An Investigation into the feasibility of Human Facial Modeling

Soteri Panagou and Shaun Bangay

soteri@rucus.ru.ac.za, cssb@cs.ru.ac.za

Department of Computer Science[1]

Rhodes University

Grahamstown, 6140

1. Abstract

In this paper, virtual videoconferencing is investigated. The goal of the project is to develop a cheap, realistic yet real-time 3D face-modeling tool using a number of camera feeds. Bandwidth utilisation can then be minimised by transferring only the actual model (and subsequently, changes to the model) of the actor's face. Using a virtual reality system, reconstruction of a human face is performed and transmitted, thereby reducing bandwidth requirements. The modeling process is discussed, with results.

Keywords: Virtual Videoconferencing, Compression, surface reconstruction

2. Introduction

Videoconferencing places undue strain on our already limited bandwidth capacity. Much research is being done into trying to find alternate forms of infrastructure which can support the requirements of such systems (i.e. real time video/audio transfers). Examples include ISDN and, of late, ADSL and ATM. The ISDN route is usually followed to make video-conferencing a reality. This is a route which is, unfortunately, very costly. This work in progress attempts to address this economic problem with virtual videoconferencing. With processing power increasing exponentially and cost decreasing (according to Moore's law) but the bandwidth capacity remaining relatively constant (constant in terms of our local South African conditions), ways must be found to make use of this processing capacity and in so doing, reduce bandwidth requirements i.e. bandwidth is a scarcer resource than processing power.

Instead of transmitting video back and forth between the various parties involved in the communication, we envisage compressing the transmission signal (i.e. video) in some intuitive way. The compression involved here will be to encode the main object of the video in the form of a 3D model. This model can then be transmitted to the destination where it can then be appropriately rendered. Subsequently, changes in facial expressions can either applied to the reconstructed model before transmission or the changes themselves can be transmitted to the destination(s). The first method incurs additional processing and higher bandwidth overhead, while the latter shifts this overhead to the destination, i.e. additional processing at the destination. The idea is to transmit a truthful model of the user's face, while at the same time cutting down on bandwidth utilisation. The focus is thus on processing a camera feed so as to produce an accurate 3D model of the object (in this case a face), and subsequently to track changes to the real-world object and reflect these changes in the reconstructed object.

3. Surface Reconstruction

A picture is the result of a projection from a 3-dimensional world onto a 2-dimensional plane, leading to a loss of depth information. Much research has been done into trying to recover some of the depth information of a scene by evaluating multiple pictures of the same scene from different angles. Examples of this type of work include [Niem and Buschmann,1994] who deal with the problem in a more obvious fasion, i.e. a convex object extraction technique similar to the one described below. Another author, [Zhang et al, May 1994], describes normalised cross correlation (section 3.2 below) and how this can be used to analytically. What follows is a description of techniques that have been implemented by both other researchers as well as the authors. It must be noted although there are numerous surface reconstruction implementations which use lasers or structured light, these are prohibitively expensive when compared to the equipment required for the algorithms mentioned below, namely cameras.

[1] This work has been funded by the FRD, and is being undertaken under the auspices of the Rhodes University Computer Science Department's Distributed Multimedia Centre of Excellence (http://cs.ru.ac.za/coe)

3.1 Convex object extraction – This algorithm can be described as performing an **and** operation of a number of pictures of an object from different angles.

The process starts off with a cube of voxels[2] bounding the reconstruction. This bounding box is assumed to be large enough to encompass the object we wish to reconstruct. We have at our disposal images of the object at different angles and also the angle of rotation for each of the images (relative to the original image). Additional constraints involve background colour and the position of the camera with respect to the object. Specifying that the background in each of the images must be some predefined colour; this allows us to differentiate between pixels representing the object and the background ones. There must also be no vertical movement of the camera with respect to the object.

