ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 9 Issue 01 January 2020, Page No. 24925-24933
ISSN: 2319-7242 DOI: 10.18535/ijecs/v9i01.4417

A Case Study on Image Morphing Through Data Hiding Technique

Beena Biswas, Dr. Vaibhav Sharma

Research Scholar (IT&CA)
CVRU Kota Bilaspur
Associate Prof.
CVRU Kota Bilaspur

Abstract
Today very important issue is to hide our sensitive data which comes under data hiding which means hiding our sensitive data as well as information from out side world. Image will become an important and sensitive information when it contain any sensitive information in it for many reasons like communication owner identification etc. Many techniques have been introduces one of which is image morphing which means changes its form from one to another without changing its other parameter to make it more secure . For morphs between faces, the metamorphosis does not look good if the two faces do not have the same shape approximately. In this project, we implemented a morphing scheme which would combine cross-dissolve with warping methods to give good morphs. This is based on "Feature-Based Image Metamorphosis. The morph process consists of a warping stage before cross-dissolving so that the two images have the same shape. The warp is specified, in this case, by a mapping between lines in the first and second images. In the following discussion, the first image will be called the source image and the last image will be called the destination image. In this survey paper we will show you different types of morphing algorithms, procedures and techniques.

Keywords: Morping ,warping, metamorphism.

Introduction:
Morphing is an image processing technique used for the metamorphosis from one image to another. The idea is to get a sequence of intermediate images which when put together with the original images would represent the change from one image to the other. The simplest method of transforming one image into another is to cross-dissolve between them. In this method, the color of each pixel is interpolated over time from the first image value to the corresponding second image value. This is not so effective in suggesting the actual metamorphosis. For morphs between faces, the metamorphosis does not look good if the two faces do not have the same shape approximately. In this project, we implemented a morphing scheme which would combine cross-dissolve with warping methods to give good morphs. This is based on "Feature-Based Image Metamorphosis" by Thaddeus Beier and Shawn Neely. The morph process consists of a warping stage before cross-dissolving so that the two images have the same shape. The warp is specified, in this case, by a mapping between lines in the first and second images. In the following discussion, the first image will be called the source image and the last image will be called the destination image.

Warping: There are two ways to warp an image. They are

1. **Forward Mapping**
   In this method, each pixel in the source image is mapped to an appropriate place in the destination image. Thus, some pixels in the destination image may not be mapped. We need interpolation to determine these pixel values. This mapping was used in our point-morphing algorithm.

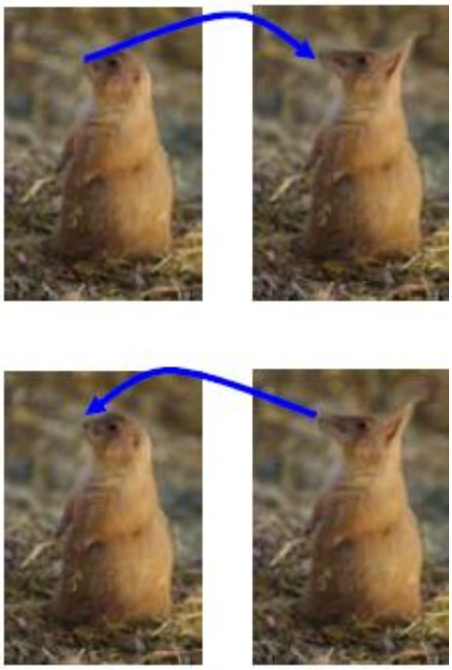


Figure 1: Forward Mapping

2. **Reverse Mapping**
   This method goes through each pixel in the destination image and samples an appropriate source image pixel. Thus, all destination image pixels are mapped to some source image pixel. This mapping has been used in the Beier/Neely line-morphing method.
   In either case, the problem is to determine the way in which the pixels in one image should be mapped to the pixels in the other image. So, we need to specify how each pixel moves between the two images. This could be done by specifying the mapping for a few important pixels. The motion of the other pixels could be obtained by appropriately extrapolating the information specified for the control pixels. These sets of control pixels can be specified as lines in one image mapping to lines in the other image or points mapping to points.

