Interpolation between Two Frames of a Video Clipping

Suhail T V Moideen Kutty
MS Student
MCIS

Dinesh Rao B
Associate Professor
MCIS

Sharmistha Sharma
MS Student
MCIS

ABSTRACT
The aim is to create an interpolated frame between two given video frames. Interpolated frame is the frame obtained by the calculation of the intermediate pixel positions of two given images. A set of interpolated frames can be generated between the two video frames.

General Terms
Grayscale, Pixel, Image Pixel Intensity, Pixel coordinates, Sub-frames.

Keywords
Frame, Interpolate Image, Out of Bound, Nearest Neighboring Pixel.

1. INTRODUCTION
In this technique we need to retrieve the pixel information from two known images and find out the intermediate positions of the image intensities. Thus obtain the developed interpolated image. Our method is based on the simple concept that a given pixel in the interpolated frames will trace out a path in the input images.

We wanted to find out which software is most appropriate for our project module. In the beginning we started of the most popular software, "Adobe AfterEffects" but found it difficult to implement our idea and no codes involved but could not achieve our goal. Hence we did a survey for a better image processing tool. Since in Bio-Medical Imaging, Warping, Morphing, Digital Image processing Fields [2] "Mat lab" is a prominent software tool.So our next step was to implement our idea on this tool. Linear interpolation methods are not suitable for edge concentrated application [2] because no special treatment is given to edges while zooming. Best quality data can't be obtained using this method. Non-linear methods are suitable for edge enhancement applications. In order to enhance the quality of the measurement it is better to use the method with more number of pixels than with the lesser ones. Higher order methods give better results than other methods but in much cases computational complexity increases. A combination of both Fuzzy and interpolation gives the best results. Coding of video sequences has led to the development in Television, Movies, Graphics and Gaming Industries. Video sequences contain a high degree of spatiotemporal redundancy that can be dramatically reduced to decrease the cost of transmission and storage [1]. Restoring the compact video sequences back to the original length is a challenging task.

2. ALGORITHM
Step 1: Read the images into variables.

Step 2: Convert these images from color to grayscale image.

.Step 3: Find the difference of the images [(ImageA - ImageB) and (ImageB - ImageA)]

Step 4: Retrieve the resolution of the image (resolution of all being the same).

Step 5: Initialize the Interpolate Image to ImageA.

Step 6: Compare images, if not equal and if the difference (ImageB - ImageA) is not equal to zero, then copy the ImageB to the interpolate Image.

Step 7: Initialize a counter1 and counter2 variables to 1.

Step 8: Compare each of the difference with threshold value, if greater, and then copy those pixel position values to two array variables for each Image. Increment the counter variables to 1.

Step 9: Initialize another variable counter3 and a flag to zero.

Step 10: Compare each of the pixel values between the difference images. The immediate eight neighboring pixel values are also compared with each pixel.

Step 11: Increment counter3 to 1 for each neighboring pixel values.

Step 12: If counter3 is greater than 5, then copy the intensity values to variables k and l.

Step 13: Then one of the image difference is taken and shifted to k rows and l columns.

Step 14: Subtract the above obtained image from the interpolate Image to get the original Interpolated Image.

3. PROCEDURE
First approach was to obtain two frames from a video; an intermediate frames in between them in time domain. Here we only try to get a frame exactly in between first and second input frames. For that purpose a normal WMP format video file was selected. Two snapshots of a video, which represents a ball moving in X direction for easier calculation Y is kept constant. In other words, the ball is moving in a horizontal direction and two frames have been randomly selected (named as ImageA and ImageB). In order to prove the concept of interpolation, we are trying to obtain an intermediate frame from the pixel position values of the input images.

We use "imread" instruction from two frames obtained from the images taken earlier from the Windows Movie Maker and then we converted them into gray scale images (intensity varying from 0 to 255) using the instruction rgb2gray.

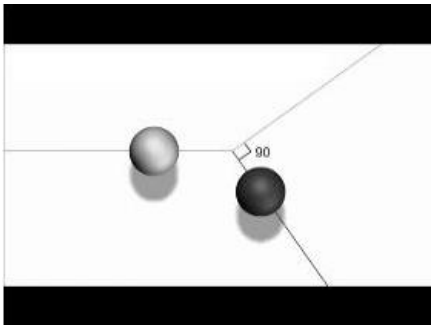

Fig. 1: First image (ImageA)

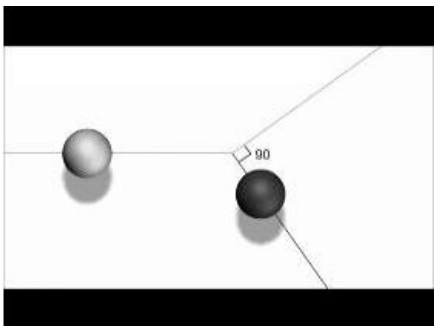

Fig. 2: Second image (ImageB)

Then we obtain the differences of the images say first image being ImageA and the second, ImageB. And we declare two variable differences of ImageA and ImageB (diff_ima) and another variable which is the difference of ImageA and ImageB and declare it as diff_imb. We display each of these images and their differences using the instruction "imshow (*image*)" in different figures simulated in different windows.

