J Math Imaging Vis (2012) 42:50–63
DOI 10.1007/s10851-011-0273-3

# Image Morphing in Frequency Domain

**M. Shahid Farid · Arif Mahmood**

Published online: 24 March 2011
© Springer Science+Business Media, LLC 2011

**Abstract** Image morphing is often used in film and television industry to create synthetic visual effects by smooth transformation of one object into another. Based upon spatial representation of images, several image morphing techniques have been proposed. Simple spatial techniques, for example cross-dissolve, suffer from lack of smooth transformation while better quality techniques, like mesh warping or field warping, have significant computational complexity. In this paper we present a simple but good quality image morphing technique based upon frequency domain representation of images. Transformation from a source image to a target image takes place by mixing low frequencies of the source image and high frequencies of the target image in varying proportions. The proposed technique has been applied to a wide variety of images. The resulting sequence of images are better in visual quality and faster in execution time.

**Keywords** Image morphing · Metamorphosis · Frequency domain morphing · Image transformation · Warping

This work was partially supported by a research grant from University of the Punjab, Lahore. We are thankful to the VC of the University of the Punjab, Prof. Dr. Mujahid Kamran for approving this research grant to carry out this research.

M.S. Farid (✉) · A. Mahmood
Punjab University College of Information Technology,
University of the Punjab, Lahore 54000, Pakistan
e-mail: shahid.fareed@pucit.edu.pk

A. Mahmood
e-mail: arifm@pucit.edu.pk

## 1 Introduction

Image morphing is an important area due to its applications in the creation of special visual effects for entertainment in film, television industry [31, 39], in education and in the field of medicine [6]. These effects are generated by smooth transformation of the objects and the colors of one image to the objects and the colors of another image. For this purpose, several spatial domain image morphing techniques have been proposed [17, 21, 22, 25, 30, 44]. Some of these techniques are computationally simple, for example, the cross-dissolve technique [39], however these simple techniques do not exhibit good quality results. Whereas, some other techniques like mesh warping [22], field warping [17] and multilevel free form deformation [21] are better in quality, however these techniques exhibit significant computational complexity. In this paper we propose a new morphing technique which is computationally simple and yields good quality results.

The image morphing problem may be considered as the transformation of a source image into a target image by generating a sequence of intermediate images. If these images are frames of a video sequence, the source image may be considered to be 'fading out' and the target image to be 'fading in' with the passage of time. Each frame in the intermediate sequence contains information from both the source and the target images. Therefore areas in the intermediate frames, where the source and the target images are misaligned, may contain 'ghost-effect', which is undesirable and should be properly handled. In simple image morphing techniques like cross-dissolve, ghost-effect has not been properly handled while the more complex techniques, like mesh and field warping, properly handle this artifact by introducing an additional step that aligns features in the source and the target images.

Image morphing techniques which effectively handle the ghost-effect may be organized into three basic steps: (1) feature specification, (2) warp generation and (3) transition control. In step (1), perceptually similar point correspondences are marked across the source and the target images. Both images may contain very different contents, therefore automatic detection of perceptually similar points is a tough process and generally these correspondences have been manually marked in the image morphing techniques. In field morphing approach, instead of point correspondences, corresponding lines are specified in the source and the target images. In mesh morphing, correspondences are specified in the form of feature polygons.

In step (2), a geometric transformation model is computed by using the feature correspondences marked in step (1). That model is then used to geometrically align the target image to the source image in order to minimize the ghost-effect. If objects in the source and the target images are significantly different in size, then a sequence of geometric models may be required to generate a sequence of transformed target frames. Information from each of the transformed target frame will be properly mixed with the source image to generate an intermediate frame. This mixing of information is controlled through a process known as 'transition control'.

In the intermediate frames, the source image information is gradually reduced and the target image information is gradually increased. This variation of the source and the target image information is considered as the 'transition control'. A proper transition control function ensures a visually smooth transformation of the source image into the target image.

Our proposed image morphing technique is based upon the frequency domain representation of the source and the target frames. We observe in frequency domain representation, high frequencies capture local details while low frequencies capture the global structure of the image. We exploit this fact for the creation of smooth transformation of source image into target image by blending low frequencies of the source image with the high frequencies of the target image. Our transition-control process consists of a set of low pass and high pass filters with gradually varying cutoff frequencies. For a particular intermediate frame generation, the source image is low passed and the target image is high passed and the filtered image frequencies are blended together.

The proposed algorithm has been implemented and tested on different types of target and source images. The experimental results demonstrate that the proposed algorithm produces results better in visual quality and faster in running time than other morphing techniques.

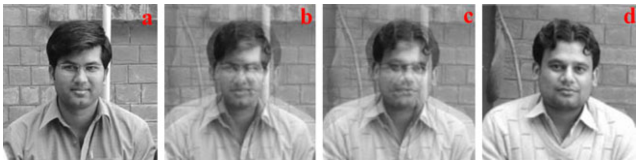

**Fig. 1** (**a**) source image, (**d**) target image, (**b**) and (**c**) are two intermediate frames

## 2 Related Work

Several morphing techniques have been introduced in the last couple of decades in spatial domain. Cross-Dissolve morphing technique is the most fundamental way to morph two images [38]. In this approach source image fades out, with the passage of time, and target image fades in. In simple words each pixel in source image starts transforming to that of target image. The problem with Cross-Dissolve method is the double exposure in the misaligned regions. This effect is referred to as 'ghost' and is particularly apparent in the middle frames, see Fig. 1.

Mesh Warping or Mesh Morphing was pioneered at Industrial Light & Magic (ILM) by Douglas Smythe for use in movie Willow in 1988 [22]. Mesh Warping have been organized in three steps: feature specification, warp generation and transition control. Matching features of the source and target images were specified to align the images which was referred to as warping or warp generation. In transition control, selected control points of the source image are mapped to that of target image by time. In mesh warping, 'ghost' was removed by performing transition control locally rather than globally. Corresponding features in two images were specified in the form of meshes. The points of source image were mapped over time to coincide with the points in the target image with color transition.

Meshes may be a convenient way to specify pairs of feature points, however, sometimes meshes turn out to be cumbersome to use. The field morphing approach developed by Bier and Neely [17] simplified the feature specification by means of line pairs. A pair of corresponding lines in the source and the target images was used to define a coordinate mapping between the source and the target images. The pixels in the target image were then warped with respect to their distance from the feature lines. Since multiple lines were usually specified, the displacement of a point in the source image was a weighted sum of the mappings due the each line pair with the weights attributed to the distance and line length.

An important thing in any morphing algorithm is the alignment of the two images, which has often been achieved by defining a mapping function between them. Features points has often been marked using correspondence, lines, meshes and snakes [18, 42]. An image alignment technique based on line is described in [36]. A line segment based morphing algorithm is given in [17]. Wolfgang Krüger described a line segment based image alignment technique in [19] to compute the mapping function between line segments of a map and line segments extracted from an image. A frequency domain image registration technique is presented in [20]. The entire registration process is done in frequency domain. *Polyaffine*, an image registration framework is presented in [2] that warps images with a small degree of freedom (DOF). A 3D real time registration technique is given by Park et al. in [28]. The technique is point-to-point based and exploits the power of graphical processing unit (GPU) for image registration. The technique uses points pair to define the line segments and a mapping function is defined for registration. There are several other techniques [3, 9, 11, 14, 29] to align digital images.

Ruprecht and Muller constructed a new mapping function for interpolation that was a combination of several basis functions [30]. This approach was named as *Radial Basis Functions or Thin Splines*. A level-set approach to image blending [37] generates the intermediate images by successively minimizing the difference matrix of the source and the target image. Lee et al. [24] extended the idea of morphing two reference images to $n$ reference images. They formulated the $n-1$ dimensional simplex model of $n$ images, which is a polyhedron with $n$ vertices. Any coordinate in this polyhedron represents a morphed image and barycentric coordinates of that point determines the weights to be used to blend the $n$ images. In Skeleton-driven approach [4], a feature graph was built from skeletons of the source and the target images and then intermediate skeletons are constructed.

