Translational Overture Effect through Polar Equations in Two-Dimensional Digital Animations.

M. Nuruzzaman, Ph.D.

Electrical Engineering Department, King Fahd University of Petroleum and Minerals,
Dhahran 31262, Saudi Arabia..

*E-mail: nzaman@kfupm.edu.sa

ABSTRACT

When two dissimilar image frames are displayed consecutively in a digital movie, the visual perception we receive is called an overture effect. Concentration has been given on studying the overture effect generation employing polar equations in spatial domain of the digital images. The effect minimizes visual discontinuity in a digital movie and algorithmic approach bypasses physical means of generating the effect. The proposition outlines a general mathematical representation describing a digital movie frame applying the notions and concepts of the discrete mathematics, the data structure of a digital movie both to the context of storage and pixel processing, a thorough description of the spatial domain based overture algorithm, and PC based animation for the algorithm implementation.

(Keywords: digital animation, spatial domain processing, special effect, 2D animation)

INTRODUCTION

Background: The history of commercial animation dates back from the 1800s. The traditional animation process requires persistent manual work in story writing, scene drawing, frame organizing, or snapshot taking. Digitally animated movies or their segments are frequently viewed on the Internet, in Video Compact Disc (VCD), and on Digital Video Disc (DVD) and the trend is increasing. Well-known movies such as Star Trek, Terminator 1 and 2, Spy Kids 1, 2, and 3, all of which render the gathering of thousands of digital movie segments. Some end uses of digital animations include the following: forensics, archeology, entertainment, education, space exploration, scientific visualization, film making, multimedia, and many more.

A movie segment is a collection of frames or still pictures typically 24 frames per second. Traditionally frames were set or drawn by hand and snapshots of these frames were taken to form a movie module. Eye catching image and breathtaking movies were thus produced and involved a great deal of hard work. Scores of experienced personnel are essential for this sort of movie making, not to mention that the money and time involvement were obviously a central factors. In order to have a movie segment virtually simulated, it requires the combination of two skills – a practical event (how it is taking place) and computer code (writing expertise to turn the event to a realistic movie module). When two dissimilar image frames are displayed consecutively in a digital movie, the visual perception we receive is called an overture effect.

Literature Review: Let us review some samples of the contemporary researchers' work in this field. The animation of fire on polyhedral surfaces is presented in [1]. Employing the notion of discrete straightest geodesics, the fire fronts directly on the surface of arbitrarily complex objects are discussed. The model [1] supports adaptivity of the fire fronts, multiple simultaneous fires, and merging of multiple fires. The proposed model appears to be suitable for the fire propagation rather than the fire rendering [1]-[4].

Augmented Reality (AR) handles the problem of dynamically enhancing the real world with the computer generated data or graphics of the virtual objects. There are two problems associated with: (a) precise alignment of real and virtual coordinate frames for overlay needs to be determined and (b) snapshoting the three dimensional environment including camera and object motions. The latter is important in the sense that users interact both with the real and virtual objects for interactive augmented reality applications.

These interactions may include collision detection [5]-[8], dynamic responses, and visual occlusions. Several research groups have explored the theme of camera tracking for the augmented reality. Some demonstrated vision based object registration and tracking for real time overlay which directly computes the image overlay instead of utilizing a pose calculation based image overlay. Some reported frame-based near real time calibration on a few fiducial marks. Some solved the problem by using an affine representation for coordinates and a transformation with a weak perspective approximation which avoids initial calibration and pose reconstruction.

Real time three-dimensional motion capture systems [9] have become commercially available in recent times. These systems hold promise as a means of producing highly realistic computer animation with more easiness and effectiveness than the conventional techniques afford. State of the art animation systems such as SOFTIMAGE™, Alias™, and Wavefront™ keep the provision for the simple motion editing tools for example curve fitting, global scaling, and translation. Editing iconic description poses a problem analogous to that of the editing a bitmapped image or a sampled hand drawn curve.

One approach to editing is to fit curves to the raw data, producing a keyframe like description that can be modified by editing the curve's control points. The drawback of this approach is that the fit curve is liable to need at least as many control points as would have been needed to keyframe the motion manually. Another approach is to edit by transforming the iconic description in a manner analogous to image morphing [9]-[11]. However commercial motion capture services possess propriety editing tools which employ curve fitting and control point adjustment rather than deformation but this approach is not suited to large scale transformation.

