© 2019 IJSRST | Volume 6 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology
DOI : https://doi.org/10.32628/IJSRST

Secure Data Management Using Face Morphing

Anvita D. Phutane[1], Dr. A. A. Gurjar[2]

[1]M.E Scholar, Department of Electronics & Telecommunication Engineering, Sipna College of Engineering and Technology, Amravati, Maharashtra, India

[2]Professor, Department of Electronics & Telecommunication Engineering, Sipna College of Engineering and Technology, Amravati, Maharashtra, India

ABSTRACT

We proposed an image morphing based method for information embedding. The basic idea is to hide a secrete data into a morphed image which is obtained from the secrete image and another reference image. To make this method practically useful, it is essential to produce natural morphed images. This is a necessary condition conceal the existence of the secret image. To produce natural morphed images, we should select a proper feature point set (FPS) for morphing. This is a monotonous work if we do it manually, because the number of possible FPSs is very large. To solve the problem more efficiently, we adopted proposed interactive algorithm in this study and conducted experiments for generating morphed images. Results show that, if we provide a relatively good initial FPS, proposed interactive algorithm can fine-tune the FPS, and produce more natural images with limited number of iteration.

Keywords : Feature Point Set, Face Morphing, Image Morphing, Video Morphing

I. INTRODUCTION

Recently there has been a great deal of interest in morphing techniques for producing smooth transitions between images. Figure 1 shows the Image Morphing process. These techniques combine 2D interpolations of shape and color to create dramatic special effects. Part of the appeal of morphing is that the images produced can appear strikingly lifelike and visually convincing. Despite being computed by 2D image transformations, effective morphs can suggest a natural transformation between objects in the 3D world. The fact that realistic 3D shape transformations can arise from 2D image morphs is rather surprising, but extremely useful, in that 3D shape modeling can be avoided.

Morphing is probably most noticeably used to produce incredible special effects in the entertainment industry. It is often used in movies such as Terminator and The Abyss, in commercials, and in music videos such as Michael Jackson‟s Black or White. Morphing is also used in the gaming industry to add engaging animation to video games and computer games. However, morphing techniques are not limited only to entertainment purposes. Morphing is a powerful tool that can enhance many multimedia projects such as presentations, education, electronic book illustrations, and computer-based training. We discuss what morphing is, different morphing techniques and examples of morphing software packages available for multimedia developers to use when creating multimedia projects.

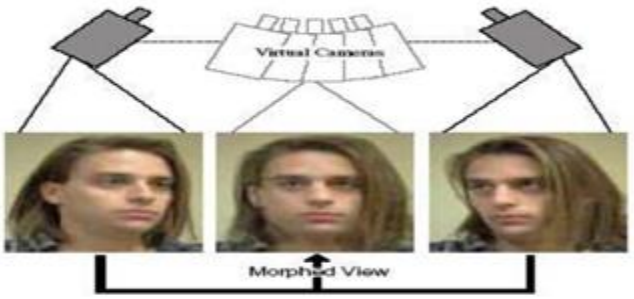

Figure 1. Image Morphing

Significance of Topic

Animation techniques are constantly increasing in quality and creativity. Because consumers demand better quality special effects and effects that will captivate and engage themselves grows profoundly, industries must strive to appease these audiences. Multimedia users and entertainment seekers are no longer satisfied with simple animation, they desire fancier transitions and animation to amaze them and keep them interested in the product. The special effect known as morphing has been one way to satisfy consumers‟ need to be entertained, impressed, and amazed by the final product. Figure 2 shows Facial view morphs. Top: Morph between two views of same person. Bottom: Morph between view of two different people.

Figure 2. Facial View Morphs. Top: Morph between two views of same person. Bottom: Morph between views of two different people.