In *layer based morphing* technique [10], objects are segmented into regions in the form of separate layers. Morphing is done between specific object segments. In this technique, ghost effect is locally minimized. Image morphing using piecewise linear curves in is proposed in [43]. In this approach, two images are represented by piecewise linear curves and a hierarchical approach is then used to find the corresponding feature transformation. J.Y. Kang and B.S. Lee used morphing to hull form generation [16]. Gotsman and Surazhsky [13] extended morphing to polygons such that the intermediate polygons are simple and intersection free. Image morphing is used in [7] to generate intermediate image between top and bottom slices of coronal loops. A morphing technique in which objects are represented by contours has been described in [35]. In this technique, intermediate frames are generated by using physics based formulation. Image morphing using elastic body spline developed in [1]. There are several other morphing techniques, for example energy minimization [25], multilevel free form deformation [21] and morphing based on optimal mass preserving mapping [15, 23, 44].

View morphing is an extension of image morphing, in which a view is constructed from two different views. It was introduced in [34] which correctly handles 3D projective camera and scene transformations. Manning and Dyer [26] extended this approach to dynamic scenes. Xiao and Shah [40] further extended their work to generate video from three uncalibrated images without using 3D model. Tri-view morphing [41] is a recent approach used to create 3D scenes using multiple image morphing. It used trifocal tensor to generate the warping transforms among three views. Often the objects to morph have different pose or view, simple morphing techniques, in such cases, do not create pleasing effects. Seitz and Dyer [32, 33] showed that from two different viewpoints of an object, any view can be created using morphing along the line connecting these viewpoints.

Most of the existing techniques have been implemented in spatial domain. In this work, we propose a morphing technique based on frequency domain. Our work is motivated by the concept of Hybrid Images [27]. In hybrid image technique, by mixing high and low frequencies of two different images, a third image is generated which may produce different perceptions, as the viewing distance changes. The hybrid image technique is not an image morphing technique. In our work, we have extended the same idea for the application of image morphing. To the best of our knowledge no such morphing algorithm has been proposed before us. Our complete algorithm is discussed in the following section.

## 3 Image Morphing in Frequency Domain

Numerous spatial domain image morphing techniques were discussed in the previous section. In this section, we describe our proposed technique which we have named as *Image Morphing in Frequency Domain* (IMFD). Our proposed technique is based on two steps: image alignment and transition control. Each of these steps is explained in the following subsections:

### 3.1 Image Alignment

In order to minimize ghost-effect, the target image should be geometrically aligned with the source image. For this purpose, in our proposed algorithm, the animator has to manually specify the perceptually similar feature correspondences in the form of points or lines. For images with little fine details, we observe, features may be specified as points whereas in case of non-regular, complex objects line segments produce better results. In the following subsections, alignment techniques used in our proposed morphing algorithm are described.

### *3.1.1 Point Based Feature Alignment*

In point based feature alignment, the animator selects perceptually similar feature point correspondences in the source

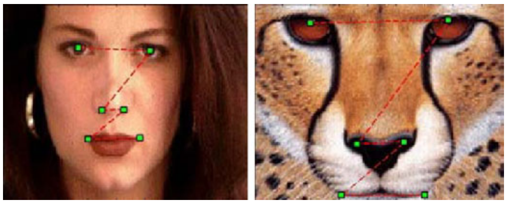

**Fig. 2** Six perceptually similar point correspondence are manually marked in the source image (*left*) and the target image (*right*). The *dotted line* shows the order of clicking the feature points

and the target images. The minimum number of correspondences depends upon the parameters in the assumed geometric model between the source and the target images. For an affine transformation model, at least three correspondences are must and for the projective model at least four correspondences are required. Specifying larger number of correspondences, than the minimum required, causes a reduction in the effect of the point click error by using the least error squared model fitting approach. Figure 2 shows six feature point correspondences specified in the source and the target images.

Once the feature correspondences are specified, the geometric model that defines the spatial relationship between the source and the target images may be computed. We empirically observed that the projective transformation model can sufficiently align the target image with the source image. Each correspondence has two points: $\{(x, y), (x', y')\}$, the point $(x, y)$ is in the source image and the point $(x', y')$ is in the target image. For one correspondence, the projective transformation is given by:

$$x' = \frac{a_1x + a_2y + b_1}{c_1x + c_2y + 1}, \tag{1}$$

$$y' = \frac{a_3x + a_4y + b_2}{c_1x + c_2y + 1}, \tag{2}$$

where $a_1, a_2, a_3, a_4, b_1, b_2$ and $c_1, c_2$ are projective parameters. These equations may also be written in the matrix form:

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_3 \\ a_4 \\ b_2 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}. \tag{3}$$

Since there are eight projective parameters and one correspondence yields two equations, we need at least four correspondences to solve for the eight parameters:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_3 \\ a_4 \\ b_2 \\ c_1 \\ c_2 \end{bmatrix}$$

$$= \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix}. \tag{4}$$

Equation (4) may be written as:

$$XP = Y, \tag{5}$$

where $X$ is $(8 \times 8)$ matrix, $P$ is $(8 \times 1)$ and $Y$ is also $(8 \times 1)$. $P$ is the geometric model that defines the spatial relationship between the source and target images. For 4 correspondences, $X$ will be non-singular if these 4 points do not lie on a straight line. In this case:

$$P = X^{-1}Y. \tag{6}$$

If the number of features points, $M$, are greater than 4, $X$ becomes $(2M \times 8)$ and $Y$ becomes $(2M \times 1)$. As $X^{-1}$ does not exist for non square matrices, the system of linear equations may be solved using the pseudo inverse technique:

$$P = (X^tX)^{-1}(X^tY), \tag{7}$$

which yields the same results as the least error squared model fitting technique. The transformation $P$ is applied to the source image so that it gets aligned with the target image. Maximum number of features should be used for better alignment. Figure 3 shows the two images shown in Fig. 2 after feature alignment. The selected feature points are aligned.

Using points to specify the feature correspondences between the source and the target image is easy and computationally fast than other alignment techniques. However, we identify three main limitations of the point based alignment technique.

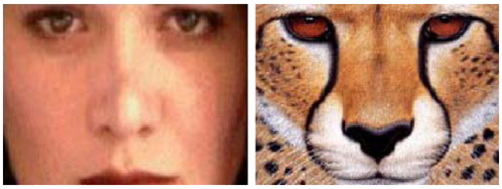

**Fig. 3** Source and target images, shown in Fig. 2, after alignment

1. **Local versus global alignment:** If the two images to be aligned do not follow one global transformation model, local transformation method must be preferred. The line segment based feature alignment technique is a local alignment technique, therefore may be preferred in such cases.
2. **Time consuming:** The quality of alignment is directly affected by the number of feature points. To have a better alignment, the animator may have to specify a large number of points which is quite tedious and time consuming process.
3. **Distorted image:** Small inaccuracy while marking feature points may result in a distorted image instead of an aligned image.

Therefore, in cases, when the images contain complex $3D$ objects like faces, the global geometric transformation models, like affine and projective, may not properly align the two images in such cases, we may use the line segments to specify the feature correspondences.

