Journal of Computer Science 6 (9): 987-992, 2010
ISSN 1549-3636


# Image Morphing Concept for Secure Transmission of Image Data Contents over Internet

[1]Anant M. Bagade and [2]S.N. Talbar
[1]Department of Information Technology, Pune Institute of Computer Technology,
Pune 411043, Maharashtra, India
[2]Department of Electronics and Telecommunication,
S.G.G.S Institute of Engineering and Technology, Nanded 431606, Maharashtra, India

**Abstract: Problem statement:** Morphing of images has evolved and become a challenging field in information hiding and data security. The objective of this study is to secure the image data over internet while transmitting using the concept of image morphing **Approach:** To address this issue, the study proposed the new approach for image data security using the concept of image morphing. The morphing algorithm produces the stego keys. These stego keys are securely transmitted over Internet using TCP/IP. The stego keys are transmitted through TCP/IP's identification field. The proposed method suggests how to transmit stego keys through identification field of IP. **Results:** Exterior sample points were manually identified and its inner values were interpolated using a triangular mesh. The complexity of Beier and Neely algorithm is O (n, p, w). Where n is the number of feature lines p is no of pixels in the image w is the amount of computation required for one pair feature line. By our approach the complexity is O (n,k),where n is the number of pixels and k is the number of triangles. Computations required are less, thus effect in increasing the performance of algorithm. The stego keys are identified during morphing process. As complexity reduces the speed of stego keys identification increases. **Conclusion:** The result showed that the proposed approach is efficient in terms of complexity and speed to generate the morph. The solution proposed for image data security over Internet is highly secure because the keys were transferred through IP identification field. The randomness in the identification field value makes this scheme no detectable.

**Key words:** Image morphing, image data security, internet

## INTRODUCTION

Morphing is defined as the animated transformation of one image into another (Beier and Neely, 1992; Wolberg, 1996; Lee *et al*., 1998; 1999). Morphing involves the image processing techniques of warping and cross dissolving (Smith, 1987). Morphing sequences produced by only using cross-dissolving (e.g., linear interpolation to fade from one image to another) of the source and destination image are visually poor (Lee *et al*., 1998; Wolberg, 1996). The results are poor because in general the features of the source and destination will not be aligned (Beier and Neely, 1992). When we simply cross dissolve, the double-exposure effect will be apparent in misaligned regions. In order to overcome this problem, warping is used to align the two images before cross dissolving (Smith, 1987; Wolberg, 1996). Warping determines the way in which the pixels in one image should be mapped to the pixels in the other image (Smith, 1987; Wolberg, 1996). For warping to work, the mapping of few important pixels needs to be specified. The motion for the other pixels is obtained by extrapolating the information specified for the control pixels (Whitaker, 2000). Since cross dissolving is very simple, warping becomes the major problem of morphing techniques. Morphing is simply a cross-dissolve applied to warped images. The different warping techniques differ in the way the mapping for the control pixels is to be specified and the interpolating technique used for the other pixels (Lee *et al*., 1998; Smith, 1987). These set of control pixels usually specify prominent features in the images.

Morphing refers to the combination of generalized image warping with a cross-dissolve between image elements (Beier and Neely, 1992; Lee *et al*., 1998; 1999; Wolberg, 1996). In order to morph between two images we define corresponding control pixels in source image $I_0$ and destination image $I_1$ (Karungaru *et al*.,

**Corresponding Author:** Anant M. Bagade, Department of Information Technology, Pune Institute of Computer Technology, Pune 411043, Maharashtra, India

2003). We then define each intermediate frame I of the metamorphosis by creating a new set of control pixels by interpolating the control pixels from their positions in $I_0$ to the positions in $I_1$. Both images $I_0$ and $I_1$ are then warped toward the position of the control pixels in I. These two warped images are cross-dissolved throughout the metamorphosis (Beier and Neely, 1992; Lee *et al*., 1998; Wolberg, 1996; Whitaker, 2000).

The growing use of the Internet has led to a continuous increase in the amount of data that is being exchanged and storage in various digital media (Hassain *et al*., 2005). This has led to some unexpected cases involving both benevolent and malevolent use of digital data (Jiang, 2008; Hassain *et al*., 2005) However, cryptography may not be secure because it tells the attacker clearly that some secret messages are contained in the data (Venkatraman *et al*., 2004). Usually the encrypted messages look very unnatural. Some malicious person or group may just concentrate on the unnatural Parts and use all computing resources to decrypt the messages (Venkatraman *et al*., 2004). Thus to make the information more secure, some other techniques are required.