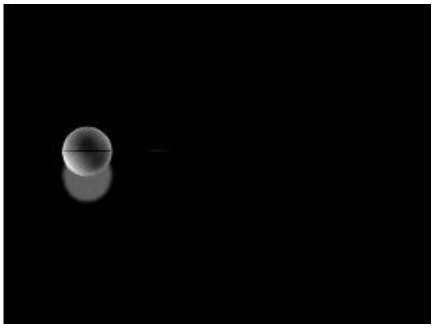

Fig. 3: Difference image of ImageA - ImageB (diff_ima)

Resolution of all the images (all being same) is obtain by using the below instruction:

[m,n] = size (image)

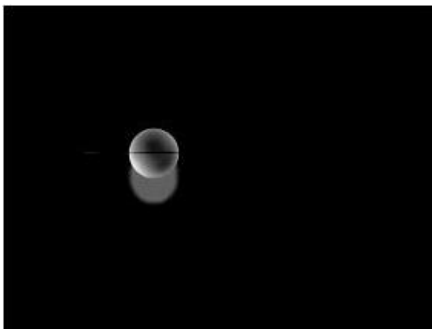

Fig. 4: Difference image of ImageB - ImageA (diff_imb)

Then we initialized one of the images (ImageA here) as an input to the new image, ImageI. Then we compare each pixel and whenever we find the differences that are copied to the new variable ImageI from the difference image of ImageB - ImageA (diff_imb) to obtain the image below:

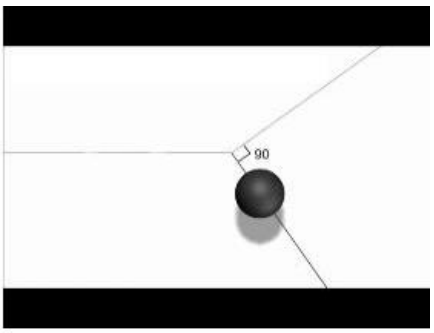

Fig. 5: Interpolate Image (without the interpolated frame)

For avoiding blurring and Ghosting or the faded in the interpolated image, we set a threshold gray image value approximated to 100 and setting two array variables to each of the image differences diff_ima and diff_imb. Then the arrays are initialized in our earlier stages of approach. Compare each of the difference with threshold value (say 100), if greater, then copy those pixel position values to the two array variables for each Image. Increment the counter variables by one.

A new variable is declared and a flag is initialized to zero. Only the difference array size is taken for the next iterative loop since we need only the sub-frames of the images where the differences are taken place. Now we move on to obtaining the position of the two difference Images and we place them into two variables k and l respectively. Further, we face similarities in the intensity of the same difference image.

4. NEAREST NEIGHBOURING PIXEL VALUE

Comparing the intensity value of each pixel was not sufficient, so we compared the neighbouring pixel value of each pixel. In other words, the intensity of each pixel along with their eighth different neighbouring pixel intensity values to compare between the difference images.

A counter variable is initialized and incremented during the comparison of each neighbouring pixel value such that a minimum five of them turns out to be valid condition.

5. ISSUES AND FINAL RESULT

Earlier the array assigned to the difference object has exceeded the resolution of the Interpolated Image.

Note: Infinite loop condition
Since the resolution of the images is 210X280, for comparing each pixel between the difference images we had to use six for loops (210X280X210X280X210X280), number of iteration extending up to 24 hours.

While taking the pixel position we encounter an error condition "Out of Bound", so we took the absolute value. Apart from this error, during averaging (for this particular case) there is also another invalid condition: where there cannot be fractional values for the pixel coordinates in the image represented as "matrices". So obtained value is taken separately to displace the position value of the difference image diff_ima. Eventually we had to reduce the number iteration by limiting the for loop condition to the value less than displacement, obtaining the Interpolated image with same resolution.