II. LITERATURE REVIEW

In 2014, Yutaro Minakawa, Mitsuru Abe, Kentaro Sekine, and Qiangfu Zhao. delineated another system for automatic face swapping employing a giant info of faces. although it's laborious for users to search out a candidate face to match the target face in appearance and cause from their pictures, the system allowed de-identification mechanically by choosing candidate face pictures from an outsized face library that's almost like the target face in look and cause. Lastly, it replaces the target candidate with selected candidate from the library image exploitation image primarily based methodology.

In 2014, Yun-Te Lin, Chung-Ming Wang. projected the system that replaces the target subject face within the target video with the supply subject face, beneath similar cause, expression, and illumination. This approach relies on 3D morph-able model and expression model info to upset expressions and therefore the input data of the supply subject face is reduced to at least one to 2 pictures. The 3D face synthesizer derives a Morphable face to suit the input image, and map the feel from the image to the derived 3D face model. A face alignment rule is applied to the target video to find the elaborate options and descriptions of the target subject face. A cause figurer exploits the face alignment results to estimate the top cause parameters of the target subject face. Here methodology employs a 3D countenance info to clone the expressions to the supply face model. to suit the expressions to the target video, Y. T. Cheng et al. projected AN rule to extract the expression parameters. In some videos, directly rendering the supply subject face model onto the target frame ends up in illumination inconsistency. A relighting rule relights the rendered supply subject face for illumination consistency. Finally, it seamlessly composites the rendered supply model with the target frame exploitation Poisson

equation. The output could be a video with the target face replaced by the supply face, with similar cause, expression, and lighting.

In 2015, Seong G. Kong. projected the strategy that permits commutation performances in video. It conjointly provides face replacement in target video from supply video. The system tracks each the faces in supply and target video exploitation multilinear model. exploitation this half-track 3D pure mathematics, supply face is crooked to focus on face in each frame of video. it's generally necessary that the temporal arrangement of the performance matches specifically within the supply and target video; this can be done by retiming rule. when trailing and retiming, it blends the supply performance within the target video to provide the ultimate result. They computed optimum seam through the video volume that maintains temporal consistency within the final composite.

In 2015 Taheer Jamil. projected a replacement face morphing approach that deals expressly with giant cause and expression variations. It recovers the 3D face pure mathematics of the input pictures employing a projection on a pre-learned 3D face topological space. The pure mathematics is interpolated by resolving the expression and cause and ranging them swimmingly across the sequence. Finally, it poses the morphing downside as a repetitious optimization with AN objective that mixes similarity of every frame to the geometry induced crooked sources, with a similarity between neighboring frames for temporal coherence. during this system, it fits a 3D form to each the input pictures. A 3D form contains 2 sets of parameters: external parameters describing the 3D cause of the face, and intrinsic parameters describing the pure mathematics of the person beneath the impact of expression. Then, it linearly interpolates each the intrinsic and external parameters of the 2 input faces, and generates a series of interpolated 3D face models. In every frame, the crooked faces square measure amalgamated along. bound strategies conjointly allowed for automatic face replacement of individuals in single image.

III. PROPOSED WORK

In this system, our aim is to enhance the performance of Image morphing and Virtual information Embedding for pictures. drawback occurred in style of dynamic environments, it's a powerful ability, however it's typically tough to get complete define of Secrete information, accountable to look the empty development, as a result the detection of secrete information isn't very easy. Therefore ought to improve the technique.

To achieve this, the subsequent specific objectives

1. To propose morphing techniques that to effectively generate intermediate morph pictures.
2. To maintain in reliableness between carrier image size & secret information.
3. To produce unbreakable wall for stegnalyzer whereas extracting the key information.
4. To maintain Image sensory activity quality. it's necessary that to avoid suspicion the embedding ought to occur while not important degradation or loss of sensory activity quality of the duvet media.

To give security to hidden message from unauthorized accesses.