**IPv4 header consideration:** This research specifically deals with data hiding possibilities in the IPv4 header (Xu *et al*., 2007). Scenarios are discussed that make use of flags and identification fields of the header (Xu *et al*., 2007). The layered architecture requires the IP datagram to encapsulate data received from the transport layer. Similarly, IP datagram headers encapsulate ICMP messages as well as IGMP's report and query messages (Paxson, 1999; Xu *et al*., 2007). Covert channels in the IPv4 header can, therefore also, associated with those identified in the TCP, ICMP or IGMP headers. This facilitates an increased amount of covert information tied with any of these messages. Therefore, flexibility of associating additional information with ICMP, IGMP and TCP traffic through IP header, is achieved, once covert channels are explored in IP header (Xu *et al*., 2007). Redundancies and multiple interpretations of the design strategy give rise to possible covert channels, which are exploited in the following IPv4 header manipulation schemes (Xu *et al*., 2007).

| Ver. | IHL | TOS | Total Length |
|---|---|---|---|
| Identification | | Flags | Fragment Offset |
| TTL | Protocol | Header Checksum | |
| Source Address | | | |
| Destination Address | | | |
| Options + Padding | | | |

Fig. 1: IPv4 Header fields

The IP header contains a number of areas where information can be sent to a remote host in covert manner (Fig. 1).

The colored field can be used to send the actual data to the remote host. We mainly focused to send the data through IP Identification field of IP header. The basic of exploitation relies in encoding ASCII values of the range 0-255 into the above areas. Using this method it is possible to pass data between hosts in packets that appear to be an initial connection request, establish data streams or other intermediate steps. Covert TCP chooses IDs that contain data to be sent (Xu *et al*., 2007). As simplified example, the string 'MORPH' can be embedded into a series of five packets where the first packet has an ID equal to ASCII value of 'M', the second has an ID equal to ASCII value of 'O' and so on.

IP identification field is used to distinguish fragments making up from one packet from fragments making up another. A scheme for embedding data in this field is described in (Xu *et al*., 2007). It uses a pseudorandom sequence, generated by a Toral Automorphosim system, to ensure that the modified field is random. The first 8 bits are used to carry data and next 8 bits are used to confirm transmission order.

## MATERIALS AND METHODS

Currently proposed a new information security technique based on image morphing and also proposed technique to transfer the stego keys securely over Internet. Originally image morphing is the process to change one image (Source image) to another (Destination image). It is interesting to notice that any intermediate image produced in the morphing process looks like a natural image and some of them can actually be used as a stego data.

While doing morphing different parameters (Stego keys) are used to morph between the two images. As the numbers of stego keys are more, the security level will be high while de morphing the image data at destination place.

The work done is a pixel wise transformation from source image to destination image (Beier and Neely, 1992). The transformation is given by the following formulas (Beier and Neely, 1992).

The coordinate mapping u and v are:

$$u = (X1 - P1).(Q1 - P1)/ \| Q1 - P1 \|^2 \quad (1)$$

$$v = (X1 - P1).Perp(Q1 - P1)/ \| Q1 - P1 \| \quad (2)$$

The value u is the position along the oriented line P1Q1 and v is the distance from this line for each pixel X1 of the intermediate image.

Perp is the vector perpendicular to the given vector.

The calculation of X2 in the source image and X3 in the destination image using u and v are given by using the given formula.

$$X2 = P2 + u.(Q2 - P2) + (v.Perp(Q2 - P2/ \| Q2 - P2 \|)) \quad (3)$$

$$X3 = P3 + u.(Q3 - P3) + (v.Perp(Q3 - P3/ \| Q3 - P3 \|)) \quad (4)$$

To specify many features in image, we have more complex transformation, which include a weighted combination of the transformation performed by each line pair.

Therefore, the weight of each pair is computed as follows (Beier and Neely, 1992):

$$\text{Weight} = [(\text{length})^{p}/(a+\text{dist})]^{b} \quad (5)$$

length = The length of a vector
dist = The distance from pixel to the vector
a,b,p = Constants can be used to change the relative effect of the vectors

Metamorphosis between two images is defined by first specifying the directed feature lines between each image (Beier and Neely, 1992). To align each object of the morphed image, the vertices of these feature lines are linearly interpolated using:

$$V_i = (1-\alpha)\, Vi^{0} + \alpha V_i^{1} \quad (6)$$

In the above equation $V_i$ are the line feature vertices of the morphed image, $Vi^{0}$ and $V_i^{1}$ are the feature vertices specified in each source image and as in Eq. 7. α is a scalar taking values $0 \le \alpha \le 1$.

