motion. The people in the scene are imaged at low resolution. Because of these characteristics we believe that this sequence is particularly difficult to interpolate by earlier interframe interpolation algorithms. In fact, none of the earlier reports of interpolation schemes discussed tests on such difficult cinematic sequences.

We present 4 original consecutive frames of the cinematic sequence in Figure 10, in the order (a), (c), (d), (b). Reference frames 1 and 4 are at $10(a)$ and $10(b)$ respectively. Frames 2 and 3 are interpolated by our interframe interpolation scheme. The interpolated frame 2 is shown in Figure $11(c)$ and the interpolated frame 3 is in Figure $11(d)$. Reference frames 1 and 4 are shown in Figures $11(a)$ and $11(b)$, directly above the interpolated frames. Reference frames in Figures $11(a)$ and $11(b)$ include small arrows marking the forward and backward optical flow.

In order to show the improvements of our interpolation algorithm with respect to earlier techniques we focus on the region where the motion is fastest, illustrated in Figures 12, 13 and 14. Figures 12 and 13 are close-ups of Figures 10 and 11 respectively. Figures $13(c)$ and $13(d)$ show some quantization of details in the interpolated pedestrians which do not seem visually important in the full frames. Figure $14(a)$ is the same as Figure $13(d)$. Figure $14(b)$ shows a "direct interpolation", with gaps marked black. In this figure part of the leg has disappeared, because background pixels are averaged with leg pixels in response to figure-ground ambiguities. Note the gaps around the moving body and leg. These gaps are produced by interframe occlusion of the background. In addition there are gaps on the pedestrians produced by occasional divergences of the rays. Figure $14(d)$ shows the frame produced by Cafforio's interpolation scheme. In this frame observe the distortion in the body of the moving man, especially in the leg. This distortion is mainly produced by resolving inadequately the gaps and ambiguities. Figure $14(d)$ shows the results of a linear interpolation. Not surprisingly, it is of poor quality. Thus here too our algorithm produces significantly better subjective results than the other two algorithms.

In Table 1, we summarize quantitative results of the performance of the three interpolation algorithms in a region of 10 × 15 pixels centered on the moving leg in Figure 14. Observe that in this region the quantitative results of our algorithm are much superior to those of the other two.

5 SUMMARY AND CONCLUSIONS

This paper describes a new scheme for motion compensated interframe interpolation, and demonstrates its applicability to video data compression. The scheme consists of two stages: ray detection and ray interpolation. In the first stage we estimate the pixel trajectories in the image-time domain using an optical flow technique. In the second stage we interpolate additional frames and suppress ambiguities and gaps that appear in the interpolation planes. In this paper we restrict our attention to cinematic sequences where the background is nearly stationary and there are no interobject occlusions.

We show that the ambiguities and gaps are produced by three cinematic phenomena: interframe occlusion, interframe zooming, and figure-ground ambiguity. We present an approach to suppress the ambiguities and gaps produced by these phenomena. Interframe interpolation results of a 4:1 temporal subsampling of a scene of a head-and-shoulders view of a salesman, and of a 3:1 temporal subsampling of a scene with pedestrians in a street have been obtained with good picture quality. These results suggest that a wide variety of cinematic sequences can be subsampled temporally by a factor of 3 or 4, and subsequently interpolated satisfactorily. Combining our interframe interpolation scheme with an intraframe compression technique such as JPEG [1], which can compress still images by a factor of 25, compression ratios up to 100:1 can be achieved.

Our results were qualitatively and quantitatively superior to those produced by the interpolation scheme of Cafforio *et al.*[16] and to those produced by linear interpolation. Our algorithm achieved good results even in the regions where the motion of the salesman and the motion of the pedestrians produced large interframe displacements. This suggests that our algorithm provides a significant improvement over earlier techniques.

Acknowledgements

We are grateful to Nippon Telegraph and Telephone Corporation (NTT) and the Direcció General d'Universitats - CIRIT Grant No. BE92–248 (Govt. of Catalonia) for their support of this research. We thank NTT for providing the sequence "people" for our tests. We also thank Dr. Satoshi Suzuki of NTT Visual Perception Laboratory and Prof. Glenn Healey for their useful comments. We are also grateful for many discussions with the members and visiting researchers of the Pattern Recognition and Machine Vision Project at the University of California, Irvine.

References

[1] G.K. Wallace, "The JPEG still picture compression standard", *Communications of the ACM*, vol. 34, No. 4, pp. 30-44, April 1991.