Point Warping
This method of image warping is based on a forward mapping technique, where each pixel from the input image is mapped to a new position in the output image. Since not every output pixel will be specified, we must use an interpolating function to complete the output image. We specify several *control points*, which will map exactly to a given location in the output image. The neighboring pixels will move somewhat less than the control point, with the amount of movement specified by a weighting function consisting of two separate components, both dependent on the distance from the pixel to each control point in the image.

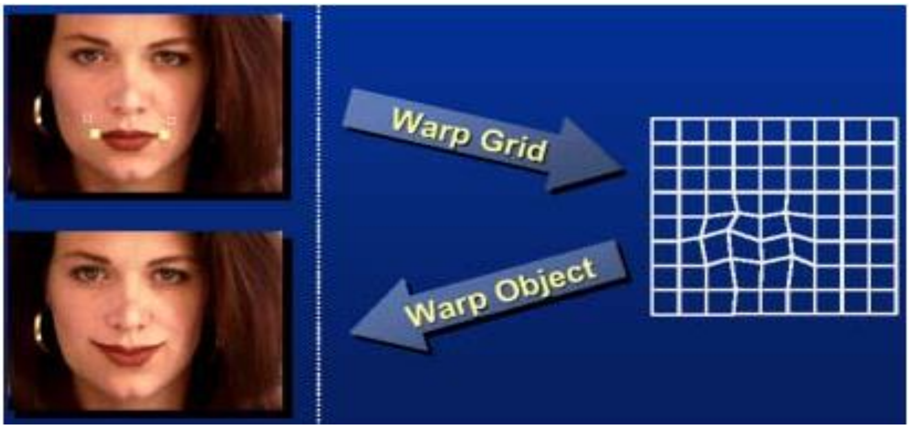


Figure 2: Point warping

The first component of the weighting function is a Gaussian function which is unity at the control point and decays to zero as you move away from the control point. The idea is to have pixels far away from a control point be unaffected by the movement of that point. The problem with this scheme is that each pixel is affected by the weighting functions of *every* control point in the image. So even though the weighting function at a control point may be one, that point will still be affected by the movement of every other control point in the image, and won't move all the way to its specified location.

In order to overcome this effect, we designed the second component of the weighting function, which depends on the relative distance from a pixel to each control point. The distance to the nearest control point is used as a reference, and the contribution of each control point is reduced by a factor depending on the distance to the nearest control point divided by the distance to that control point. This serves to force control point pixels to move exactly the same distance as their associated control points, with all other pixels moving somewhat less and being influenced most by nearby control points.

The following examples show some of the uses of warping. The first set of images shows how facial features and/or expressions can be manipulated. The second set shows how the overall shape of the image can be distorted (e.g., to match the shape of a second image for use in the morphing algorithm).

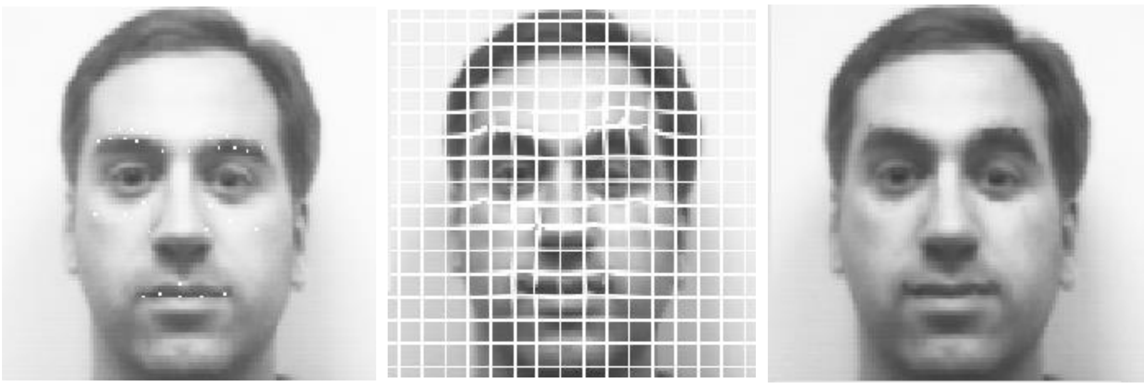


Figure 3: Morphing Image

The figure 3 is a "normal" Roger with the control points we used identified. The figure 4 shows how those points were moved. The figure 5 is the more cheery Roger that results.