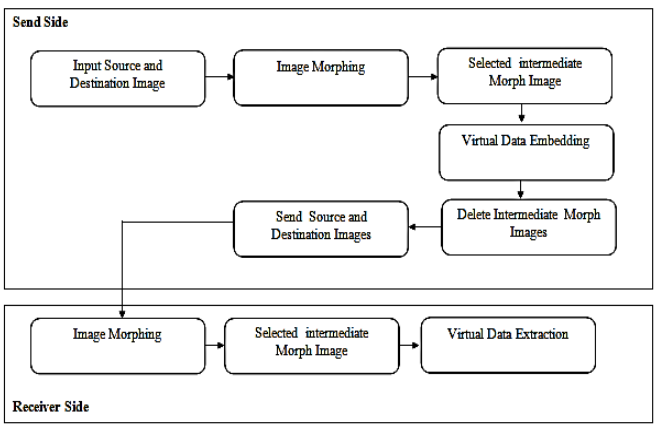

Figure 3. block diagram

Description

We proposed an image morphing based method for information embedding. Figure 3. Shows the architecture of proposed method. The basic idea is to hide a secrete data into a morphed image which is obtained from the secrete image and another reference image. To make this method practically useful, it is necessary to produce natural morphed images.

Sender Side

Input Source and Destination Image

Morphing is the method to perform some operation on an image. In this method, we have to select an input image and an output image. There will be number of intermediate images in between input and an output image. Number of intermediate images depends upon the sender.

Image Morphing

Sleek animated transition from one image into another is known as image morphing. After selecting an input and the destination image, number of intermediate images gets generated in between input and an output image. In morphing process, input image gets converted into destination image, by means of intermediate images. This process is referred to as image morphing.

Selected Intermediate Morphed Images

Our main aim is to hide the data behind one of the intermediate image. For this, one of the intermediate images gets selected, which were generated by the sender in between input and destination image during the process of image morphing. Suppose 'x' number of intermediate morphed image has been selected, then that 'x' image will act as a carrier image for data hiding.

Virtual Data Embedding

In this process, we are hiding the data virtually. For this virtual data embedding, we have to select one of the morphed image which will act as a carrier image. Then there will be selection of the master pixel component of the carrier image. After the extraction of RGB channels, we need to select master pixel component. After this extraction, we need to select the secrete text. Now there will be the process in which secrete data gets converted into the binary form. After this process, there will be virtual data hiding. In this process, position number is mentioned for each bit. Due to this position number to each bit, a random key gets generated of that secrete text. That key gets compressed by the technique known as key compression. For that, we need to select the key length. In this way, data gets hide virtually.

Delete Intermediate Morph Image

After hiding the data virtually, we need to delete all the intermediate morphed images. This process is necessary, as by deleting the intermediate morphed images, malicious third party will not be able to extract the secrete data, and our data remains safe. The next step is to send source and the destination images to the receiver side.

Receiver Side

Image Morphing

At the receiver side, we have to select an input image and a destination image. These input and a destination image should be same as that of the sender side. Again at the receiver side, input image gets converted into the destination image by means of the morphing process.

Select Intermediate Morph Image

For the extraction of the secrete data, it is necessary to select the morph image from one of the intermediate images, which were generated by the sender in between input and the destination image during the process of morphing. One important condition at the receiver side is that, number of intermediate images generated must be same as that of the number of intermediate images generated at the sender side. And the intermediate morph image

which we are selecting for extracting the secrete data must have to be same with the one selected at the sender side for hiding the data.

Virtual Data Extraction

For the extraction of virtual data, the compressed key which has been transferred to the receiver side has to be decompressed. The actual key gets generated due to decompression. For decompression, we need the master table and the compressed key. Again the process of image morphing has to be done. For the virtual data extraction, we need the master component. That master component gets converted into the random decompressed key. Then we need to convert that decompressed key into the binary form. And the binary form of that key gets converted into the secrete data. In this way, the secrete text gets extracted at the receiver side.