### 3.1.2 Line Segment Based Feature Alignment

In line segment based feature alignment, the animator marks lines across a feature in the two images. A line covers a large feature length (that may require a large number of points otherwise) and it is easy to mark a few lines as compared to a large number of feature points. Each correspondence in this case consists of two lines defined by two pixels. Let $AB = \{(x_1, y_1), (x_2, y_2)\}$ be a feature line segment in target image and $A'B' = \{(x'_1, y'_1), (x'_2, y'_2)\}$ is the corresponding feature line in the source image. Let $P = (x, y)$ be a pixel in the target image and $P' = (x', y')$ be the new coordinates of pixel $P$ in the source image is computed by the approach outlined in [17]. Let $u$ be the position of the point $P$ along the line $AB$ (Fig. 4) and may be computed as:

$$u = \frac{(P - A) \cdot (B - A)}{|B - A|^2}, \tag{8}$$

where $\cdot$ is dot product of vectors $(P - A)$ and $(B - A)$, and $|B - A|$ is the magnitude of vector $B - A$. As $P$ moves from $A$ to $B$, the value of $u$ changes from 0 to 1. The value of $u$ is less than 0 or greater than 1 if point $P$ is outside the line $AB$, that is:

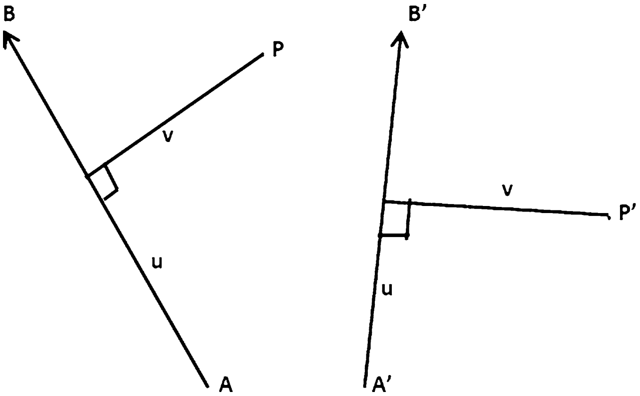

**Fig. 4** Points $A$, $A$ and $A'$, $B'$ are marked by the animator. $P$ is any pixel in target image and $P'$ is the new position of $P$ in the source image

$$u = \begin{cases} [0, 1] & \text{if } A \le P \le B, \\ < 0 & \text{if } A > P, \\ > 1 & \text{if } B < P, \end{cases}$$

$v$ is the perpendicular distance in pixels of $P$ from line $AB$

$$v = \frac{(P - A) \cdot (B - A)^{\perp}}{|B - A|} \tag{9}$$

$(B - A)^{\perp}$ is perpendicular vector to vector $(B - A)$ and may be computed as:

$$(B - A)^{\perp} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (B - A).$$

The new position $P'$ of point $P$ is given by $A'$ and the transformed distance of $P$ computed using the values of $u$ and $v$

$$P' = A' + u(B' - A') + \frac{v(B' - A')^{\perp}}{|B' - A'|}. \tag{10}$$

Generally multiple lines are marked in the source and the target images. In that case, the position $P'$ os computed by considering the effect of all the line pairs. The reader must see [17] for complete algorithm. The next section describes the transition control procedure to generate the intermediate images between the source and target images.

## 3.2 Transition Control

In transition control, a morph sequence is generated which is a set of intermediate frames between the source and the target images. When intermediate frames are viewed in sequence the source image transforms into the target image seamlessly. Let $s$ and $t$ be the aligned source and the target images each of size $m \times n$ pixels that are to be transformed from spatial domain into the frequency domain. The most commonly used transforms include Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT). We have experimented with both DFT and DCT, however, we prefer to use DCT due to the Following reasons:

1. Space complexity of DCT is less than DFT. It is because of the fact that DCT coefficients are real whereas the DFT coefficients are complex numbers having real as well as imaginary components.
2. In DCT, the transition function may consist of a set of low pass and high pass filters with cutoff frequencies varying with uniform step size. In contrast, in case of DFT, in order to produce seamless transition, the step size must be non uniform, which complicates the transition function. It is because of the fact that different frequency bands in DFT have more variation in perceptional significance as compared to DCT coefficients.

Let $S$ and $T$ be the frequency domain representations of the source and the target images and let $(u, v)$ be the indices in the frequency domain: $0 \leq u < m$, $0 \leq v < n$ and $S = \mathrm{DCT}(s)$ and $T = \mathrm{DCT}(t)$.

In frequency domain, the frequencies near $(u, v) = (0, 0)$ are low frequencies and as we move away from $(0, 0)$, we get high frequencies. The highest frequencies are around the other corners of the images $(m-1, 0)$, $(0, n-1)$ and $(m-1, n-1)$. We observe that the low frequencies correspond to the structure or big details in the image while the high frequencies add the fine details in the image. Frequency for $(u, v) = (0, 0)$ is DC-component, which is the average of all the image pixels. The size of the image when represented in frequency domain is same as in spatial domain. The distance of any point from the origin $(0, 0)$ may be computed using Euclidean formula. The distance of a point at $(u, v)$ from the origin $(u, v) = (0, 0)$ is:

$$D(u, v) = \sqrt{u^2 + v^2}. \tag{11}$$

The maximum distance is at corner $(m-1, n-1)$ and is $\sqrt{(m-1)^2 + (n-1)^2}$.

To extract a particular band of frequencies we use different types of low pass and high pass filters. The filter that cuts-off all the high frequencies from the transform is called a low pass filter and the filter that cuts-off all the low frequencies is a high pass filter [12]. Commonly used filters include Ideal, Butterworth and Gaussian filters to extract certain frequencies from an image. We use Gaussian filters in our algorithm because it does not produce the ringing effect in the filtered images. In frequency domain, Gaussian low pass filter may be written as:

$$H_{lp}(u, v) = e^{-(D^2(u,v))/(2D_o^2))}, \tag{12}$$

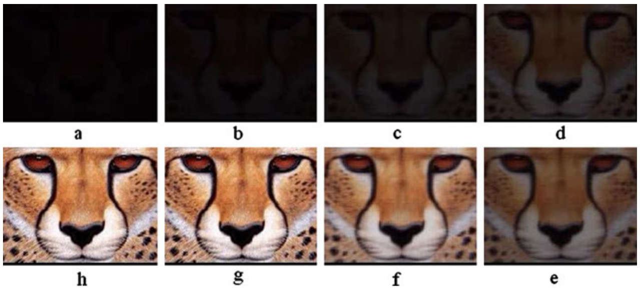

**Fig. 5** Eight low pass filtered images of the target shown in Fig. 3 with different cut-off frequencies, mentioned in Table 1

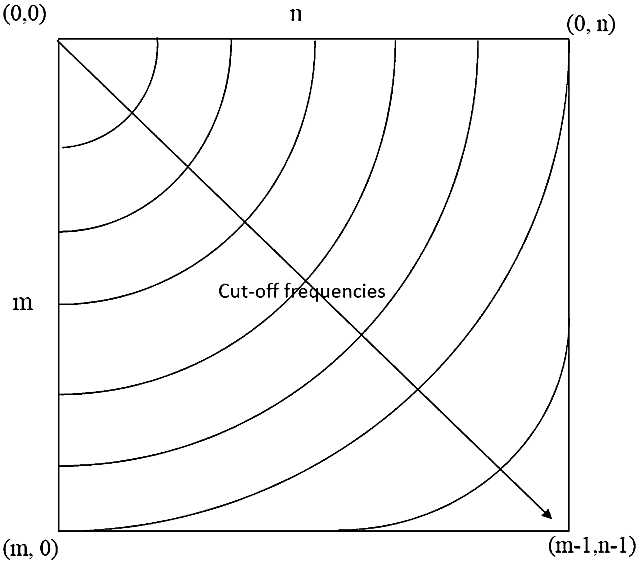

**Fig. 6** For image size $m \times n$, the cut off frequencies vary from $(0, 0)$ to $(m-1, n-1)$ in equal steps for DCT transform

where $D(u, v)$ is given by (11) and $D_0$ is the cut off frequency. For an image of size $m \times n$ pixels, the value of cut-off frequency $D_0$ ranges from $D(0, 0) = 0$ to $D(m-1, n-1) = \sqrt{(m-1)^2 + (n-1)^2}$, see Fig. 6. Figure 5 shows eight low pass filtered images of the target image shown in Fig. 3, obtained by applying Gaussian low pass filter with different *cut-off frequencies* listed in Table 1, denoted by $D_0(u, v)$. Since Gaussian filter is not defined for 0 variance, the first cut-off frequency $D_0 = 0.001$.