The resulting field morphing function f(X) is defined as:

$$f(X) = X + \sum_i Wi(Xi' - X) / \sum_i Wi \quad (7)$$

f(X) defines the deformation field d(x,y).

In the (Beier and Neely, 1992) morphing algorithm d(x,y) was computed via the use of a sparse set of directed line features. For a point in the morphed image, its corresponding point in each source image was computed by weighting the corresponding point found using each line feature. Although the Beier and Neely algorithm does a good job of producing the convincing object metamorphosis its performance is highly dependent on right choice of line features and values of a, b and p. Observing the Eq. 6, one fiends that with Beier and Neely d(x,y)~f(x,y),where f(x,y) is a field warping function. The method discussed will achieve better approximation to d(x,y) that require less manual intervention.

**Triangular image warping algorithm:** With this algorithm, a few exterior sample points of d(x,y) are manually specified and its inner values are interpolated using a triangular mesh defined over a convex hull of the sample points. To compute the deformation field $d_r(x,y)$ a triangular mesh is computed over a convex hull of the feature points of each source image. Assuming α = 0.5, applying this method gives the triangular mesh. This mesh is directly applied to the feature points of each source image and the morphed image.

Let X be a point in the morphed image located inside triangle T(P,Q,R) with vertices P,Q,R. Let T'(P',Q',R') be the corresponding triangle in one of the source images and X' be the corresponding point in T'. We can express X as a linear combination of the vertices:

$$X = A + \beta(B - A) + \gamma(C - A) \quad (8)$$

$$X = \alpha A + \beta B + \gamma C$$

Where $\alpha = 1-(\beta+\gamma)$ such that $\alpha+\beta+\gamma = 1$.To find X' we apply the weights α, β, γ found using Eq. 8 to the vertices of T':

$$X' = \alpha A' + \beta B' + \gamma C' \quad (9)$$

**Algorithm: Triangular image warping:** Let I and $I_w$ be the input and warped images respectively, M be the triangular mesh defined over the convex hull of the feature points in $I_w$.

```
for all X Є I do
    for all T Є M do
        Compute α, β and γ using Eq. 8
If α ≥ 0, β, γ ≤ 1 then
 Compute X' using Eq. 9
Iw.(X)=Interpolate(I,X')
end if
  end for
    end for
```

Equation 8 and 9 collectively defines the deformation field $d_r(x,y)$. A point X is a running time of O(n,k),where n is the number of pixels and k is the number of triangles. Better performance can be achieved by considering the points of each triangle instead of looping over entire image. The point X is inside a triangle if $\alpha \ge 0$ and $\beta, \gamma \le 1$.

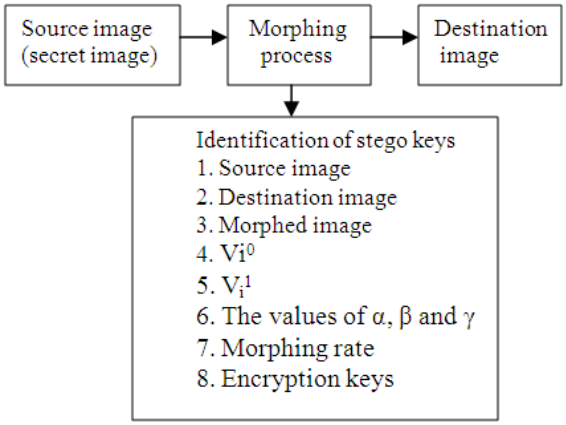


Fig. 2: Stego keys identification while morphing

**Identification of stego keys during morphing:** The inputs for the above algorithm are the two images one is source and destination image. The parameters while morphing required are source image, destination image, the morphed image, the feature points of source image, the feature points of destination image $Vi^0$ and $V_i^1$ respectively, morphing rate, encryption keys, the values of α, β and γ. These are the stego keys identified while morphing process.