Making compelling animation of characters is a grim task. Motion capture data is by and large used to remake complicated character motions in digital animation systems but it is very difficult to reuse motion capture data. Because there are differences between two motion sequences in gestures and velocities in a real time interactive animation system, the two motion sequences are not commonly concatenated directly. The transition between two motion sequences needs to be created, and the concatenated point for motion blending must be chosen properly. In a wide sense there are two approaches for the complex character animation editing, one of which is called the motion blending [12]-[14] technique and the other of which is the video texture technique [15]-[16].

Directors of TV and commercial programs (nowadays internet ones also) convey their intentions to the actors and the production staff using storyboards. Not all their intentions are conveyed to the actors and the production staff since storyboards indicate only images of important moments of the scenes. In order to resolve such problems, a system that automatically generates animation storyboards called Virtual Studio System (VSS) is used to report information [17].

The VSS should analyze scenarios written in natural languages by the director and create the image automatically. The proposition in [17] imports scenario and create virtual storyboards which consists of two modules. The first module is a translating system that translates the natural languages of actors or characters to virtual studio modeling language or VSML and the second module is an animation generating module called VSML player. Abstract concepts become real when fully interpreted in 3D thereby requiring the expertise of 3D animation. Reference [18] reports contemporary usage of 3D animation in Malaysia to the context of NewTek's LightWave Software. One of the scenes of the recent sci-fi movie Matrix 2 is addressed. Composition of 2D and 3D (referred to as 2.5D) other than the normal procedure of modeling, texturing, and rendering of 3D scene is addressed.

There is a growing concentration from ecologists, physicians, biologists, etc. in the use of biologically inspired computer based systems for simulation and evolution of natural systems [19]. Traditional artificial intelligence tools together with other techniques in the field of artificial life or virtual life for designing naturally changing environments and a complex behavior for the inhabitants within these environments is employed in [20] in which a platform allows the user to create a virtual environment and inhabitants which perceive, think, and react in it.

There are two major mechanisms to specify the motion and behavior for a synthetic computer generated character – replication mechanism that reproduces the motion behavior of a real actor and modeling the mechanism where the motion and behavior are based on an algorithmic approximation. However, if the inhabitant that has been modeled [20] is a cockroach within its own habitat, this serves the purpose of generating a synthetic actor in the animation.

Our intensive research in the computer animation literature has made us learn that enough attention has not been given to overture effect generation in two dimensional digital animations. Because of its importance in ever increasing digitally animated world, we concentrate in this paper on studying the overture effect generation between two digital images through the use of polar equation in spatial domain. We study few cases of spatial domain processing because of the openness in the effect.

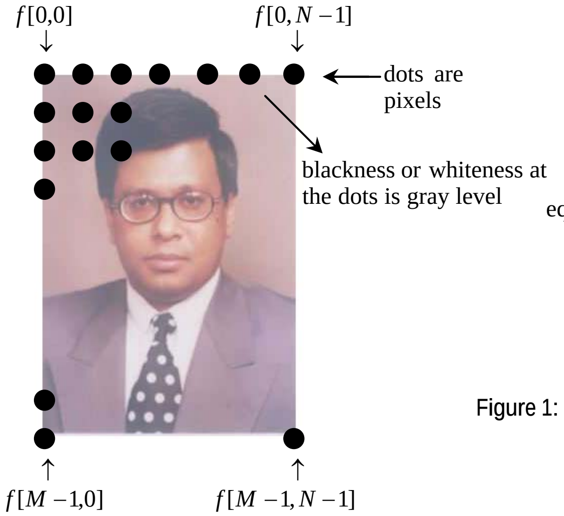
DIGITAL MOVIE FRAME ON FINITE DIFFERENCE MODEL

A real image frame is always a continuous one whereas a digital image frame is discrete both to the context of space and intensity. Mathematically discrete two dimensional intensity function $f[m,n]$ represents a monochrome digital image. Another way of explaining the $f[m,n]$ is that it is simply a rectangular matrix. If you reduce/enlarge the size of the image to half, so does the size of the rectangular matrix. Referring to the figure 1, we imagine some dots based on the resolution of the imaging system in the vertical and horizontal directions of the picture area for the digitization of the image which are the pixel or pel.