Any intermediate frame is obtained by blending the low frequencies of the source image $S$ with the high frequencies of the target image $T$. We apply Gaussian low pass filters with gradually decreasing cut-off frequencies to filter the high frequencies from $S$, and Gaussian high pass filters to remove the low frequencies from the $T$. The blend of the corresponding low pass filtered source images with high pass filtered target image produces the morph sequence. In the transition control function, $D_o$ is varied from the maximum value to the minimum value in equal steps. If there are $N$ intermediate frames to be generated between the source

**Table 1** Details of cut-off frequencies $D_0$ for each image in Fig. 5

| Image label | Cut-off frequency $D_o$ |
|---|---|
| a | ≈0 |
| b | 30 |
| c | 60 |
| d | 90 |
| e | 120 |
| f | 150 |
| g | 180 |
| h | 210 |

and the target images, the total range of $D_o$ is divided into $N$ equal steps with step size:

$$Step\ Size = \frac{\sqrt{(m-1)^2 + (n-1)^2}}{N}. \tag{13}$$

For the $i$th intermediate frame, the cut-off frequency is given by:

$$D_o^i = \frac{i}{N}\sqrt{(m-1)^2 + (n-1)^2}, \tag{14}$$

where for $i = 1, 2, \dots, N$. For each intermediate frame, $S$ is multiplied by the $i$th Gaussian low pass filter $H_{lp}^i(u, v)$

$$H_{lp}^i(u, v) = e^{-(D^2(u,v))/2{D_o^i}^2)} \tag{15}$$

and $T$ is multiplied by the $i$th Gaussian high pass filter $H_{hp}^i(u, v)$:

$$H_{hp}^i(u, v) = 1 - H_{lp}^i(u, v). \tag{16}$$

Figure 7 shows eight high pass filtered images of the target. This is obtained by applying Gaussian high pass filters with the same cut-off frequencies as used in Fig. 5.

Both filtered spectrum are then added up:

$$F_i(u, v) = S(u, v)H_{lp}^i(u, v) + T(u, v)H_{hp}^i(u, v), \tag{17}$$

where $F_i(u, v)$ is the frequency domain representation of the $i$th intermediate frame. Equation (17) can be simplified by substituting the value of $H_{hp}(u, v)$ from (16):

$$F_i(u, v) = (S(u, v) - T(u, v))H_{lp}^i(u, v) + T(u, v). \tag{18}$$

For all intermediate frames, $(S(u, v) - T(u, v))$ remains same therefore it may be computed only once to reduce the computational cost. An inverse DCT of the $F_i(u, v)$ will yield the intermediate frame in spatial domain:

$$f_i(x, y) = \text{iDCT}(F_i(u, v)). \tag{19}$$

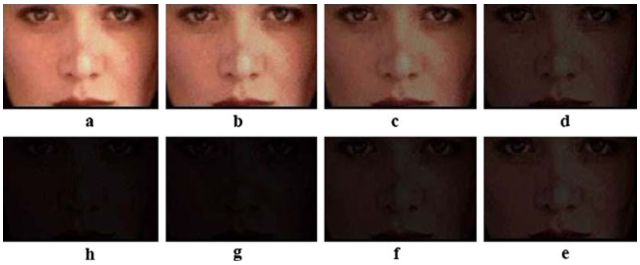

**Fig. 7** Eight high pass filtered images of the source shown in Fig. 3 with different cut-off frequencies, mentioned in Table 1

The morph sequence can now be generated between the two images shown in Fig. 3 by adding the low pass filtered images of the source image to its corresponding high pass filtered images of the target. Figure 10 shows the morph sequence generated as the result. Algorithm 1 for generating $N$ number of intermediate frames is summarized in the following listing, and Fig. 9 is the block diagram of the same algorithm. In case where images are colored, the DCT is applied to each color component.

**Algorithm 1** IMFD

```
$S(u, v) \leftarrow \text{DCT}(s(x, y))$
$T(u, v) \leftarrow \text{DCT}(t(x, y))$
for $i = 1$ to $N$ do
    $D_o^i \leftarrow \frac{i}{N}\sqrt{(m/2)^2 + (n/2)^2}$
    $H_{lp}(u, v) \leftarrow$ Gaussian low pass filter with radius $D_o^i$
    $F_i(u, v) \leftarrow (S(u, v) - T(u, v))H_{lp}^i(u, v) + T(u, v)$
    $f_i \leftarrow \text{iDCT}(F_i(u, v))$ {iDCT is inverse of DCT, $f_i$ is the $i$th intermediate frame}
    $SAVEframe\ f_i$
end for
```

### 3.3 Choice of Features Versus Quality of the Morphed Sequence

Quality of the morphed sequence depends on two main factors. The quality of the selected features for alignment and the number of features selected. For feature selection and alignment, either point based or line segment based alignment techniques may be used depending on the complexity of the images. Each approach has its own advantages and disadvantages as already discussed in Sect. 3.1.

#### *3.3.1 Effect of the Quality of Selected Features on the Quality of Morphed Sequence*

Selection of good quality features is very important for any morphing algorithm. The animator should carefully choose the feature points because a bad choice may lead to a badly morphed image sequence. Human vision is more sensitive to

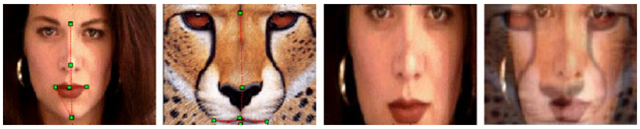

**Fig. 8** From *left* to *right*: *First image* and *second image* is the source and target images respectively. The *third image* is the source image after alignment and last image is an intermediate image. The ghost effect is visible around eyes, nose and lips

the structure of any object than its color. We propose that the main features of the structure in an image should be selected as points of correspondence. For example in any facial image eyes, nose, ears, eyebrose, chin and lips are the structural features and hence are good candidates. We have repeated the same experiment but with different features as shown in Fig. 8 where ghost effect especially around eyes, nose and lips area is clearly visible. This shows that the choice of the selected features is very important to the quality of the morphed sequence.

### *3.3.2 Effect of Number of Features on the Quality of the Morphed Sequence*

Quality of morphed sequence also depends on the number of features selected. Greater the number of features, better the alignment and hence better the quality of the morphed sequence. The number of features required for a good quality morphing depends on the details in the two images and the similarity between the objects to morph. We empirically observed that to morph one human facial image to another requires almost 20 features, but to morph a human face to a horse face will require around 100 features. The next section describes the time complexity and space analysis of the proposed technique.