IV. PROPOSED ALGORITHM

Morphing

1. Start
2. Read Source (S_i) & Destination(D_i) Video Frames
3. For i=0 to H(Si) * W(Si)
   Extract RGB of Si & Di
   Morph $P_i(D_i)$ into $P_i(s_i)$
   Save Intermediate Image (ID_i)
   End
4. Stop

Algorithm: Data Hiding

1. Start.
2. Input Frame.
3. Extract RGB Channels.
4. Choose Master Pixel (MP).
5. Input Secrete Data.
6. Convert Secrete data to Binary.
7. For i=1:length(Secrete_Binary)
   If MP (i) ==Secrete_Binary (i)
   Add I to Key File
   End
   End
8. Save Key File
9. Stop

Algorithm: Key Compress

1. Start.
2. Input Key_File.
3. Set int c= 65, n=Length of substring etc.
4. For i=1:length(Key_File)
   For j=1: n
   String sub = substring (Key_File, j, n)
   If (Frequency_Occurance (sub)>=2)
   Replace sub with (char) c from Key_File
   Add (char) c, sub to master table
   c= c+1;
   End
   N = n-1;
5. Save Compress Key_file.
6. Stop.

Algorithm: Key decompression

1. Start.
2. Load compressed Key_File & Master_Table.
3. For i=1:RowCount(Master_Table)
   Replace (char) c with sub of Master_Table with Key_File.
   End
4. Save Decompressed Key.
5. Stop.

Algorithm: Data Extraction

1. Start
2. Input Frame.
3. Extract RGB Channel from Frame.
4. Choose Master_Pixel component.
5. Load decompressed Key_File.
6. For i=1:length(Key_File)
   Read bit at position
   I from master_Pixel & add to extracted data.
   End

7. Convert Extracted Data to ASCII character format.
8. Save ASCII character format data.
9. Stop.

V. RESULT

Analysis with respect to PSNR value, NC, MSE, AD value, MD, NAE, SC and Time Constraint

Peak signal-to-noise ratio (PSNR)

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs. The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not be. One has to be extremely careful with the range of validity of this image; it is only conclusively valid when it is used to compare results from the same codec (or codec type)and same content PSNR can be calculates,

PSNR=10 log 255^2

Mean Square Error (MSE)

It measures the average of the square of the error. The error is the amount by which the pixel value of original image differs from the pixel value of decrypted image.

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j) - y(i,j))^2}{MN}$$

Where x (i, j) represents the original image, y (i, j) is the decrypted image and (i, j) represent the pixel positions of the MN image. Here, M and N are the height and width of image respectively.

Normalized Correlation (NC)

It measures the similarity representation between the original image and decrypted image.

$$NC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j) * y(i,j))}{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j))^2}$$

Average Difference (AD)

Average Difference is measurement of differences between two images. Here we calculated the average difference by the formula given. As we know that large value of maximum difference means that image is poor in quality.

$$AD = \frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N}(x(i,j) - y(i,j))$$

Maximum Difference (MD)

Difference between any two pixels such that the larger pixel appears after the smallest pixel. As we know that large value of maximum difference means that image is poor in quality. MD is defined as.

$$MD = MAX\ |x\ (i\ ,\ j) - y\ (I\ ,\ j)|$$

Normalized Absolute Error (NAE)

The large value of normalized absolute error means that image is poor quality. NAE is defined as

$$NAE = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}|x(m,n)-y(m,n)|}{\sum_{m=1}^{M}\sum_{n=1}^{N}|x(m,n)|}$$

Structural Content (SC)

The structural content measure used to compare two images in a number of small image patches the images have in common. The patches to be compared are chosen using 2D continuous wavelet which acts as a low level corner detector. As we know that large value of structural content SC means that image is poor quality. SC is defined as,

$$SC = \frac{\sum_{i=1}^{M}\sum_{j-1}^{N}(y(i,j))^2}{\sum_{i=1}^{M}\sum_{j-1}^{N}(x(i,j))^2}$$