Assuming that the image area of the Figure 1 has $M \times N$ pixels, then the coordinate system of the pixels in which any pixel coordinate is denoted by (m,n). For an image of size $M \times N$ pixels, the pixel coordinate variation is as follows: $\begin{Bmatrix} m \text{ varies from } 0 \text{ to } M-1 \\ n \text{ varies from } 0 \text{ to } N-1 \end{Bmatrix}$ clearly m or n is positive integer.

Any practical digital image by discrete mathematics turns to two dimensional intensity function $f[m,n] \in 2^p$ where p is any positive integer which is closely related with the imaging resolution.

The representation of the monochrome image in terms of digital $f[m,n]$ indicates that pixel coordinates (m,n) transform to matrix element indexes of the Figure 1 and the 2^p becomes the matrix element corresponding to that coordinate thereby corresponding the four corner points of the image in the figure 1 to $f[0,0]$, $f[0,N-1]$, $f[M-1,0]$, and $f[M-1,N-1]$ for the upper left, upper right, lower left, and lower right respectively. Even though most image processing literature starts with the intensity function $f[m,n]$, a practical digital movie frame is color and its meaningful representation evolves from the color theory. It is an established fact that all colors are derived from three basic colors – red, green, and blue. Splitting the color of a pixel into red-green-blue components employs three different intensity functions or in other words three different image matrices like the $f[m,n]$.



equivalent to
$$\begin{bmatrix} f[0,0] & f[0,1]\cdots & f[0,N-1] \\ f[1,0] & f[1,1] & \\ f[2,0] & & \\ & \vdots & \\ f[M-1,0] & \cdots & f[M-1,N-1] \end{bmatrix}$$

digital image matrix $f[m,n]$

Figure 1: Illustration of a Digital Image and its Matrix Equivalent .

Let us call the three color component matrices as $r[m,n]$, $g[m,n]$, and $b[m,n]$ respectively, each of which is identical in size (or $M \times N$) and bear the aforementioned meanings pertaining to the discrete space variables m and n. Therefore any digital image movie frame is mathematically just the matrix triplet $\begin{Bmatrix} r[m,n] \\ g[m,n] \\ b[m,n] \end{Bmatrix}$ and whose matrix model is shown in Figure 2 (Q= 2^p).

$$\begin{array}{ccccc} r[m,n] & & g[m,n] & & b[m,n] \\ \begin{bmatrix} \text{red image} \\ \text{matrix} \\ \text{elements,} \\ \text{positive} \\ \text{integers} \\ \text{say } 0 \text{ to } Q-1 \end{bmatrix} & \& & \begin{bmatrix} \text{green image} \\ \text{matrix} \\ \text{elements,} \\ \text{positive} \\ \text{integers} \\ \text{say } 0 \text{ to } Q-1 \end{bmatrix} & \& & \begin{bmatrix} \text{blue image} \\ \text{matrix} \\ \text{elements,} \\ \text{positive} \\ \text{integers} \\ \text{say } 0 \text{ to } Q-1 \end{bmatrix} \end{array}$$

Figure 2: Matrix Representation Based Digital Image Frame Model.

DEFINING THE OVERTURE PROBLEM IN 2D DIGITAL ANIMATION

In order to understand the overture effect, the actual nature of a digital movie needs to be addressed first. A digital movie frame is just the matrix triplet $\begin{Bmatrix} r[m,n] \\ g[m,n] \\ b[m,n] \end{Bmatrix}$ and a digital movie segment is born by placing such triplet one after another. Figure 6 presents the schematic representation of a digital movie segment. Most PCs hold the provision for 15 triplets per second but commercial animations require 24 or higher triplets per second. So if there are M triplets and the system supports f_r triplets per second, the duration of the movie segment becomes $\frac{M}{f_r}$. Digital animation making means generation of the M triplets based on the storyboard.

Suppose on a computer monitor screen, one still image frame is being displayed (Figure 3). To the context of the digital animation, it means every $\frac{1}{f_r}$ sec, we bring the same image frame and display it. When another dissimilar image frame is placed, a sense of discontinuity appears. This discontinuous visual perception is termed as the overture effect. Fourier theory says that any discontinuity inherits infinite sinusoids and the overture effect also acts likewise. One can exploit the variability hidden in the infiniteness to generate some animation between two dissimilar movie frames. The paper is all about the overture generation but the question is which overture? Since there are infinite possibilities, the problem is an open one. If the user defines some scene overture, it means the user concentrates in one overture out of infinity. We have chosen the polar equation based transition and developed an algorithm for the overture realization.