## 3.4 Execution Time and Space Complexity Analysis of IMFD

Time complexity of generating one intermediate frame between the two images, after alignment, is the time spent on domain transformation, that is converting the image from spatial domain to discrete cosine transform. Discrete cosine transform (DCT) of an image $I$ of size $m \times n$ is calculated as:

$$F(u,v) = \alpha_u \beta_v \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(i,j) \cos\left[\frac{\pi}{m}\left(i+\frac{1}{2}\right)u\right] \times \cos\left[\frac{\pi}{n}\left(j+\frac{1}{2}\right)v\right], \tag{20}$$

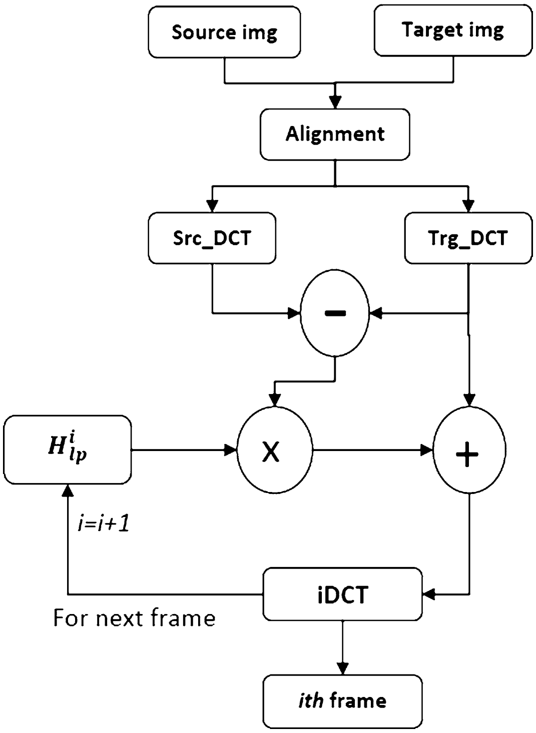

**Fig. 9** IMFD algorithm

where

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{m}} & \text{if } u = 0 \\ \sqrt{\frac{2}{m}} & \text{if } 0 < u < m \end{cases}$$

and

$$\beta_v = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } v = 0 \\ \sqrt{\frac{2}{n}} & \text{if } 0 < v < n. \end{cases}$$

Direct implementation of (20) requires $O(N^2)$ time, where $N = mn$. Discrete cosine transform can be computed through fast Fourier transform *FFT* with $O(N)$ preprocessing and $O(N)$ post-processing steps [5]. For $N$ elements, FFT can be computed in $O(N \log N)$ time. Frigo and Johnson [8] gave an adaptive, parallel algorithm to compute discrete Fourier transform. This implementation is based on SIMD (Single Instruction Multiple Data) architecture. So, the proposed algorithm is optimal in this way. The asymptotic time complexity of the proposed technique is to generate one intermediate frame between source and target images of size $m \times n$ is $O(mn \log(mn))$. Often one dimentional FFT is used to compute 2 dimentional FFT, resulting in $O(mn \log m + mn \log n)$. Hence, time complexity of IMFD to generate $k$ number of intermediate frames is $O(k(mn \log(mn))$. The proposed technique uses $O(mn)$ ex-

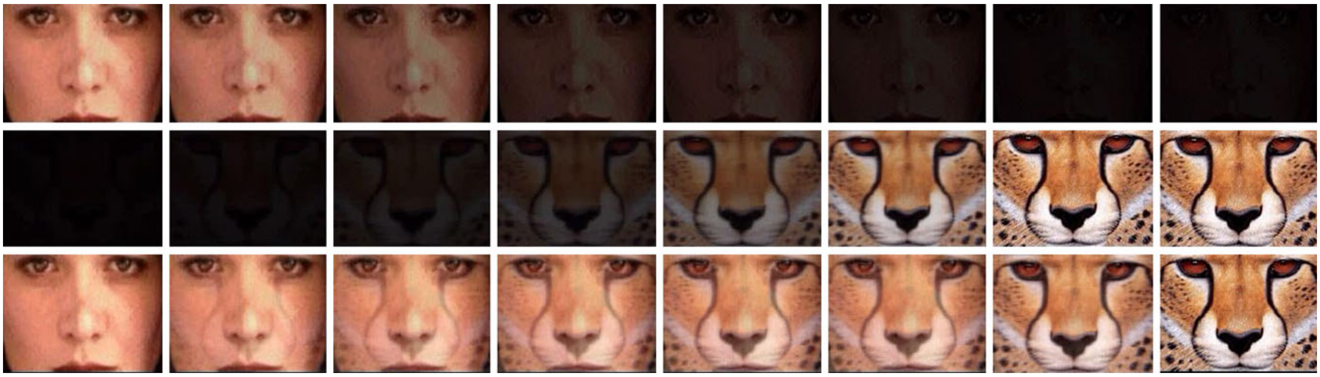

**Fig. 10** Experiment 1: *Top row*: Eight low passed source images with decreasing cut-off frequency from *top* to *bottom*. *Middle row*: Eight high passed target images with the same cut-off frequencies as the corresponding images in the *left column*. *Bottom row*: The resultant eight intermediate frames. See Table 2 for more details of Experiment 1

**Table 2** Experimental details: Image size in pixels, maximum cut-off frequency ($D_o$), step size, number of Intermediate Frames (IF) and the execution time in seconds

| Exp. | Size | max $D_o$ | IF | StepSize | Time |
|---|---|---|---|---|---|
| 1 | 175 × 129 | 216 | 8 | 27 | 0.781 |
| 2 | 477 × 360 | 596.2 | 55 | 10.84 | 28.363 |
| 3 | 258 × 273 | 374.2 | 80 | 4.67 | 19.625 |
| 4 | 252 × 218 | 331.7 | 50 | 6.63 | 7.016 |
| 5 | 380 × 300 | 482.7 | 35 | 13.79 | 11.688 |

tra space to store discrete cosine transform of the source image, target image and intermediate frame of size $m \times n$.

## 4 Experiments and Results

The proposed algorithm has been implemented in Matlab and tested for the quality of morphing in terms of the smoothness of the transformation and execution time on a wide variety of images. The results of five different experiments have been reported in this paper. In each of the experiment, different number correspondences are marked for alignment (either using point based or line segment based techniques). The intermediate frames in each experiment are generated by using a uniform step size in the cut-off frequency of the low pass and high pass filters. The image size, the number of intermediate frames, the maximum value of cut-off frequency, the step size used in each experiment and the total execution time taken by each experiment are reported in Table 2. The execution time reported in each experiment is the total time required to generate the full sequence of intermediate frames, including the file I/O times as well. These execution times are reported upon Dell Optiplex 330 with Intel core 2 duo CPU 2.2 GHz and 1 GB RAM.

In the first experiment as shown in Fig. 10, a facial image of a girl transforms into a lion. The alignment of the two images is done by using point based technique. Six feature points are marked in the source and the target images shown in Fig. 2. In this experiment, 8 intermediate frames are generated in less than 0.781 seconds. In next experiment shown in Fig. 11, a man transforms into a woman. In this experiment 55 intermediate frames are generated in only 38.363 seconds, costing 0.697 seconds per frame. Figure 11 shows only 8 intermediate frames. In this experiment, line segment based alignment is done with 12 feature lines.

Figure 12 shows third experiment,[1] where a boy transforms to another boy. In this experiment, alignment is done using line segment based technique with 8 feature lines and 80 intermediate frames are generated in 19.625 seconds. In fourth experiment shown in Fig. 13, a brown cat transforms into a black cat. Fifty intermediate frames are generated in 7.016 seconds. Fifteen feature lines were marked to align the source and the target images. Figure 13 shows every fifth frame in the morphed sequence. Figure 14 shows the fifth experiment, where alignment is done using line segment based technique and 35 intermediate frames were generated. Figure 14 shows every third frame in the morphed sequence.

### 4.1 Comparison with Mesh Warping and Field Morphing Techniques

The proposed algorithm has also been compared with field morphing and mesh morphing for the execution time speedups and quality of the morphed sequences. The five experiments listed in Table 2 are also carried out using Mesh Morphing and Field Morphing techniques in the same environment and on the same machine using the same tool, Matlab.

[1] Thanks to Mr. Umair and Mr. Hassan for images used in third experiment.