The stego keys generated by above morphing process shown in Fig. 2 are to be sent to destination place by using a secure channel. The solution to send these stego keys and images securely over internet to destination place is depicted as follows. The stego keys identified in Fig. 2 are sent to destination host by using the IP identification field. The algorithms given below are practically used to transfer the data through IP identification field.

**Algorithm: Text encryption:**

Input: Text Data
Output: Encrypted text

Start
Read the file to be sent character by character.
Do loop
For (x= first character of file; x++; x! ='\0')
Read (character's ASCII value)
Reverse (bytes of ASCII value)
Assign (reversed value to the IP identification field of packet). Maintain sequence number of each packet.
Follow steps until x= '\0' i.e. while x='\0'
File reading and encryption of each character is finished.
End

**Algorithm: Text decryption:**

Input: Encrypted text
Output: Original text

Start
Read (jpcap object value)
Extract (IP identification field value)
Store (all IP identification field values in vector)
Sort (vector by sequence number)
Do loop
    For (x = ASCII value first byte of cipher data; x! = '\0'; x ++)
Decrypt (reverse ASCII value)
Store (reversed value in new file)
While x! = '\0'
End.

**Algorithm: Image encryption:**

Input: Image Data
Output: Encrypted image

Start
Ask user to enter sub key value (any numeric value)
K, k1 = sub key value
Read image data as in binary format
Do loop
For (x = first byte of image or audio data; x! = '\0' x ++)
Convert (a=eight bytes in numeric value) Compute (b = ASCII value of a)
Encrypt (i = b + k1)
Assign ( i as IP identification field value)
K1++
Send packet with assigned IP identification value = i
While x! = '\0'
    Read (k)
    K2= Reverse (k)
Assign (k2 as IP identification field value)
Send last packet with K2 as IP identification field value
End.

**Algorithm: Image decryption:**

Input: Encrypted image data
Output: Original image

Start
Read (jpcap object values)
Extract (IP Identification field values)
Store (in vector expect end packet's value)
Store (k, k1= last packet value)

Reverse (variable value)
Compute (value from reversed ASCII value)
Sort (vector)
Read (vector byte by byte)
Do loop
For (x = first eight bytes of vector; x! = '\0'; x ++)
Decrypt (i = x - k1)
Compute (b =ASCII value of numeric value)
Convert (b in 8 bits format)
Store (8 bits)
K1--
For next byte
Decrypt (i = x - k1)
Compute (b =ASCII value of numeric value)
Convert (b in 8 bits format)
Append (8 bits)
While x! = '\0'
Read stored file with all appended bits
End.

## RESULTS AND DISCUSSION

The stated algorithm for image warping is practically implemented by taking source and destination image as an input. The algorithm will work on same size images as source and destination. There is a need to draw the points on both the images. It generates 'n' number of intermediate frames. The quality of image depends upon considering the points of each triangle instead of looping over entire image. The Fig. 3 shows the practical results while morphing with respect to each α value. The Fig. 4 shows the whole morphing process.

The stated algorithm will take less time for computation because it considers the points inside a triangle defined over the convex hull. The Table 1 gives the average warping time for each of the available algorithms.

As compared with Table 1, this algorithm takes 0.085 sec with 10 sample points on Intel P-IV 2.66 GHz processor. The less time required for generation of final morph.

As the algorithm is fast the stego key identification process is fast. The stego keys identified are sent over a secure channel over Internet by using identification field of IP header.

This study presented a conceptual schema of using identification field to send the image and text data contents generated by morphing over Internet using IP header. The randomness in the identification filed values makes this schema non detectable. As the number of stego keys is more, more security will be there for the image data. The algorithms stated for image and text data encryption and decryption are useful to send the data through identification field of IP header and decrypt the data at destination host.