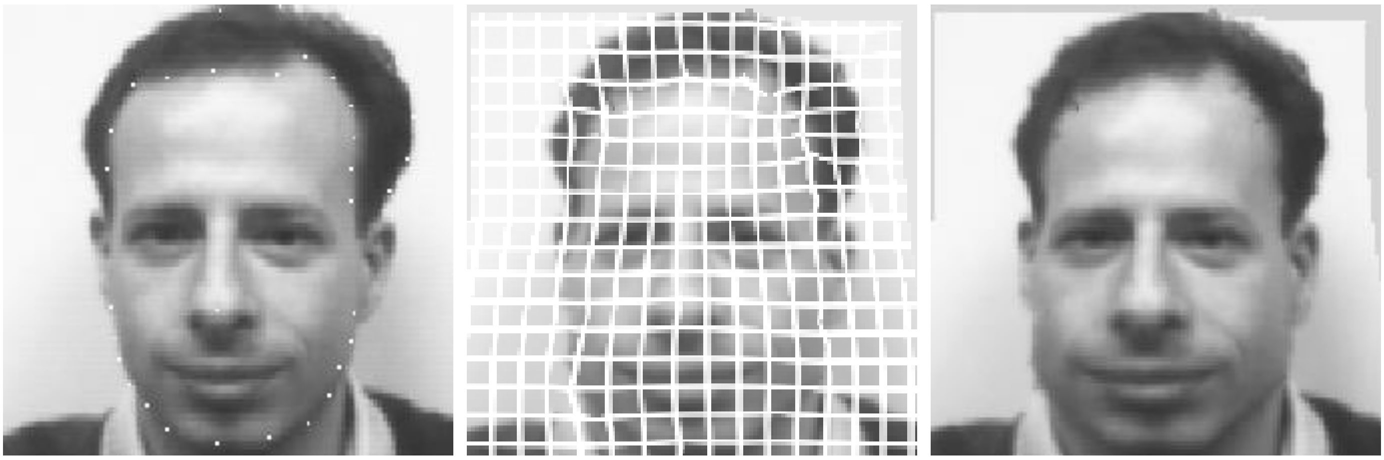


Figure 4: Morphing Image

Here we show what happens to Kevin when he combs his hair a little differently and goes off his diet.

Image Morphing:

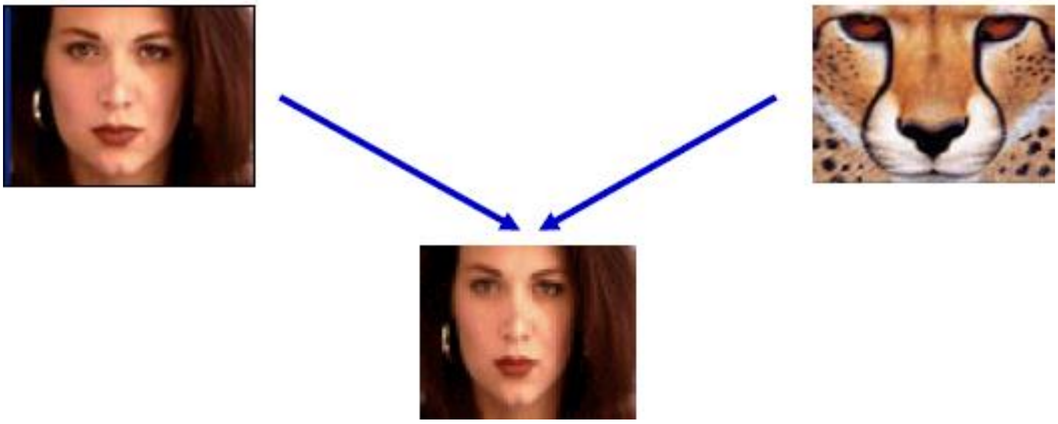
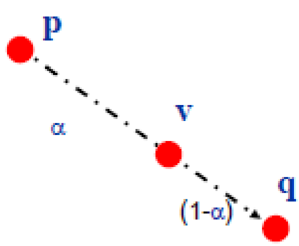


Figure 5 Image morphing

Averaging vectors

v = **p** + α (**q** – **p**) = (1 - α) **p** + α **q** where α = ||**q** - **v**||



Where p and **q** can be anything:

- points on a plane (2D) or in space (3D)
- Colors in RGB or HSV (3D)
- Whole images … etc