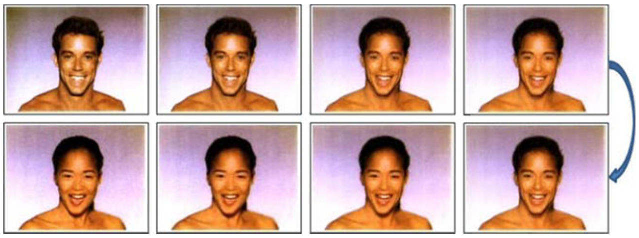

**Fig. 11** Experiment 2: A man transforms into a woman. *Left image* in the *top row* is source image and *left image* in the *bottom row* is target image. The *left* to *right* in the *top row* and *right* to *left* in the *bottom row* are intermediate images

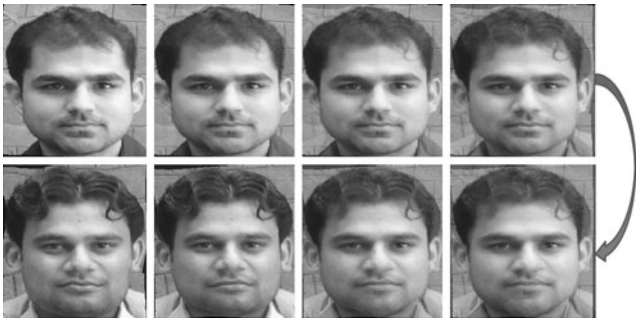

**Fig. 12** Experiment 3: A boy transforms into another boy. *Left image* in the *top row* is source image and *left image* in the *bottom row* is target image. The *left* to *right* in the *top row* and *right* to *left* in the bottom row are intermediate images

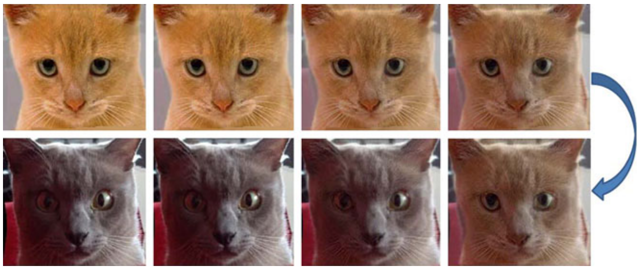

**Fig. 13** Experiment 4: A brown cat transforms into a black cat. *Left image* in the *top row* is source image and *left image* in the *bottom row* is target image. The *left* to *right* in the *top row* and *right* to *left* in the *bottom row* are intermediate images

In Sect. 4.1.1 the quality of the morphed sequences generated by mesh warping technique is compared to that of the proposed technique. Execution time for each experiment is also recorded and presented in Table 3. Section 4.1.2 describes quality comparison of Field Morphing technique with the proposed technique. Execution time in each experiment is reported in Table 4. Section 4.1.3 describes time comparison of the proposed technique with mesh warping and field morphing.

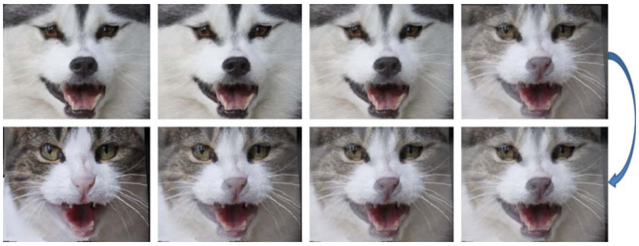

**Fig. 14** Experiment 5: A dog transforms into a cat. *Left image* in the *top row* is source image and *left image* in the *bottom row* is target image. The *left* to *right* in the *top row* and *right* to *left* in the *bottom row* are intermediate images

**Table 3** Execution time of the five experiment listed in Table 2 using Mesh Warping technique with the same number of intermediate frames. Time is in seconds

| Exp. | No. of IF | Execution time |
|---|---|---|
| 1 | 8 | 2.593 |
| 2 | 55 | 35.890 |
| 3 | 80 | 29.938 |
| 4 | 50 | 19.922 |
| 5 | 35 | 21.827 |

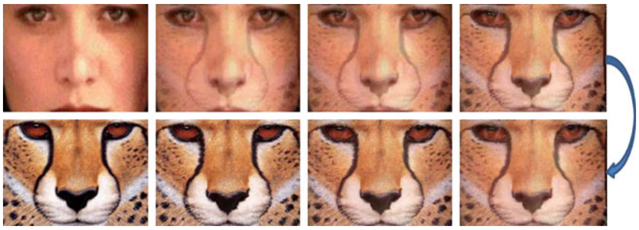

**Fig. 15** Experiment 1 (with mesh warping): Woman transforms into a tiger

### 4.1.1 *Comparison with Mesh Warping*

Figures 15, 16, 17, 18 and 19 show the same five experiment carried out using mesh warping technique. You might observe one problem in the morphed sequence generated in these experiments and that is the edge regions are blurred and give unnatural look. This is because the pixels are moved in blocks or meshes in this technique. This effect is evident in the region around the shoulders in Fig. 16 and in the background transition in Fig. 17 especially in second and third images in the top row and fourth image in the bottom row. This problem is very visible in the right eye in the transformation of brown cat to black cat in Fig. 18. A blur wave is visible in Fig. 19 at the bottom of intermediate images. From all these experiment, it is evident that the quality of proposed technique is better than mesh warping. Table 3 reports the execution time of each experiment with number of frames.

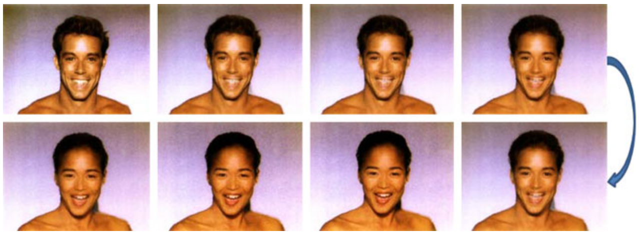

**Fig. 16** Experiment 2 (with mesh warping): A man transform into a woman

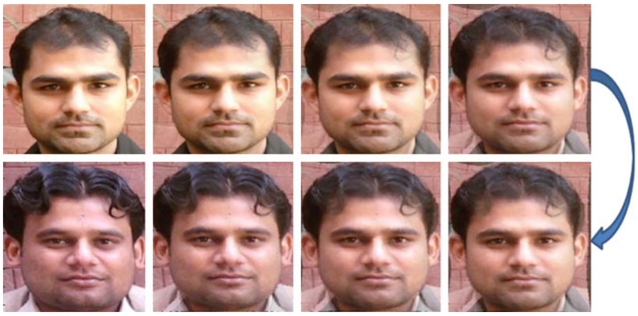

**Fig. 17** Experiment 3 (with mesh warping): One facial image transforms into another

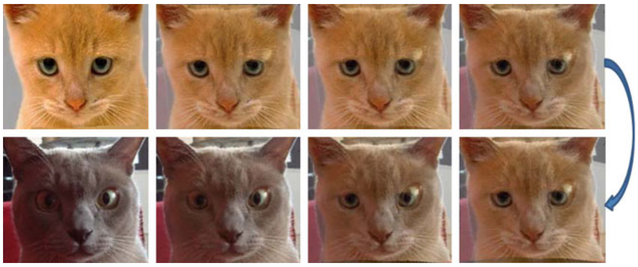

**Fig. 18** Experiment 4 (with mesh warping): A brown cat transforms to a black cat

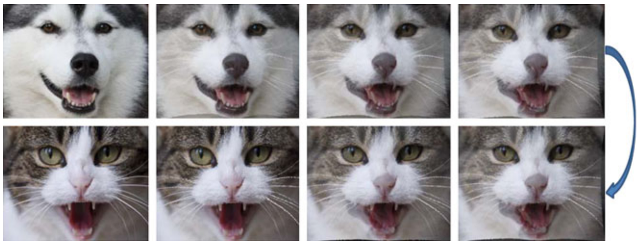

**Fig. 19** Experiment 5 (with mesh warping): A dog transforms into a cat

### *4.1.2 Comparison with Field Morphing*

Figures 20, 21, 22, 23 and 24 show the five experiment carried out using field morphing technique. There are two major problems in field morphing technique. First it is very slow as each pixel in every intermediate frame is computed with respect to all the features lines. The second is related to the quality of the morphed sequence. As each pixels is interpolated with respect to feature lines, sometimes unexpected interpolation errors occur and object is distorted in intermediate frame. This problem is visible in Fig. 21. See the region of right cheek and the chin (especially in middle intermediate images), the right cheek and chin is distorted (compare this with Fig. 11 where the same experiment is done with the proposed technique, there is no such distortion in that sequence). Experiment 3 in Fig. 22 suffers with this problem as well. Note the lips in the morphed sequence are distorted (clearly visible in middle intermediate images in this morphed sequence). Compare this sequence with the sequence generated by the proposed technique in Fig. 12, which is free from distortion of lips and region around lips. The Execution time of each experiment is reported in Table 4.