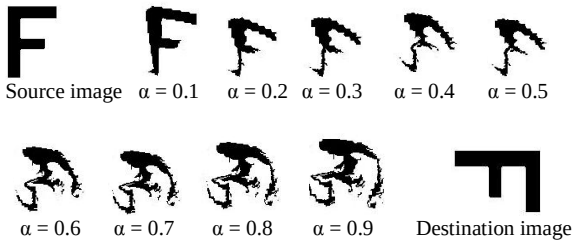


Fig. 3: Metamorphosis between source and destination images

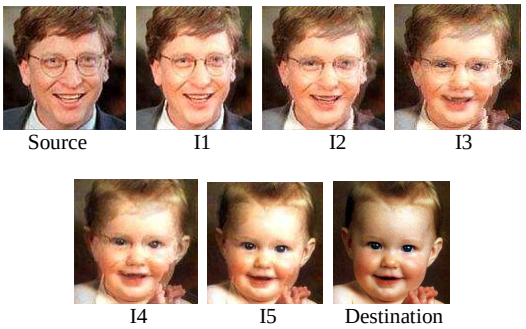


Fig.4: Morphing process

Table 1: Average warping time

| Algorithm name | Computation time (sec) |
|---|---|
| Mesh warping | 0.15 sec with a 10×10 mesh |
| Feature-based warping | 0.75 sec with 11 feature lines |
| Thin plate spline warping | 0.45 sec with 5 control points |

## CONCLUSION

By observing the results it is clear that the stated algorithm is efficient than the existing algorithm in terms of time complexity. The warped algorithm takes running time of O(n,k) where n is the number of pixels and k is the number of triangles. The stego keys identified during morphing process are sent securely through identification field of IP header. The encryption and decryption algorithms are used to send the data in encrypted format through identification filed and get the original data at destination host using decryption algorithms stated. The image data is highly secure because of the stated methods.

The future research is to identify the stego keys automatically during the morphing process and to enhance the mechanism stated to send the stego keys securely over Internet.

## ACKNOWLEDEGMENT

Earlier version of this study was published by TATA McGraw Hill India, 2009.

## REFERENCES


Beier, T. and S. Neely, 1992. Feature-based image metamorphosis. ACM SIGGRAPH Comput. Graph., 26: 35-42. DOI: 10.1145/133994.134003

Hassain, K., N. Abdulla, S. Rajan and G. Moussa, 2005. Preventing the capture of sensitive information. Proceedings of the 43rd Annual Southeast Regional Conference, Mar.18-20, ACM Press, Kennesaw, Georgia, pp: 154-159. DOI: 10.1145/1167253.1167291

Jiang, N., 2008. A novel analysis method of information hiding. Proceedings of the 2008 Congress on Image and Signal Processing, May 27-30, IEEE Xplore Press, Sanya, China, pp: 621-625. DOI: 10.1109./CISP.2008.62

Karungaru, S., M. Fukumi and N. Akamatsu, 2003. Morphing face images using automatically specified features. Proceedings of the 2003 IEEE International Symposium on Micro-NanoMechatronics and Human Science, Dec. 30-30, IEEE Xplore Press, Cairo, pp: 741-744. DOI: 10.1109/MWSCAS.2003.1562393

Lee, S., G. Wolberg and S.Y. Shin, 1998. Polymorph: Morphing among multiple images. IEEE Comput. Graph. Appli., 18: 58-71. DOI: 10.1109/38.637304

Lee, A.W.F., D. Dobkin, W. Sweldens and P. Schroder, 1999. Multiresolution mesh morphing. Proceeding of the 26th Annual Conference on Computer Graphics and Interactive Techniques, Aug. 1999, ACM Press, New York, USA., pp: 343-350. DOI: 10.1145/311535.311586

Paxson, V., 1999. End-to-end internet packet dynamics. IEEE/ACM Trans. Network., 7: 277-292. DOI: 10.11109190.779192

Smith, A.R., 1987. Planar 2-pass texture mapping and warping. Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, (CGIT'87), ACM Press, New York, USA., pp: 263-272. DOI: 10.1145/37401.37433

Venkatraman, S., A. Abraham and M. Paprzycki, 2004. Significance of steganography on data security. Proceedings of the International Conference on Information Technology: Coding and Computing, Apr. 5-7, IEEE Computer Society, Washington DC., USA., pp: 347-351. DOI: 10.1109/ITCC.2004.1286660

Whitaker, R.T., 2000. A level set approach to image blending. IEEE Trans. Image Process., 9: 1849-1861. DOI: 10.1109183.877208

Wolberg, G., 1996. Recent advances in image morphing. Proceedings of the 1996 Conference on Computer Graphics International, IEEE Computer Society, Washington DC., USA., pp: 64-71. DOI: 10.1109/CGI.1996.511788

Xu, B., J.Z. Wang and D.Y. Peng, 2007. Practical protocol steganography: Hiding data in IP header. Proceeding of the 1st Asia International Conference on Modeling and Simulation, Mar. 27-30, IEEE Xplore Press, Phuket, pp: 584-588. DOI: 10.1109/AMS.2007.80