Analysis with respect to Entropy, standard deviation and mean intensity

Entropy

Information entropy is occasionally called Shannon's entropy in honor of Claude E. Shannon, who formulated many of the key ideas of information theory. Shannon defines entropy in terms of a discrete random variable X, with possible states (or outcomes) x1...xn as

$$H(X) = \sum_{i=1}^{n} p(x_i)\log_2\left(\frac{1}{p(x_1)}\right)$$

$$= -\sum_{i=1}^{n} p(x_i)\log_2 p(x_1)$$

Where is the probability of the ith outcome of X. That is, the entropy of the variable X is the sum, over all possible outcomes xi of X, of the product of the probability of outcome xi times the log of the inverse of the probability of xi (which is also called xi surprisal - the entropy of X is the expected value of its outcome's surprisal). We can also apply this to a general probability distribution, rather than a discrete-valued event. Shannon shows that any definition of entropy satisfying his assumptions.

$$-K\sum_{i=1}^{n} p(x_i)\log p(x_i)$$

Where K is a constant (and is really just a choice of measurement units). Shannon's definition of entropy, when applied to an information source, can determine the minimum channel capacity required to reliably transmit the source as encoded binary digits.

Standard Deviation

The standard deviation is a statistic that measures the dispersion of a dataset relative to its mean and is calculated as the square root of the variance. It is calculated as the square root of variance by determining the variation between each data point relative to the mean. If the data points are further from the mean, there is higher deviation within the data set; thus, the more spread out the data, the higher the standard deviation.

In finance, standard deviation is a statistical measurement; when applied to the annual rate of return of an investment, it sheds light on the historical volatility of that investment. The greater the standard deviation of a security, the greater the variance between each price and the mean, which shows a larger price range. For example, a volatile stock has a high standard deviation, while the deviation of a stable blue-chip stock is usually

rather low. The formula for standard deviation shown below

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Mean Intensity

Intensity is the power transferred per unit area, where the area is measured on the plane perpendicular to the direction of propagation of the energy. In the SI system, it has units watts per meter (W/m^2). It is used most frequently with waves (e.g. sound or light), in which case the average power transfer over one period of the wave is used. Intensity can be applied to other circumstances where energy is transferred. For example, one could calculate the intensity of the kinetic energy carried by drops of water from a garden sprinkler. The word "intensity" as used here is not synonymous with "strength", "amplitude", "magnitude", or "level", as it sometimes is in colloquial speech.

Result Using Input and Destination Image

Sender side

Input Source and Destination Image

Morphing is the method to perform some operation on an image. Figure 4 shows the Input Source and Destination Image. In this method, we have to select an input image and an output image. There will be number of intermediate images in between input and an output image. Number of intermediate images depends upon the sender.

Figure 4. Input Source and Destination Image

Image Morphing

Sleek animated transition from one image into another is known as image morphing. After selecting an input and the destination image, number of intermediate images gets generated in between input and an output image. Figure 5 shows the image morphing using intermediate images. In morphing process, input image gets converted into destination image, by means of intermediate images. This process is referred to as image morphing. Figure 6 shows the image morphing output.

Figure 5. Image Morphing Using Intermediate Images

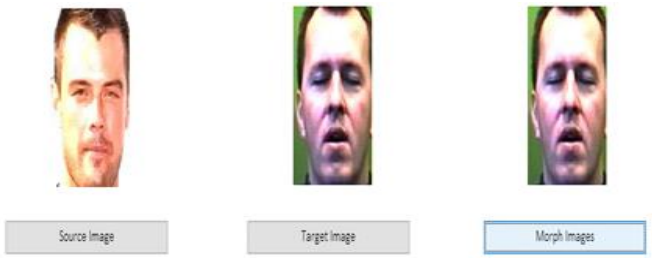

Figure 6. Image Morphing Output

Selected Intermediate Morphed Image

Our main aim is to hide the data behind one of the intermediate image. For this, one of the intermediate image gets selected, which were generated by the

sender in between input and destination image during the process of image morphing. Figure 7 shows selected intermediate morphed image. Suppose 'x' number of intermediate morphed image has been selected, then that 'x' image will act as a carrier image for data hiding.