**Table 4** Execution time of the five experiment listed in Table 2 using Field Morphing techniques. Time is in seconds

| Exp. | No. of IF | Execution Time |
|---|---|---|
| 1 | 8 | 3.312 |
| 2 | 55 | 68.180 |
| 3 | 80 | 60.612 |
| 4 | 50 | 30.493 |
| 5 | 35 | 37.153 |

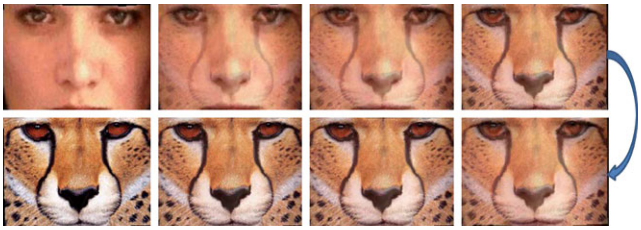

**Fig. 20** Experiment 1 (with field morphing): Woman transforms into a tiger

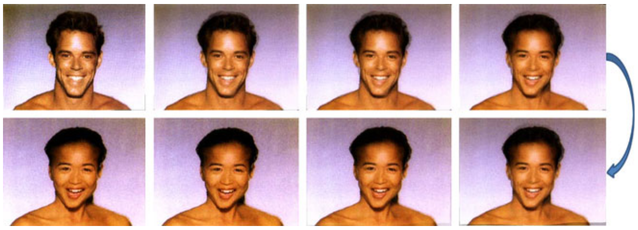

**Fig. 21** Experiment 2 (with field morphing): A man transform into a woman

### *4.1.3 Time Speedup over Mesh Warping and Field Morphing*

Time spent to generate same number of intermediate frames for each experiment using the proposed technique, Mesh

**Table 5** Execution time comparison of the proposed technique with Mesh Warping and Field Morphing techniques. Time is in seconds and MW stands for Mesh Warping and FM stands for Field Morphing

| Exp. | No. of IF | IMFD | MW | Speedup over MW | FM | Speedup over FM |
|---|---|---|---|---|---|---|
| 1 | 8 | 0.781 | 2.593 | 3.32 | 3.312 | 4.24 |
| 2 | 55 | 28.363 | 35.890 | 1.26 | 68.180 | 2.41 |
| 3 | 80 | 19.625 | 29.938 | 1.52 | 60.612 | 3.09 |
| 4 | 50 | 7.016 | 19.922 | 2.82 | 30.493 | 4.34 |
| 5 | 35 | 11.688 | 21.827 | 1.86 | 37.153 | 3.17 |

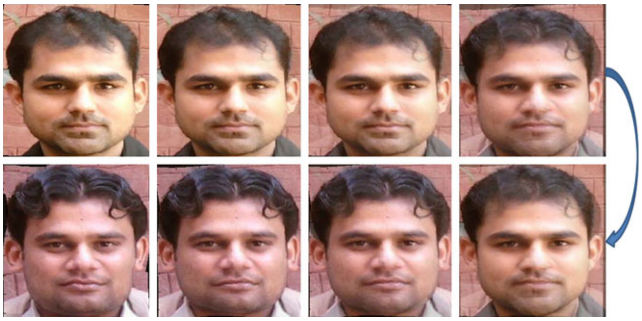

**Fig. 22** Experiment 3 (with field morphing): One facial image transforms into another

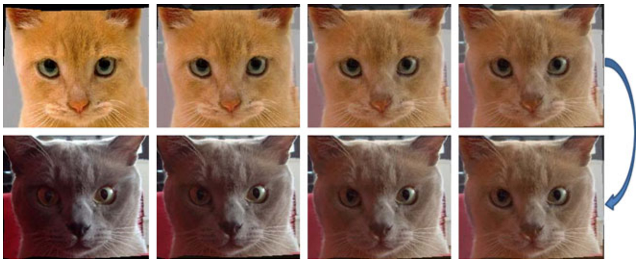

**Fig. 23** Experiment 4 (with field morphing): A brown cat transforms to a black cat

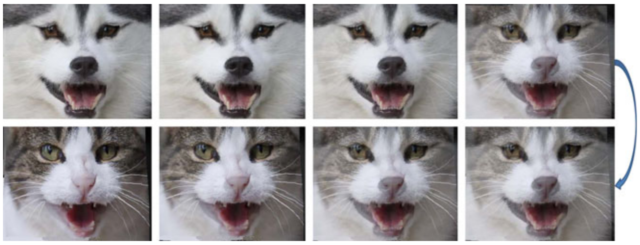

**Fig. 24** Experiment 5 (with field morphing): A dog transforms into a cat

Warping and Field Morphing techniques was reported in Tables 2, 3 and 4 respectively. Table 5 summarizes the whole experiment section. The execution time speedup of our algorithm is on the average 2.16 over mesh warping and on the average 3.45 over field morphing for same number of intermediate frames. We observe that the proposed technique is much faster than field morphing and significantly faster than mesh warping technique without compromising the quality of the morphed sequence. This is because in both of the techniques all features points are referenced to calculate one pixel in an intermediate frame. That is, their execution time is directly dependent on the number of corresponding features, there is no such dependency in the proposed technique.

## 5 Conclusion

In this paper a new image morphing technique based upon frequency domain representation of images is presented. The proposed technique is significantly faster in terms of execution time than existing spatial domain techniques without compromising the quality. The technique works in two main steps. In the first step, the source and the target images are aligned using either the point based or line based alignment technique. In point based feature alignment approach, global geometric transformation model is computed, while in the line based approach local transformation model of computed at each pixel. We find that in case the image pair is not following one global geometrical transformation model, the local approach produces better results. In the second step, intermediate frames are generated between the source image and the target image by blending low frequencies of source with high frequencies of the target image by using a transition function. The proposed technique is useful when large number of intermediate frames are to be generated or morphing has to be done between images of larger sizes.

**Acknowledgements** We are thankful to Stephen Mullens for Mesh Warping code. We are also thankful to Prof. Dr. Syed Mansoor Sarwar for his support.

We are thankful to the referees and the Editor of J. Math. Imaging Vis for their constructive feedback and comments that resulted in a significant quality improvement in the original manuscript.