In reality the overture is conducted through the practical means. Let us say the picture of the Figure 3 is being displayed in a movie. We intend to turn off the scene so that at the end we end up with the image frame of the Fgure 4.

While taking the snapshot by the video camera, the camera focus is kept on the real image frame considering that the frame is available. If the picture is to be seen, the light on the picture must remain on. Manually one turns off the light on the picture area to obtain the frame of the Figure 4. Picture to black screen transition does not happen instantaneously but takes split-second due to the light switching off. The visual event received by the camera during the transition is the overture effect which is conducted completely manually.

Figure 3: A Picture.

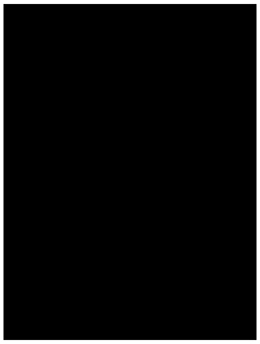


Figure 4: The picture in Figure 3 is Turned to Black.

The overture takes place not only for the transition of two pictures but also for the motion of the camera. But again the camera has to move physically to render the effect. That is how the overtures are generated in most digital movies.

HOW THE APPROACH DIFFERS FROM THE CONVENTIONAL ONE?

Our proposition is based on the spatial domain processing. We make an effort to simulate the user-defined overture virtually without the physical movement of the real scene, photograph, or camera.

The real world is three dimensional (3D). While one takes the snapshot of a real world scene, the real world scene transforms to a two dimensional picture which we call digital movie frame and whose mathematical model is discussed earlier.

Having the two pictures before and after the overture transition available, we gain access to the pixels or in other words to the variables m and n of both triplets. Apart from the spatial domain access, admission to the color level might be required depending on the overture. Anyhow Figure 5 depicts the basic principle in our proposition for the generation of the two dimensional digital video or animation.

In last section we explained how the picture in Figure 3 turns to black by turning the lights off on the picture area. Now we explain how the algorithmic approach simulates the same phenomenon without manual turning off of the light.

Let us assume that the red, green, or blue color levels of the digital image frame change in [0,255] scale. Color theory illustrates that the levels 0 and 255 for each color represent the absence and complete presence respectively. Whatever be the level value for any of the colors at any pixel (m,n) in the triplet $\begin{Bmatrix} r[m,n] \\ g[m,n] \\ b[m,n] \end{Bmatrix}$, the pixel value must turn to $\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$ in order to turn the pixel black at the end. We assume that the transition from $\begin{Bmatrix} r[m,n] \\ g[m,n] \\ b[m,n] \end{Bmatrix}$ to $\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$

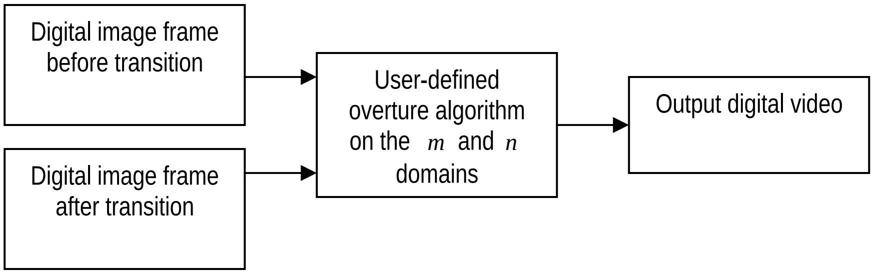


Figure 5: Basic Structure of the Proposed Algorithm.

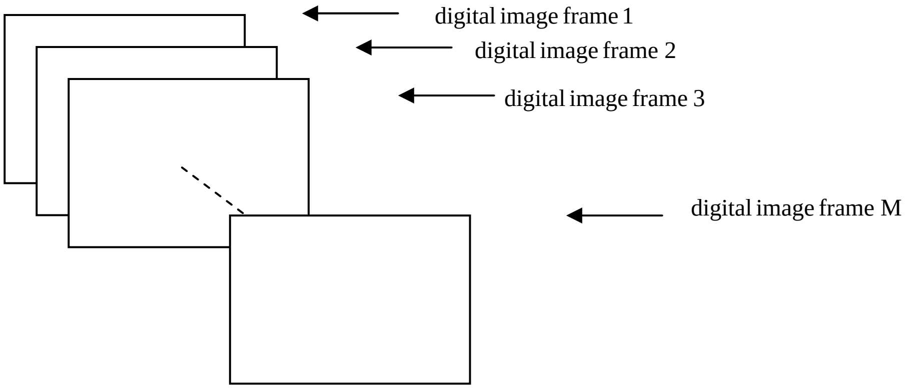


Figure 6: Formation of a Movie Segment from the Digital Image Frames.