[2] B. K. P. Horn, *Robot Vision* M.I.T Press, Cambridge, Massachusetts, Figure 12-2.

[3] R. Agarwal and J. Sklansky, "Estimating optical flow from clustered trajectories in velocity-time", *1992 Int. Conf. Pattern Recognition*, The Netherlands, September 1992.

[4] A. Singh, "Incremental estimation of image-flow using a Kalman filter", *Proc. Workshop on Visual Motion, IEEE Computer Society*, (Princeton, N.J.), pp. 36-43, October 1991.

[5] J. Aisbett, "Optical-flow with an intensity weighted smoothing", *IEEE Trans. Pattern Anal. Machine Intell*, vol. 5, pp. 512-522, 1989.

[6] B. G. Schunck, "Image flow segmentation and estimation by constraint line clustering," *IEEE Trans. Pattern Anal. Machine Intell.* , vol. 11, no. 10,pp. 1010-1026, Oct. 1989.

[7] J. Hutchinson, K. Koch, and C. Mead, "Computing motion using analog and binary resistive networks", *Computer*, pp. 52-63, 1988.

[8] B.G. Haskell and R.L. Schmidt, "A low bit-rate interframe coder for videotelephone," *Bell Syst. Techn. J.*, vol. 54, no. 8, pp. 1475-1495, Oct. 1975.

[9] J. Klie, "Codierung von fernsehsignalen fur niedridge ubertragungsbitraten," *Ph.D. dissertation, Tech. University of Hannover*, Hannover, Germany, 1978.

[10] R. Lippmann, "Video transmission of aerial scenes at reduced frame rates using motion compensation," presented at the *ICC, Seattle*, Wa, June 1980.

[11] H.C. Bergmann, "Motion adaptative frame interpolation", *Proc. 1984 Int. Zurich Seminar Digital Commun.*, Zurich, 1984, pp. D2.1-D2.5

[12] A. Furukawa, T. Koga, and K. Linuma, "Motion adaptative interpolation for videconference pictures", *Proc. Int. Conf. Commun.*, Amsterdam, pp. 707-710, 1984.

[13] H.G. Musmann, P. Pirsch, and H.J. Grallert, "Advances in picture coding", *Proc. IEEE*, vol. 73, pp. 523-548, Apr. 1985.

[14] H. Yamaguchi and M. Wada, "Encoding visual-telephone video signal in 48 Kbps," *Picture Coding Symp.*, Tokio, Japan, Apr. 1986.

[15] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background", *Signal Processing: Image Communication I*, pp. 191-212, 1989.

[16] C. Cafforio, F. Rocca, and S. Tubaro, "Motion compensated image interpolation", *IEEE Trans. Commun.*, vol. 38, no. 2, pp. 215-222, Feb. 1990.

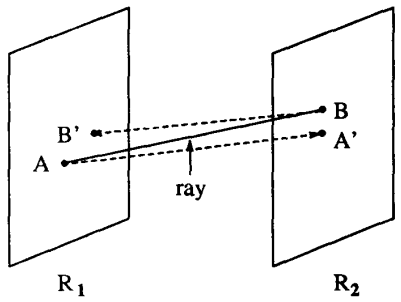

Figure 1: A ray joining two corresponding pixels A, B

Table 1: Statistics of ϵ for Figure 9.

Sequence No. 1 "salesman"	% pix $\epsilon < 0.05$	% pix $\epsilon < 0.1$	% pix $\epsilon < 0.2$	% pix $\epsilon > 0.5$
1 New technique	67.33	72.56	78.47	5.52
2 Cafforio *et al.*	65.97	71.17	76.73	12.49
3 Linear interp.	59.72	67.01	72.56	10.76