Method 1: Cross-Dissolving / Cross-fading

Interpolate whole images:

I halfway = α***I**1 + (1- α)***I**2

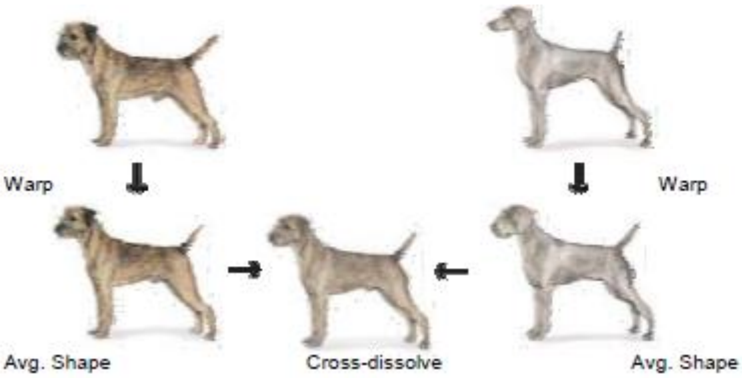
- This is called **cross-dissolving** in film industry
- But what if the images are not aligned?

Method 2: Align, then cross-disolve

Align first, then cross-dissolve

- Alignment using global warp – picture still valid

Method 3: Local warp & cross-dissolve



Morphing procedure:

1. Find the average shape (the "mean dog"☺) local warping
2. Find the average color Cross-dissolve the warped images

Morphing Sequence

1. **Input:** two images **I**0 and **I**N

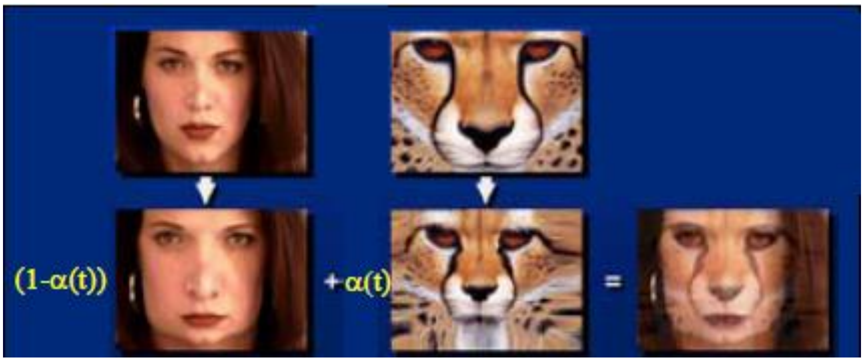
2. **Output:** image seq. $\mathbf{I}_i$, with $i=1..N-1$

3. User specifies sparse correspondences on the images

Pairs of vectors $\{(\mathbf{p}_{0j}, \mathbf{p}_{Nj})\}$

For each intermediate frame $\mathbf{I}_t$

- Interpolate feature locations $\mathbf{p}_{ti} = (1-\alpha(t))\,\mathbf{p}_{0i} + \alpha(t)\,\mathbf{p}_{1i}$
- Perform **two** warps: one for $\mathbf{I}_0$, one for $\mathbf{I}_1$
- Deduce a dense warp field from a few pairs of features
- Warp the pixels
- Linearly interpolate the two warped images



Comparative Study of existing algorithms:

Algorithm Name	**Computation Time**
Mesh Warping	0.15 s with a 10X10 mesh
Feature-based Warping	0.75 s with 11 feature lines
Thin Plate Spline Warping	0.45 s with 5 control points

Mesh Warping: Mesh warping is a two-pass algorithm that accepts a source image and two 2-D arrays of coordinates S and D. The S coordinates are the control pixels in the source image. The D coordinates specify the location to which the S coordinates map. The final image is the source image warped by means of meshes S and D. The 2-D arrays in which the control points are stored impose a rectangular topology to the mesh. The only constraint is that the meshes defined by both arrays be topologically equivalent i.e. no folding or discontinuities. Therefore the entries in D are coordinates that may wander as far from S as necessary, as long as they do not cause self-intersection.

The first pass is responsible for re-sampling each row independently. An intermediate array of points I, whose x coordinates are same as those in D and whose y coordinates are the same as those in S, is created. Vertical splines are generated to fit each column of data in S and I. The data for each span (region) in a row is interpolated to create intermediate image I.