Final implementation was to move from the pixel positions of one of the Images (if ImageA is selected then the displacement will be = position + $(f(x), (f(y))$ and if ImageB is selected then the displacement will be = position - $(f(x), f(y))$. Following image is eventually obtained after several iterations:

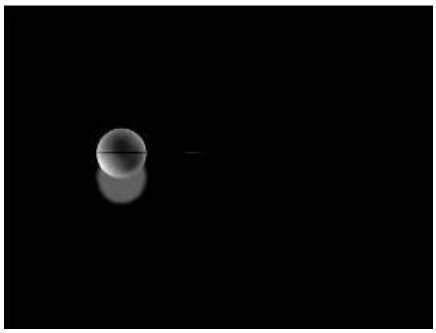

Fig 6: Interpolate Image in difference

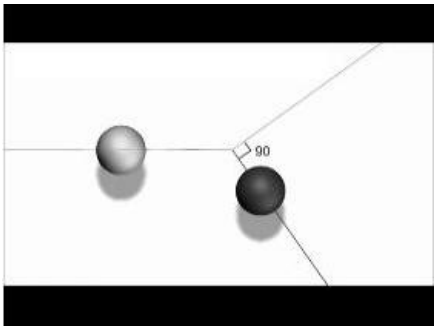

Fig 8: Interpolated Image in the original Image frame

Eventually we obtained the exact middle point (77, 96) of points (55, 96) and (98, 96).

Table 1. Tables of Images

Image	Pixel Information at the center of the ball (X,Y) coordinate
ImageA	(55,96)
ImageB	(98,96)
Difference of ImageA – ImageB	(98,96)
Difference of ImageB – ImageA	(55,96)
Neutral Image	NO BALL
Interpolate after resize	(77,96)
Final Interpolated Image	(77,96)

.

6. Formulae for Interpolation

$$f(x) = unsignedinteger\{ f1(i) - f2(j) \}$$

Here the $f1(\mathrm{i})$ represents the pixel intensity value of the first image difference (diff_ima) and $f2(j)$ represents that of the second image (diff_imb). We use unsigned integer in order to get non-negative values so that we get only the difference of the object, one position at a time.

7. APPLICATION

Our idea can be used in low bit rate application [6]. It is useful in controlling image qualities. Interpolation is found in enhancing gray scale images for future implementation in digital camera. Our interpolation algorithm also improves the subjective quality of the interpolated images over conventional linear interpolation; applied to areas where two video sequences, where time-varying features must be tracked [2]. The challenge for this approach is preserving accuracy when pixel displacements are large as in the case of high motion. Better performance can be achieved by applying more robust motion models that better account for the mapping between the reference and the current frame. Currently under investigation are mappings that transform the reference frame into the current frame without any error. This method is especially useful to resample images that must preserve accurate edges.

8. CONCLUSION

This paper has reviewed the growth of image interpolation and described recent advances in the field. Most interpolation methods implement iterations in a large number. We should focus our attention on two parameters; changing image intensity co-ordinates, and the number of iterations. We need to figure out optimum values for these two quantities. If any discrete value is changed, the resultant image also changes. The image does not converge to the same value as the number of iterations increase. Simple interpolation methods are preferred to other linear interpolation methods because of complexity constraints in processing.

There are several types of frame interpolation schemes already proposed. The early attempt utilized the frame repetition, linear interpolation, non-linear interpolation etc. This method always results in the motion jerkiness and the ghost effect. The linear interpolation technique in [2] produced the interpolated frame by performing a gradual

fading transition between the references frames similar to this paper. Unfortunately, the results were not pleasing due to the appearance of artifacts and the blurred moving areas. In this paper, we reconstruct a region in which the image intensities are changing.

9. REFERENCE

[1] WANG, H., XU, N., RASKAR, R., AND AHUJA, N. 2005. Videoshop: A new framework for spatio-temporal video editing in gradient domain. Proc. IEEE Conf. Computer Vision and Pattern Recognition 2, 1201.

[2] Dhruv Mahajan Columbia University, Fu-Chung Huang UC Berkeley, Wojciech Matusik Adobe Systems, Inc.Ravi Ramamoorthi UC Berkele,Peter Belhumeur Columbia University: Moving Gradients: A Path-Based Method for Plausible Image Interpolation.

[3] Bojan Vrcelj and PP Vaidyanathan, "Efficient implementation of all-digital interpolation," IEEE trans on ImageProcessing, pp. 1639–1647, November 2001.

[4] B.-D. Choi, S.-H. Lee, and S.-J. Ko, New Frame Rate Up-conversion Using Bi-directional Motion Estimation. IEEE Trans. on Consumer Electronic, vol.46, pp. 603-609, Aug. 2000.

[5] Frame Interpolation and Bidirectional Prediction of Video Using Compactly Encoded Optical-Flow Fields and Label Field Ravi Krishnamurthy, John W. Woods, *Fellow, IEEE*, and Pierre Moulin, *Senior Member, IEEE*

[6] Ying Bai Dept of Comput. Sci. & Eng., Johnson C. Smith University: On the comparison of fuzzy interpolation and other interpolation methods in high accuracy measurements, Fuzzy Systems (FUZZ), 2010 IEEE International Conference on.

[7] *M. Tagliasacchi, A. Trapanese, S. Tubaro , J. Ascenso, C. Brites, F. Pereira* :EXPLOITING SPATIAL REDUNDANCY IN PIXEL DOMAIN WYNER-ZIV VIDEO CODING. 2010