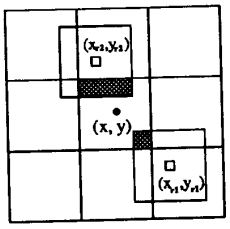

Figure 2: Area contributions on a pixel area

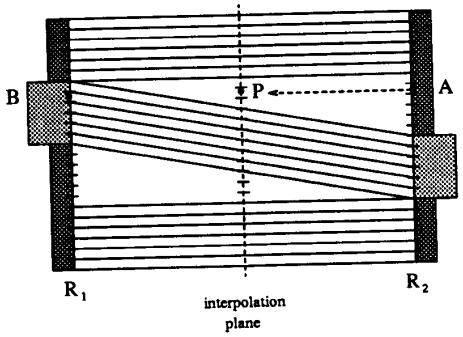

Figure 3: Interframe occlusion

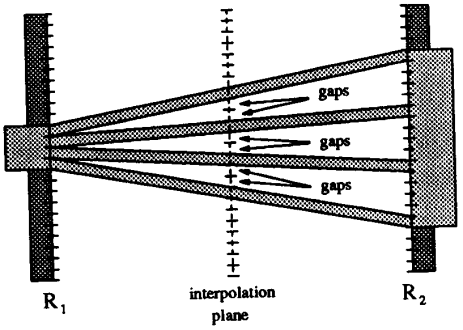

Figure 4: Interframe zooming

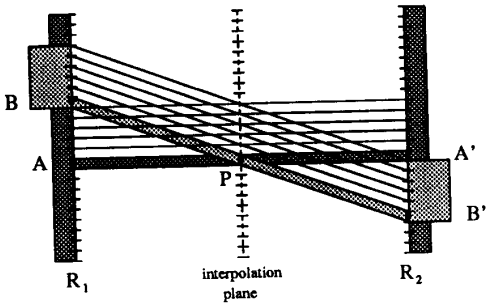

Figure 5: Figure-ground ambiguity

Figure 6: Sequence "salesman". Given frames 2,3,4 and interpolated frames 2.3.4

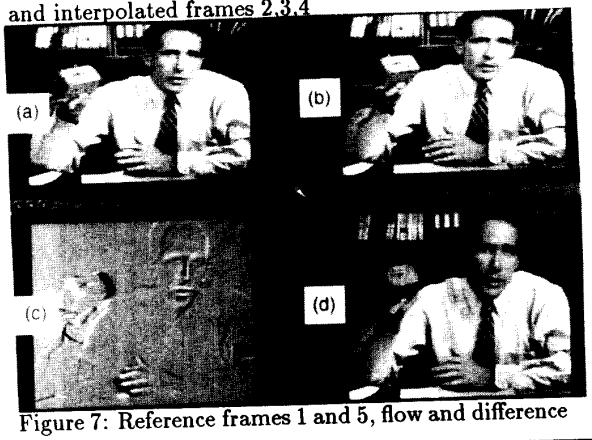

Figure 7: Reference frames 1 and 5, flow and difference

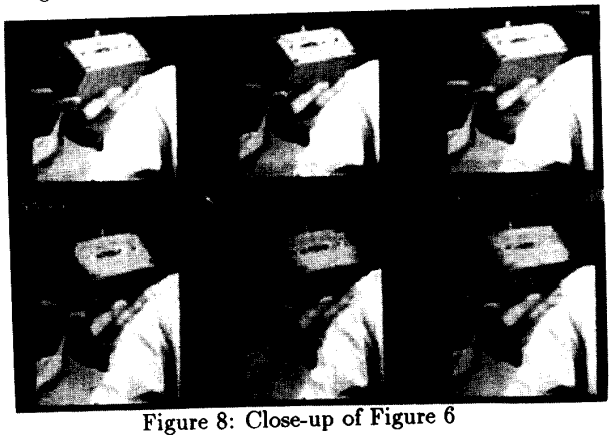

Figure 8: Close-up of Figure 6

Table 2: Statistics of ϵ for Figure 14.

Sequence No. 2 "people"	% pix $\epsilon < 0.05$	% pix $\epsilon < 0.1$	% pix $\epsilon < 0.2$	% pix $\epsilon > 0.5$
1 New technique	84.00	96.00	100.00	0.00
2 Cafforio *et al.*	70.66	81.33	89.33	1.33
3 Linear interp.	70.66	76.66	86.00	0.00