happens in f_r digital image frames which is user-defined. Also we assume that each of the three color level variations takes place linearly. Considering the red image, the discrete color level selection at any pixel coordinate (m,n) occurs as $r[m,n]$, $\frac{(f_r-2)r[m,n]}{f_r-1}$, $\frac{(f_r-3)r[m,n]}{f_r-1}$, $\frac{(f_r-4)r[m,n]}{f_r-1}$, ... 0 within the f_r frames respectively. Similar color selection also happens for the other two color images.

Now there are f_r digital color image frames or triplets at our disposal. One can put these image frames one after another like the figure 6 to form the digital movie. Next question is what the duration of the movie is. Suppose our computer graphics card supports f_s frames per second. So the digital movie we generated (i.e. turning the image of the Figure 3 to black) becomes $\frac{f_r}{f_s}$ seconds long. Thus we overcome manually turning off the switch as illustrated in the last section.

POLAR EQUATION BASED OVERTURE ALGORITHM

Selecting some specific geometric polar equation and applying that on the pixel pattern in a digital movie frame brings forth different overture digital animations, examples of which can be $r=\cos\theta$, $r=\sin 2\theta$, etc. Any image plane pixels are mapped to the user-defined polar curve through rectangular to polar point transformation which is given by $\left\{\begin{array}{l} r=\sqrt{(x-x_0)^2+(y-y_0)^2} \\ \theta=\tan^{-1}\left(\frac{x-x_0}{y-y_0}\right) \end{array}\right\}$.

The point (x_0,y_0) in the image plane is also user-defined. If (x_0,y_0) is chosen halfway within the image that is $x_0=\frac{M}{2}$ and $y_0=\frac{N}{2}$, the first frame of the movie or animation starts from the middle of the image. If we choose some other point somewhat around the corner, the animation starts in that point. In order to develop the algorithm on this overture, the image pixel grid generation concept is vital. Figure 7 is the pixel grid arrangement of any digital image frame.

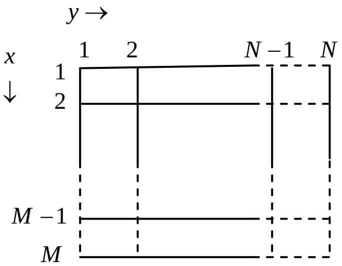


Figure 7: Arrangement of the $x-y$ Coordinates of the Pixels of a Digital Movie Frame.

If we separate the x and y coordinates of the image pixel maintaining the pixel position common, we end up with two repetitive matrices like $X = \begin{bmatrix} 1 & 1 & & 1 \\ 2 & 2 & & 2 \\ 3 & 3 & \cdots & 3 \\ \vdots & \vdots & & \\ M & M & & M \end{bmatrix}$ and $Y = \begin{bmatrix} 1 & 2 & 3 & \cdots & N \\ 1 & 2 & 3 & \cdots & N \\ & & \vdots & & \\ 1 & 2 & 3 & \cdots & N \end{bmatrix}$ for x and y respectively. These two matrices hold some interesting properties. The first of which is both are identical in size with that of the original image (i.e. $M \times N$). The first matrix is composed of N repetitive columns in which each column is formed from the positive integers 1 through M (row dimension of the original image). The second one comprises from M repetitive rows in which each row is formed from the positive integers 1 through N (column dimension of the original image). Knowing the image pixel size, one can easily generate these two rectangular grid matrices.