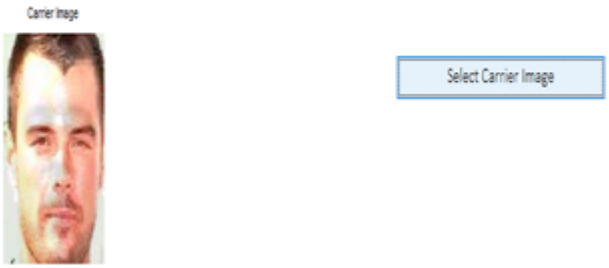

Figure 7. Selected Intermediate Morphed Image

Virtual Data Embedding

In this process, we are hiding the data virtually. For this virtual data embedding, we have to select one of the morphed image which will act as a carrier image. Then there will be selection of the master pixel component of the carrier image. Figure 8 shows the RGB channels for the extraction of master pixel component. After the extraction of RGB channels, we need to select master pixel component.

Figure 8. RGB channels for the extraction of Master Pixel Component

After this extraction, we need to select the secrete text. Figure 9 shows the selection of secrete data. Now there will be the process in which secrete data gets converted into the binary form. Figure 10 shows the conversion of secrete data into binary form.

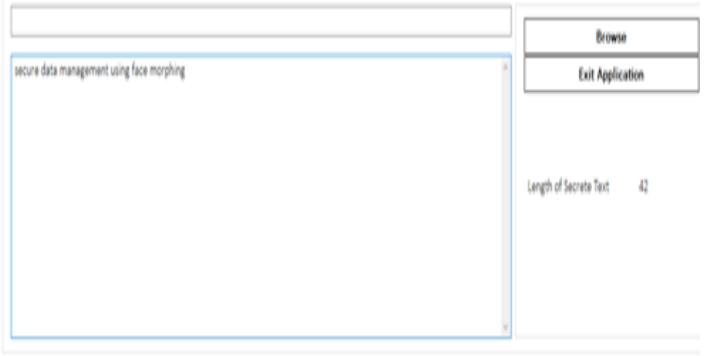

Figure 9 Selection of Secrete data

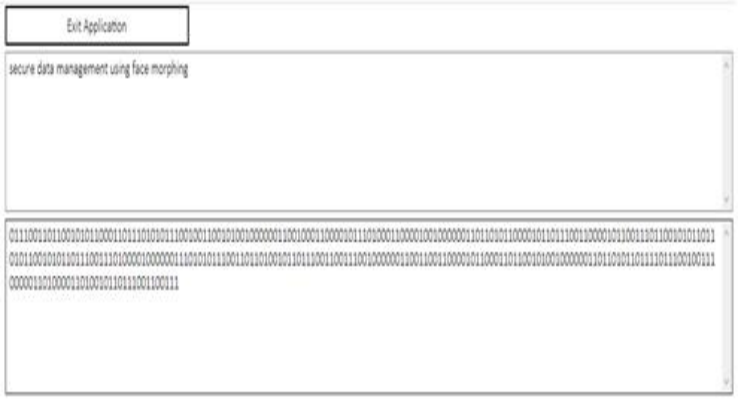

Figure 10. Conversion of secrete data into Binary Form

After this process, there will be virtual data hiding. In this process, position number is mentioned for each bit. Due to this position number to each bit, a random key gets generated of that secrete text. That key gets compressed by the technique known as key compression. Figure 12 shows key compression process. For that, we need to select the key length. In this way, data gets hide virtually. Figure 11 shows virtual data hiding