The second pass is responsible for re-sampling each column independently. Horizontal splines are generated to fit each row of data in arrays I and D. The data for each span (region) in a column is interpolated from intermediate image I to create destination image D.

The collection of vertical splines fitted through S and I in the first pass, along with the horizontal splines fitted through I and D in the second pass, are shown in Figure 1

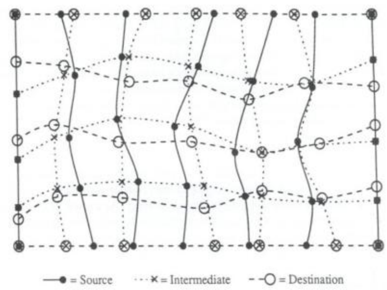


Figure 6

Feature Based Image Warping: This method gives the animator a high level of control over the process. The animator interactively selects corresponding feature lines in the 2 images to be morphed. The algorithm uses lines to relate features in the source image to features in the destination image. It is based upon fields of influence surrounding the feature lines selected. It uses reverse mapping (i.e. it goes through the destination image pixel by pixel, and samples the correct pixel from the source image) for warping the image.

A pair of lines (one defined relative to the source image, the other defined relative to the destination image) defines a mapping from one image to the other.

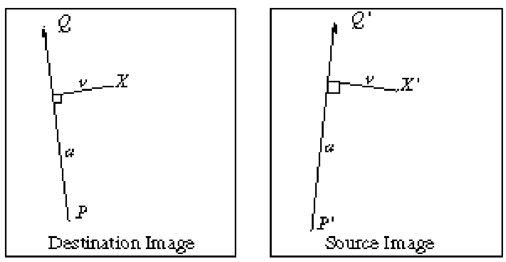


Figure 7

Thin Plate Spline (TPS) Based Image Warping:

Thin-plate Spline is a conventional tool for surface interpolation over scattered data. It is an interpolation method that finds a *"minimally bended"* smooth surface that passes through all given points. The name "Thin Plate" comes from the fact that a TPS more or less simulates how a thin metal plate would behave if it was forced through the same control points.

Let us denote the target function values v_i at locations (x_i , y_i) in the plane, with i=1,2,…….p , where p is the number of feature points. In particular, we will set v_i equal to the coordinates (x_i' , y_i') in turn to obtain one continuous transformation for each coordinate. An assumption is made that the locations (x_i , y_i) are all different and are not collinear.

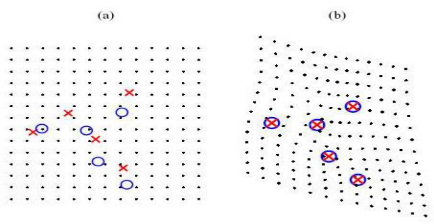


Figure 8

Figure 8 is a simple example of coordinate transformation using TPS. Let's consider two sets of points for which we assume the correspondences to be known (a). The TPS warping allows an alignment of the points and the bending of the grid shows the deformation needed to bring the two sets on top of each other (b). Note that in the case of TPS applied to coordinate transformation we actually use two splines, one for the displacement in the x direction and one for the displacement in the y direction. The two resulting transformations are combined into a single mapping.

Comparison

When we compared the three algorithms on a scale of computational speed, we found the Mesh Warping algorithm to be the best. This results from the fact that the region is divided into a mesh and each mesh patch essentially has a local region of influence. Hence the computation is localized and independent, thus allowing for high level of parallelism. But a word of caution is in place. The computational advantage ofthe Mesh Warping algorithm is greatly offset by the huge amount of time overhead required to select mesh nodes all over the image. The main disadvantage of the Feature-based and Thin Plate Spline algorithms is speed. As the warping here is global the entire set of feature lines/control pixels that are specified need to be referenced for each pixel. As a result amount of time taken for each frame is proportional to the product of the number of pixels in the image and the number of control lines/pixels used.Table-1gives the average warping time for each of our algorithms

Finally we wish to put in a word on the individual advantages and disadvantages of the three algorithms. The mesh warping algorithm requires that all four edges of the source and destination image be frozen. But this seemingly limiting constraint provides the mesh warping algorithm with its simplicity of implementation. The mesh warping algorithm also requires that the source and destination meshes be topologically equivalent i.e. no folding or discontinuities. This all adds to the problem of selecting mesh nodes spread through out the image. In case of the feature-based warping algorithm, sometimes unexpected and unwanted interpolations are generated due to some line combinations. Additional image processing efforts are required to fix these distortions and improve the quality of results.