Figure 8 presents the flow chart of the digital animation using polar curve based overture. It is assumed that the digital movie frames before and after the overture are defined that is finite difference based frame function is available. The algorithm is explained in the following:

Step 1: The given two images are acquired in order to have the two RGB triplets $\left\{\begin{array}{l} r_1[m,n] \\ g_1[m,n] \\ b_1[m,n] \end{array}\right\}$ and $\left\{\begin{array}{l} r_2[m,n] \\ g_2[m,n] \\ b_2[m,n] \end{array}\right\}$. From the acquired images, the triplet dimension $M \times N \times 3$ is obtained

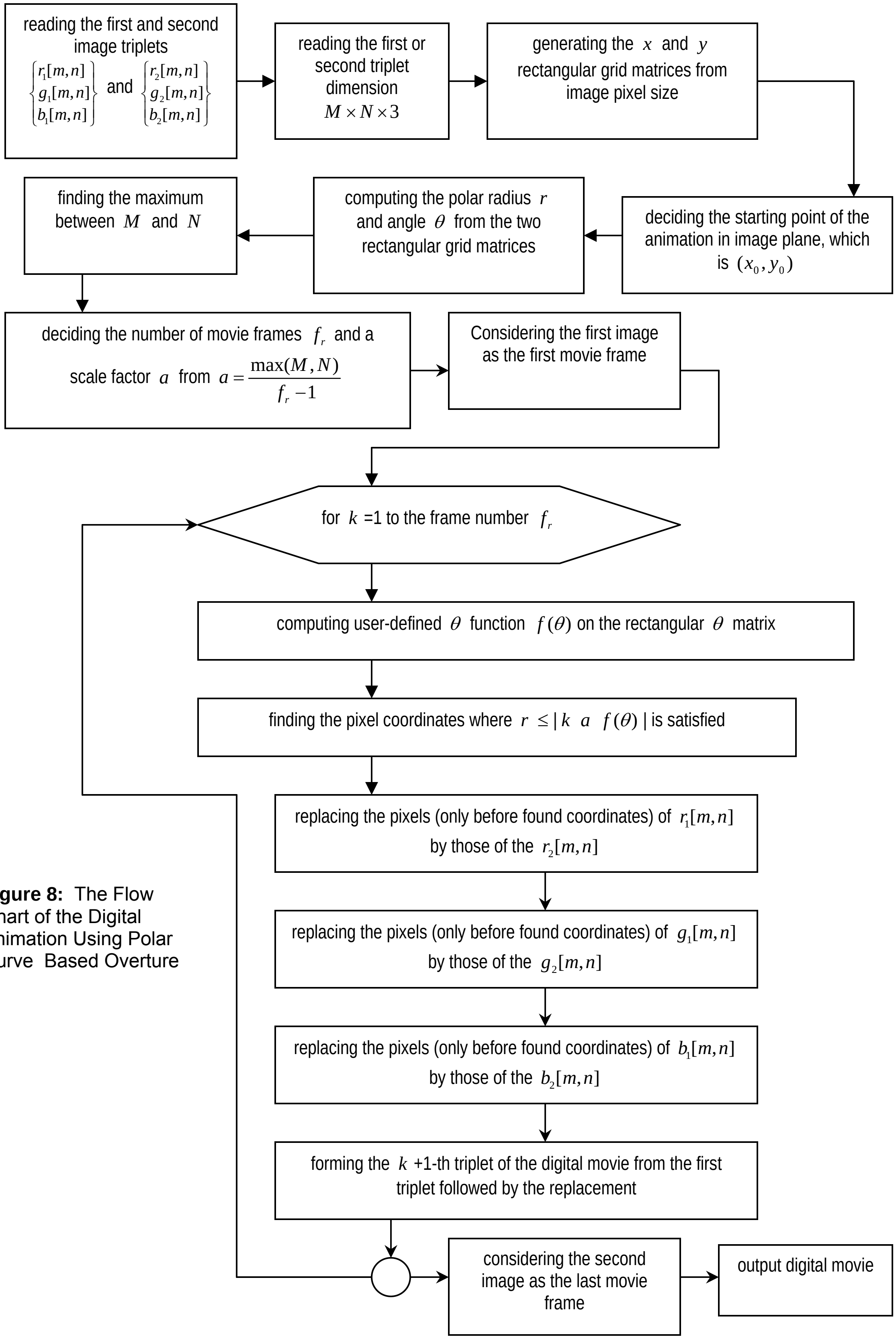


Figure 8: The Flow Chart of the Digital Animation Using Polar Curve Based Overture

Step 2: From the dimension of the image frame, the rectangular grid matrices X and Y are generated so that each one has the given image pixel size and elements in the matrix correspond to the image pixels.