## References

1. Aboul-Ella, H., Karam, H., Nakajima, M.: Image metamorphosis transformation of facial images based on elastic body splines. Signal Process. **70**, 129–137 (1998)
2. Arsigny, V., Commowick, O., Ayache, N., Pennec, X.: A fast and log-Euclidean polyaffine framework for locally linear registration. J. Math. Imaging Vis. **33**, 222–238 (2009)

3. Bigot, J., Gadat, S., Loubes, J.M.: Statistical m-estimation and consistency in large deformable models for image warping. J. Math. Imaging Vis. **34**, 270–290 (2009)
4. Che, W., Yang, X., Wang, G.: Skeleton-driven 2d distance field metamorphosis using intrinsic shape parameters. Graph. Models **66**, 261–261 (2004)
5. Chen, W.H., Smith, C., Fralick, S.: A fast computational algorithm for the discrete cosine transform. Commun., IEEE Trans. **25**(9), 1004–1009 (1977)
6. Dykstra, C., Celler, A., Greer, K., Jaszczak, R.: The use of image morphing to improve the detection of tumors in emission imaging. Nucl. Sci., IEEE Trans. **46**(3), 673–679 (1999)
7. Faria, L.N., Mascarenhas, N.D.A., Morón, C.E., Saito, J.H., Rosa, R.R., Sawant, H.S.: A parallel application for 3d reconstruction of coronal loops using image morphing. Image Vis. Comput. **25**(1), 95–102 (2007). SIBGRAPI
8. Frigo, M., Johnson, S.: The design and implementation of fftw3. Proc. IEEE **93**(2), 216–231 (2005)
9. Fuchs, M., Jüttler, B., Scherzer, O., Yang, H.: Shape metrics based on elastic deformations. J. Math. Imaging Vis. **35**, 86–102 (2009)
10. Gong, M., Yang, Y.H.: Layer-based morphing. Graph. Models **63**, 45–59 (2001)
11. González, J., Arévalo, V.: Mesh topological optimization for improving piecewise-linear image registration. J. Math. Imaging Vis. **37**, 166–182 (2010)
12. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall, Upper Saddle River (2006)
13. Gotsman, C., Surazhsky, V.: Guaranteed intersection-free polygon morphing. Comput. Graph. **25**(1), 67–75 (2001)
14. Hagege, R., Francos, J.M.: Parametric estimation of affine transformations: an exact linear solution. J. Math. Imaging Vis. **37**, 1–16 (2010)
15. Johan, H., Koiso, Y., Nishita, T.: Morphing using curves and shape interpolation techniques. In: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications (PG '00), p. 348. (2000)
16. Kang, J.Y., Lee, B.S.: Mesh-based morphing method for rapid hull form generation. Comput. Aided Des. **42**, 970–976 (2010)
17. Karam, H., Hassanien, A., Nakajima, M.: Feature-based image metamorphosis optimization algorithm. In: Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM'01), pp. 553–554 (2001)
18. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
19. Krüger, W.: Robust and efficient map-to-image registration with line segments. Mach. Vis. Appl. **13**, 38–50 (2001)
20. Larrey-Ruiz, J., Verdú-Monedero, R., Morales-Sánchez, J.: A fourier domain framework for variational image registration. J. Math. Imaging Vis. **32**, 57–72 (2008)
21. Lee, S.Y., Chwa, K.Y., Shin, S.Y.: Image metamorphosis using snakes and free-form deformations. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95), pp. 439–448 (1995)
22. Lee, S., Woberg, G., Chwa, K.Y., Shin, S.Y.: Image metamorphosis with scattered feature constraints. IEEE Trans. Vis. Comput. Graph. **2**, 337–354 (1996)
23. Lee, S., Wolberg, G., Shin, S.Y.: Scattered data interpolation with multilevel b-splines. IEEE Trans. Vis. Comput. Graph. **3**, 228–244 (1997)
24. Lee, S., Wolberg, G., Shin, S.Y.: Polymorph: morphing among multiple images. IEEE Comput. Graph. Appl. **18**, 58–71 (1998)
25. Lee, S.-Y., Chwa, K.-Y., Hahn, J., Shin, S.Y.: Image morphing using deformation techniques. Visualization and Computer Animation **7**
26. Manning, R.A., Dyer, C.R.: Dynamic view morphing. In: Proc. SIGGRAPH 96, pp. 21–30 (1996)
27. Oliva, A., Torralba, A., Schyns, P.G.: Hybrid images. In: ACM SIGGRAPH 2006 Papers (SIGGRAPH '06), pp. 527–532 (2006)
28. Park, S.Y., Choi, S.I., Kim, J., Chae, J.: Real-time 3d registration using GPU. Mach. Vis. Appl. 1–14
29. Reyes-Lozano, L., Medioni, G., Bayro-Carrochano, E.: Registration of 2d points using geometric algebra and tensor voting. J. Math. Imaging Vis. **37**, 249–266 (2010)
30. Ruprecht, D., Müller, H.: Image warping with scattered data interpolation. IEEE Comput. Graph. Appl. **15**, 37–43 (1995)
31. Seitz, S.: Bringing photographs to life with view morphing. In: Proc. Imagina 97 Conf, pp. 153–158 (1997)
32. Seitz, S.M., Dyer, C.R.: Physically-valid view synthesis by image interpolation. In: Proc. IEEE Workshop on Representations of Visual Scenes, pp. 18–25 (1995)
33. Seitz, S.M., Dyer, C.R.: Toward image-based scene representation using view morphing. In: Proc. 13th Int. Conf. on Pattern Recognition, pp. 84–89 (1996)
34. Seitz, S.M., Dyer, C.R.: View morphing. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96), pp. 21–30 (1996)
35. Singh, R., Papanikolopoulos, N.P.: Planar shape recognition by shape morphing. Pattern Recognit. **33**(10), 1683–1699 (2000)
36. Wang, W.H., Chen, Y.C.: Image registration by control points pairing using the invariant properties of line segments. Pattern Recognit. Lett. **18**, 269–281 (1997)
37. Whitaker, R.T.: A level-set approach to image blending. IEEE Trans. Image Process. **9**(11), 1849–1861 (2000)
38. Wolberg, G.: Digital Image Warping, 1st edn. IEEE Computer Society Press, Los Alamitos (1994)
39. Wolberg, G.: Image morphing: a survey. Vis. Comput. **14**(8–9), 360–372 (1998)
40. Xiao, J., Shah, M.: From images to video: view morphing of three images. In: VMV, pp. 495–502 (2003)
41. Xiao, J., Shah, M.: Tri-view morphing. Comput. Vis. Image Underst. **96**, 345–366 (2004)
42. Xu, C., Prince, J.L.: Snakes shapes, and gradient vector flow. IEEE Trans. Image Process. **7**(3), 359–369 (1998)
43. Yang, W., Feng, J.: Technical section: 2d shape morphing via automatic feature matching and hierarchical interpolation. Comput. Graph. **33**, 414–423 (2009)
44. Zhu, L., Yang, Y., Haker, S., Tannenbaum, A.: An image morphing technique based on optimal mass preserving mapping. IEEE Trans. Image Process. **16**(6), 1481–1495 (2007). doi:10.1109/TIP.2007.896637

**M. Shahid Farid** obtained his B.Sc. in Computer Science from University of the Punjab, Lahore, Pakistan in 2004. He did M.Sc. and M.Phil. in computer science from Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore, Pakistan in 2006 and 2011 respectively. He is currently a Lecturer at PUCIT (www.pucit.edu.pk). His research interests include image morphing, image inpainting, structure from motion estimation and image enhancement.

**Arif Mahmood** did B.Sc. Engineering from UET Lahore (www.uet.edu.pk), in 1994. He completed MS in Computer Science from Lahore University of Management Sciences (www.lums.edu.pk) in 2003, where he received gold medal for top academic merit. Currently, he is close to the completion of Ph.D. in Computer Science from LUMS School of Science and Engineering, Lahore, Pakistan. His Ph.D. research is focused on enhancing the speed of computational algorithms, in particular for template matching and classification applications. He was the winner of the Best Student Paper Award at International Conference on Machine Vision, 2007. He is Assistant Professor at Punjab University College of information Technology (www.pucit.edu.pk), Lahore, Pakistan.