Conclusion

In this paper we have shown different morphing algorithms and its comparisons. Paper also contains steps involved in morphing techniques it show how morphing process takes place and how it will be done. We have also include concept of data hiding in it to make a relation between data hiding and image morphing this will help researchers to correlate both the concept and motivate them to work on the field of image morphing nad data hiding. It will definitely able to provide a secure and reliable platform for the researchers.

Reference:

[1] Bhumika H. Bhatt, Hita M. Joshi, “Feature based Image Morphing”, International Journal of Computer Science & Technology, vol. 2, issue 3, pp. 46-47, sept. 2011.

[2] Urvshi Bhushan, Dr. G.P.Saroha and Disha Tivari, “An implemention of image morphing through mesh morphing algorithm”, Journal of Computer Science, vol. 2, issue 7, pp. 74-76, Jul. 2012.

[3] George Wolberg, “Image Morphing Survey”, Department of Computer Science, City College of New York, New York, USA, Jun. 1998.

[4] Seungyong Lee, George Wolberg and Sung Yong Shin, “Polymorph:An Algorithm for Morphing among multiple images”, Visual Computer(1998), Springer-Verlag, 1998, pp. 360-371, Jul. 2008.

[5] Manfred Kopp,Werner Purgthofer, “Multi Resolution Image Morphing”, Vienna Univ. of Technology, Institute of Computer Graphics, Austria, Jul. 2004.

[6] Srinath Setty, Swati Rallapalli “Implemention of Morphing Algorithm with feature based Image metamorphosis”, The Univ. of Texas, Austin, Jun. 2010

[7] Anat M. Bagade, S.N.Talbar, “Image morphing cocept of secure transmission image data contents over internet”, Journal of Computer Science, pp. 987-992, Jul. 2010.

[8] George Wolberg, “Recent Advance in Image Morphing”, M. Eng. thesis, Department of Computer Science, City College of New York, New York, USA, Jun. 1996.

[9] Prashant K. Oswal,Prashanth Y. Govindaraju, “Image Morphing: A Comparative Study”, Department of Electrical and Computer Engineering, Clemson Univ., Clemson Eng., Clemson, sept. 2010.

[10] Lei Zhu,Yan Yang,Steven Haker and Allen Tannenbaum, “An Image Morphing Technique Based on Optimal Mass Preserving Mapping”, IEEE Transactions on Image Processing, vol.16, no.6,pp. 1481-1483, Jun. 2007.

[11] Kinjal N. Shah, Tushar Gohil, “Image Morphing”, Department of Information Technology, SCET, Tech. Rep. 6-9, Sept. 2007-2008

[12] Aleandru Vlad FECIORESSCU, “Image Morphing Techniques,” Journal of Industrial Design and Engineering Graphics, vol.5, issue 1, pp. 25-28, Jun. 2010.

[13] Pramod Potlri, Krshna Sagriraj and Venkatachalam Tubati "Image Morphing: Feature based,view,Mesh", Department of Electrical and Computer Eng., Clemson Univ., Clemson Eng., Clemson, Aug. 2009.

[14] Robert de Waard, “Image Morphing Algorithm”, Aug. 2011. http://www.infi.nl/blog/view/id/116/Image_Morph_Algorithm

[15] Martin Bichsel, “Automatic Interpolation and Recognition of Face Images by Morphing”, 2nd international conference on automatic face and gesture recognition, pp. 128-135, Jan. 2001.

[16] Manolis,Manolis Kamvysselis, Matt D. Blum and Hooman Vassef,Krzysztof Gajos, “3D Morphing”, Tech. Rep. 1-13, Nov.2003

[17] http://alumni.media.mit.edu/~maov/classes/comp_photo_vision08f/lect/07_Image%20Morphing.pdf

[18] https://www.owlnet.rice.edu/~elec539/Projects97/morphjrks/morph.html

[19] Jignasha B. Panchal, “An Implementation of Enhanced Image Morphing Algorithm using Hybrid Approach” International Journal of Computer Applications (0975 – 8887) Volume 66– No.20, March 2013