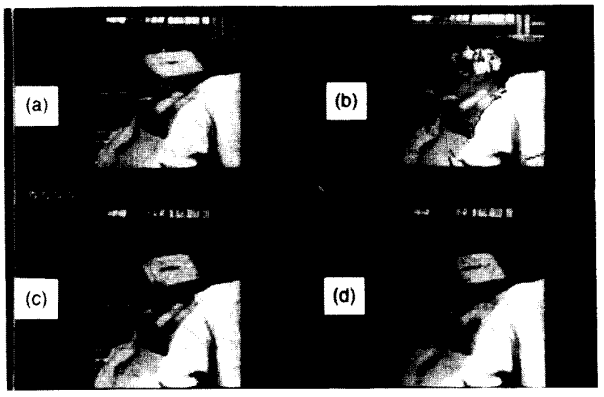

Figure 9: Comparisons with other techniques

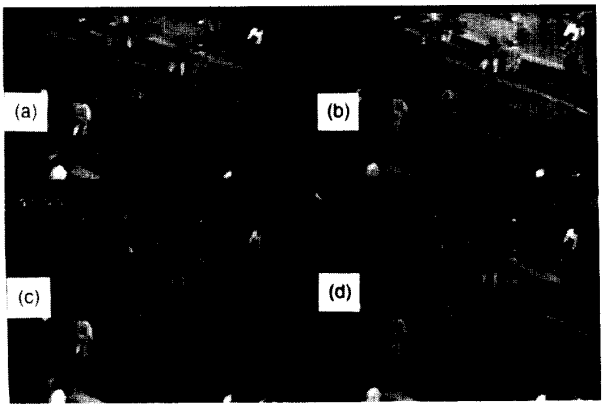

Figure 10: Sequence "people". Given frames 1,2,3,4

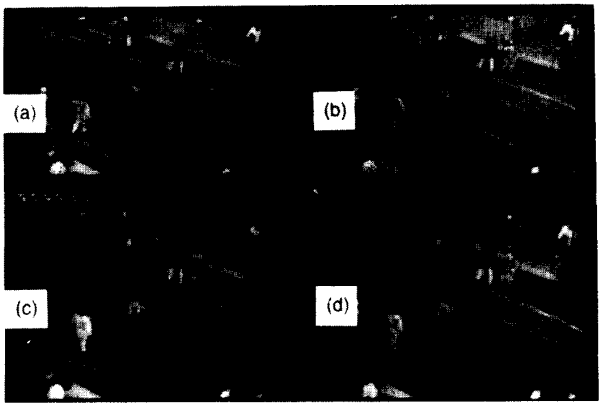

Figure 11: Reference frames 1 and 4, optical flow and interpolated frames 2 and 3

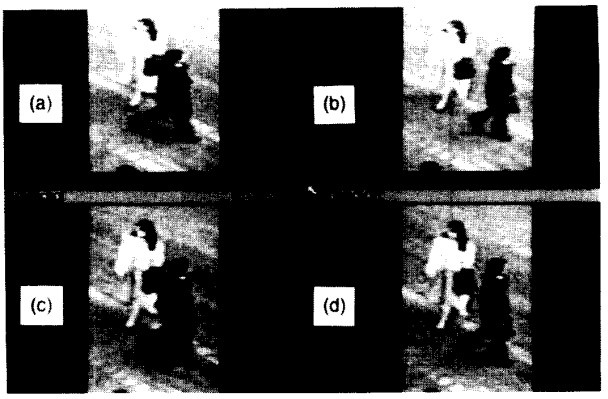

Figure 12: Close-up of Figure 10

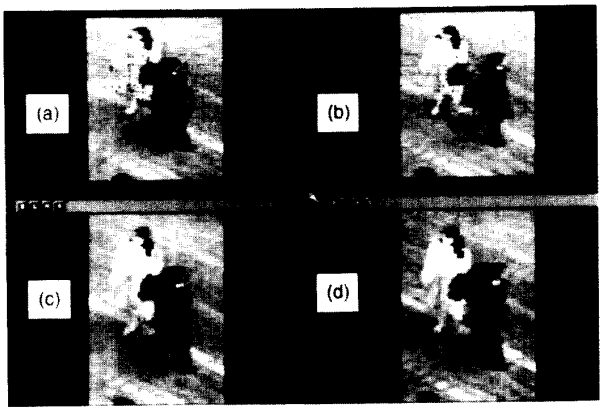

Figure 13: Close-up of Figure 11

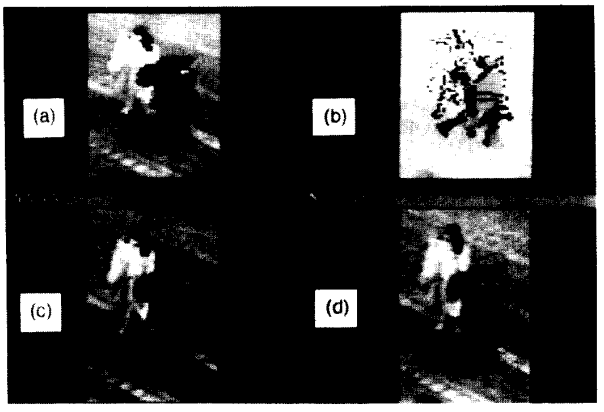

Figure 14: Comparisons with other techniques