Step 3: We decide the coordinates (x_0, y_0) from which the animation starts. Primarily the coordinates must be within the image plane as far as two dimensional animation is concern. The selected domain belongs to $1 \le x_0 \le M$ and $1 \le y_0 \le N$ for now.

Step 4: We calculate the radius vector and polar angle matrix grids from the starting point (x_0, y_0) by using the earlier quoted (r, θ) transformation. This gives rise two more identical size matrices like the x and y rectangular grid matrices whose matrix elements also correspond to the image pixels' position.

Step 5: We find the maximum between the M and N. If we do not find the maximum, the algorithm might stack in the square image within the given rectangular frame based on the minimum between M and N.

Step 6: We decide the number of movie frames f_r and a scale factor a from $a = \frac{\max(M, N)}{f_r - 1}$. The scale factor basically controls the radius vector values. It is the scale factor of the geometric shape that pronounces the shape of the overture change. For a single scalar, we would not see any animation. We assumed the scale factor a to be inversely proportional to the frame number. Assumption of any other variation may render different visual effects.

Step 7: We consider the first image as the first movie frame.

Step 8: We enter into a programming loop where the loop index k changes from 1 to the number of frames needed with positive integer increment. Inside the loop for every index k, we carry out the following:

(a) compute the user-defined θ function $f(\theta)$ on the rectangular θ matrix. For example choosing $f(\theta) = \cos\theta$ means taking the cosine on all elements of the θ matrix found in step 4.

(b) find the pixel coordinates where the condition $r \le |k\ a\ f(\theta)|$ is satisfied. Of coarse, the comparison takes place on every pixel basis keeping in mind that both the r and $f(\theta)$ share identical matrix size. The absolute value gives equal overture on either side of the user-defined starting point (x_0, y_0). Let us call the set of the pixels as S ($S \in r \le |k\ a\ f(\theta)|$).

(c) replace the pixels in S of the $r_1[m,n]$, $g_1[m,n]$, and $b_1[m,n]$ by those of the $r_2[m,n]$, $g_2[m,n]$, and $b_2[m,n]$, respectively.

(d) form the k +1-th triplet of the digital movie from the first triplet followed by the replacement.

(e) consider the second image as the last movie frame.

Step 9: Finally we place all triplets in order to form the complete video data stream which is our digital video

SIMULATION RESULTS

In this section we present the simulation results that demonstrate the application of the algorithm. In contrast conventional papers our result is a digital video so paper display of the digital video is not possible yet we include one sample frame out of the digital video just to prove the worthiness of the algorithm.

As dissimilar image of the overture we chose the images of the two kids and water lily as shown in the figures 9 and 10 respectively. Each RGB triplet has the pixel size 600×800 with each color level in [0,255]. As digital video frame number, f_r=9 is chosen. The digital video is generated by choosing $f(\theta) = \cos\theta$ with (x_0, y_0)=(34.5,67.5).

Figure 11 shows the 7th frame in the video. If the reader is interested in viewing the digital video, a request can be made at nzaman@ymail.com. You can use Microsoft windows media player or other to watch the video if obtained. The file size of the generated video is 12.659 MByte.

Figure 9: Picture of Two Kids.

Figure 10: Picture of Water Lily.

Figure 11: One Overture Frame Using Polar Equation.

The video generation took place on RGB image format but the algorithm equally worked for other types of images for instance indexed, intensity, or bitmapped ones. As example we presented the result for $f(\theta) = \cos\theta$. It is indeed amazing that simple polar curve equation change yields different animation. Some other curves such as $r = \cos 2\theta$, $r = \frac{1}{4}\cos ec 3\theta$, $r = \cos\theta + \sin\theta$, etc can be considered as well for the animation whose result is not attached for space reason.

CONCLUSION

An overture based two-dimensional digital animation algorithm by using polar equation is illustrated and implemented in this paper. The algorithm is proven to be effective and has the adaptability to intake different image types and polar equations. Since polar curve equations render various digital animations, commercial usefulness of the algorithm is easily understood.

The animation algorithm can readily be applied to any movie event. Only the translational overture cases are considered in future the rotational and warping cases on the overture will be studied.

ACKNOWLEDGEMENT

The author acknowledges King Fahd University of Petroleum and Minerals (KFUPM) for providing the library and computer facilities necessary for conducting this research.