Figure 11. Virtual Data Hiding

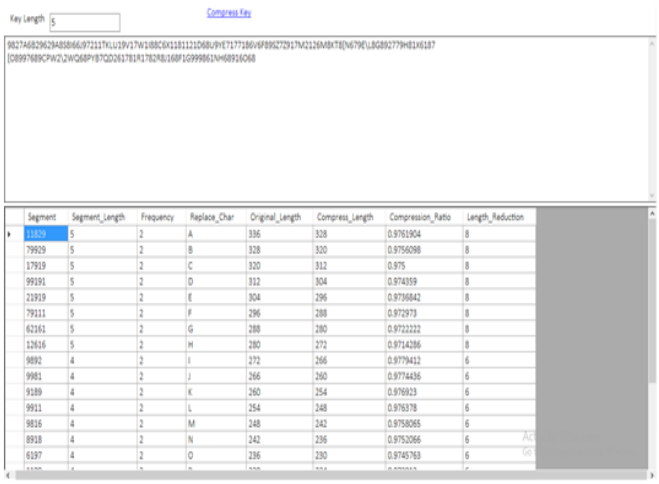

Figure 12. Key Compression

Delete Intermediate Morph Image

After hiding the data virtually, we need to delete all the intermediate morphed images. This process is necessary, as by deleting the intermediate morphed images, malicious third party will not be able to extract the secrete data, and our data remains safe. The next step is to send source and the destination images to the receiver side.

Receiver Side

Image Morphing

At the receiver side, we have to select an input image and a destination image. Figure 13 shows selection of intermediate images at the receiver side. These input and a destination image should be same as that of the sender side. Again at the receiver side, input image gets converted into the destination image by means of the morphing process. Figure 14 shows the image morphing process at the receiver side.

Figure 13. Selection of Intermediate Images at Receiver Side

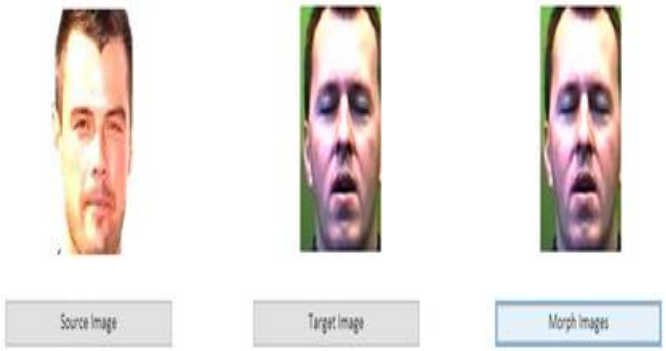

Figure 14. Image Morphing Process at Receiver Side

Select Intermediate Morph Image

For the extraction of the secrete data, it is necessary to select the morph image from one of the intermediate images, which were generated by the sender in between input and the destination image during the process of morphing. One important condition at the receiver side is that, number of intermediate images generated must be same as that of the number of intermediate images generated at the sender side. And the intermediate morph image which we are selecting for extracting the secrete data must have to be same with the one selected at the sender side for hiding the data. Figure 15 shows the selection of intermediate morph image at the receiver side.

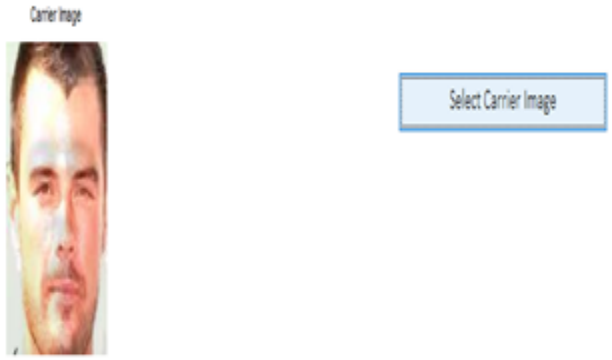

Figure 15. Selection of Intermediate Morph Image at Receiver Side

Virtual Data Extraction

For the extraction of virtual data, the compressed key which has been transferred to the receiver side has to be decompressed. The actual key gets generated due to decompression. For decompression, we need the master table and the compressed key. Figure 16 shows the extraction of decompressed key.

Figure 16. Extraction of Decompressed Key

Figure 17. Virtual Data Extraction

Again the process of image morphing has to be done. For the virtual data extraction, we need the master component. That master component gets converted into the random decompressed key. Then we need to convert that decompressed key into the binary form. And the binary form of that key gets converted into the secrete data. Figure 17 shows virtual data extraction. In this way, the secrete text gets extracted at the receiver side.