REFERENCES

1. Lee, H., L. Kim, M. Meyer, and M. Desbrun. 2001 "Meshes on Fire". *Proceedings of the Eurographics Workshop on Computer Animation and Simulation*. September 2001.
2. Chiba, N., S. Ohkawa, K. Muraoka, and M. Miura. 1994. "Two-Dimensional Visual Simulation of Flames". *The Journal of Visualization and Computer Animation*.
3. Hargrove, W.W. 200. "Simulating Fire Patterns in Heterogeneous Landscapes". *Ecological Modeling*.
4. Stam, J. and E. Fiume. 1995. "Depicting Fire and Other Gaseous Phenomena Using Diffusion Process". *Computer Graphics Processing, ACM SIGGRAPH*.
5. Koller, D., G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. "Real-Time Vision-Based Camera Tracking for Augmented Reality Applications". *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST-97).* September 1997. 87-94.
6. Breen, D.E., R.T. Whitaker, E. Rose, and M. Tuceryan. 1996. "Interactive Occlusion and Automatic Object Placement for Augmented Reality". *Proceedings of Eurographics' 96.* Elsevier Science Publishers: Poitiers, France. 11-22.

7. Grimson, W., T. Lozano-Perez, W. Wells, G. Ettinger, and S. White. 1994. "An Automatic Registration Method for Frameless Stereotaxy". *Image, Guided Surgery and Enhanced Reality Visualization, IEEE Conference on Computer Vision and Pattern Recognition*. Seattle, WA. June 19-23, 1994. 430-436.

8. Klinker, G.J., K.H. Ahlers, D.E. Breen, et al. 1997. "Confluence of Computer Vision and Interactive Graphics for Augmented Reality". *Teleoprerations and Virtual Environments – Special Issue on Augmented Reality*. 6(4):433-451.

9. Witkin, A. and Z. Popović. 1995. "Motion Warping", *Computer Graphics Proceedings, SIGGRAPH 95*. Los Angeles, CA. August 6-11.

10. Boulic, R., Z. Huang, N.M. Thalmann, and D. Thalmann. 1994. "Goal Oriented Design and Correction of Articulated Figure Motion with the Track System". *Computers and Graphics (UK)*. 18 (Jul-Aug):443-52.

11. Ko, H. and N.I. Badler. 1993. "Straight Line Walking Animation Based on Kinematic Generalization That Preserves the Original Characteristics". *Graphics Interface Proceedings'93*. May 1993. 9-16.

12. Qingge Ji. 2005. "Weight Computation for Motion Blending", *International Conference on Computer Graphics, Imaging and Vision (CGIV05)*. Beijing, China. July 26-29, 2005. 144-150.

13. Bruderlin, A. and L. Williams. 1995. "Motion Signal Processing". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press. 97-104.

14. Rose, C., M.F. Cohen, and B. Bodenheimer. 1998. "Verbs and Adverbs: Multidimensional Motion Interpolation". *IEEE Computer Graph. Appl.* 18(5): 32-40.

15. Arikan, O. and D.A. Forsyth. 2002. "Interactive Motion Generation from Examples". *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press. 483-490.

16. Kovar, L. and M. Gleicher. 2003. "Flexible Automatic Motion Blending with Registration Curves". *Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer Animation*. Eurographics Association. 214-224.

17. Haginoya, R., H. Aoyama, K. Honda, and K. Odaka. 2004. "Automatic Animation Generation from Natural Languages". *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV04)*. School of Mathematical Sciences, USM. 103-109.

18. Kamsin, A., M.I.M. Sabri, and A.L.S. Sze. 2004. "The Use of 3D Animations in Malaysia: The Incorporation of Virtual Characters with Live Environment Using Light Wave". *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV04)*. School of Mathematical Sciences, USM. 110-116.

19. Beer, R.D., R.D. Quinn, H.J. Chiel, and R.E. Ritzmann. 1997. "Biologically Inspired Approaches to Robotics". *Communications of the ACM*. 40(3):31-38.

20. Pina, A. and F.J. Serón. 2000. "Behaviour Modelling, Computer Animation and Ecology". *Proceedings of the IASTED International Conference of Computer Graphics and Imaging*. November 19-23, 2000. Las Vegas, NV. 313-318.

SUGGESTED CITATION

Nuruzzaman, M.. 2009. "Translation Oveture Effect through Polar Equations in Two-Dimensional Digital Animations". *Pacific Journal of Science and Technology.* 10(1):300-309.

Pacific Journal of Science and Technology