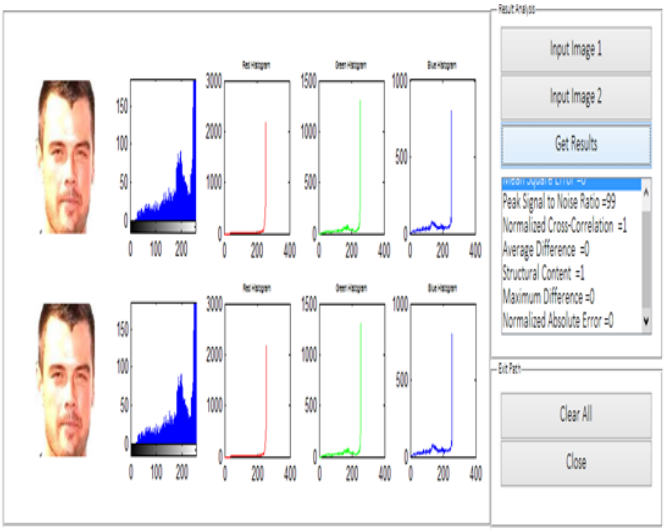

Figure 18. Result Analysis

VI. CONCLUSION

The Thesis has been designed and completed successfully. It is executing properly and is error free.

We have seen the image morphing right from introduction to application. Here we have used morph points instead of morph lines because of its simplicity, and also we have gone through the fundamentals of image morphing in which we have seen different processes used in image morphing.

I have also come to the conclusion that the Visual C++ programming language is one of the most stable programming languages and has a vast expanse and a very good scope for future software development.

REFERENCES

[1] Qiangfu Zhao and Mayuko Akatsuka, Generating Facial Images for Data embed Based on IGA and Image Morphing, 2012 IEEE.

[2] Lin Yuan and Touradj Ebrahimi, IMAGE TRANSMORPHING WITH JPEG, Third Edition, Morgan Kaufmann Publishers, 2015 IEEE.

[3] Yutaro Minakawa, Mitsuru Abe, Kentaro Sekine, and Qiangfu Zhao, "Neural Network Based Feature Point Detection for Image Morphing," Signal Processing, 90, pp. 727-752, 2010.

[4] Yun-Te Lin, Chung-Ming Wang, " A Novel Data Embedding Algorithm for High Dynamic Range Images " 2016 IEEE.

[5] Seong G. Kong, " Head Pose Estimation From a 2D Face Image Using 3D Face Morphing With Depth Parameters," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 24, NO. 6, JUNE 2015.

[6] Taheer Jamil,Liu Chao, "A framework for Automatic Testing of Image Processing Application," 2016 IEEE

[7] Ryota Hanyu, Kazuki Murakami, and Qiangfu Zhao, "Verification of an Image Morphing Based Technology for Improving the Security in Cloud Storage Services" 2014 IEEE

[8] Vanmathi C, Dr. S. Prabu, "Distortion less Reversible Data Embedding based on Dual Repeat Accumulate Coding Technique" 2014 IEEE.

[9] Sikha Mary Varghese, Alphonsa Johny , Dr.Jubilant Job "A Survey on Joint Data-Embedding and Compression Techniques based on SMVQ and Image Inpainting" 2015 International Conference on Soft-Computing and Network Security (ICSNS -2015), Feb. 25 – 27, 2015, Coimbatore, INDIA.

[10] Hao-Tian Wu, Reversible Image Data Embedding with Contrast Enhancement, IEEE SIGNAL PROCESSING LETTERS, VOL. 22, NO. 1, JANUARY 2015 81.

[11] Xinpeng Zhang, Jing Long, Zichi Wang, and Hang Cheng, Lossless and Reversible Data Embedding in Encrypted Images with Public Key Cryptography, 2015 IEEE

[12] Zhaoxia Yin, Andrew Abel, inpeng Zhang, Bi Luo "Reversible Data Embedding In Encrypted Image Based On Block Histogram Shifting," IEEE 2016.