Having images with rotational information allows us to begin reconstructing a 3D model of the object. Different views of the object are applied to the bounding box. The reconstruction process takes place as follows:

```
For (each voxel in the bounding box)
   Voxel.flag = 1;    /* assume all voxels are to be kept */
For (each voxel in the bounding box)
{
      translate the voxel relative to (0,0,0); /* (0,0,0) is the centre of
                                                  the  bounding box */
      rotate by –θ;      /* θ =degree of current picture
                                rotation */
      reverse translate;
      if (picture[voxel.x,voxel.y].colour == backcolour)
         voxel.flag = 0;      /* ignore voxel */
}
```

The voxels remaining after this algorithm has been applied with each of the images represent the object.

The problem with convex object extraction is the word “convex”. This algorithm is unable to extract complete depth information from the different images and the result is thus a completely convex reconstruction of the object. Facial structure is a good example of a generally convex object with important concave features. The “cling-wrap” effect best describes any resulting object generated with this algorithm. A facial reconstruction would thus look like a face covered with a taught plastic layer. Figure 1 shows how the “cling-wrap” effect occurs. The shaded areas in the image represent the areas for which no information

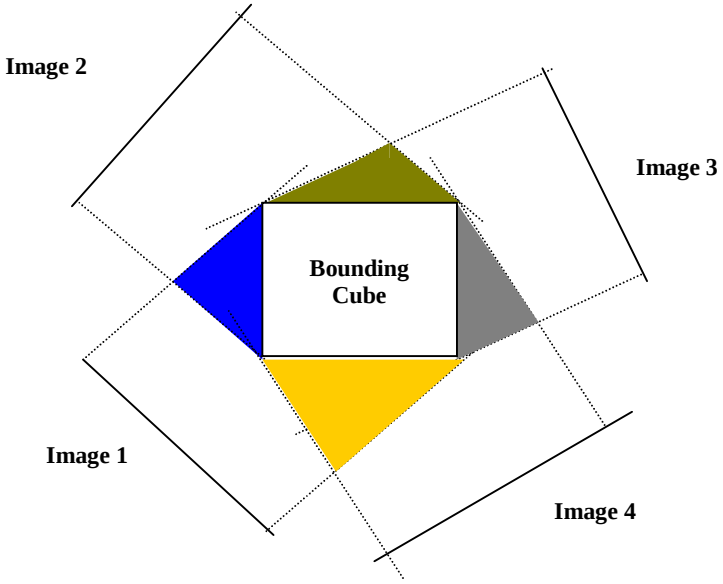

Fig 1: Convex Object Extraction in action. The shaded areas represent the regions where all concavities will be lost. The object represents the bounding box containing the object we want to reconstruct. As more and more images are used to carve out space from the bounding box, so the resulting object. The view depicted is a top-down one.

[2] The term voxel refers to a point with a x,y and z component i.e. a 3D point. It can also refer to a 1x1x1 cube in 3D space.

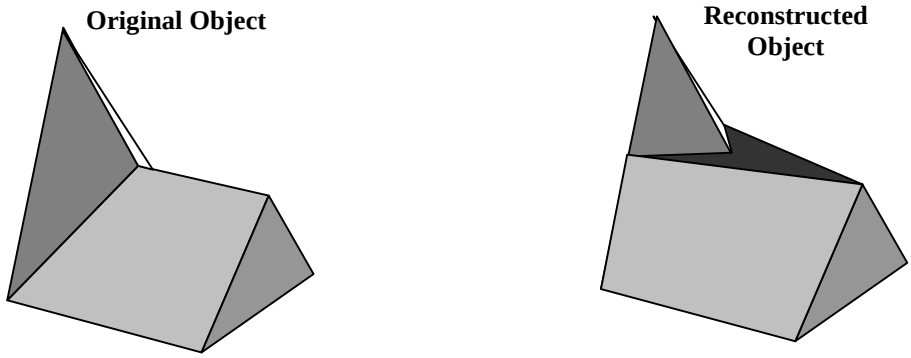

Fig 2: Reconstruction of the image on the left with the Convex Object Extraction algorithm leads to the object on the right. All concavity in the object has been removed.

about the object can be gathered. Figure 2 also an example of the resulting reconstruction this algorithm will generate given an object with a single concavity.

3.2 Normalised Cross Correlation(NCC) - Assuming a stereoscopic pair of images (a left and right image pair) of an object, depth information can be obtained if a pixel in the first image can be correctly matched to a corresponding pixel in the second. If a match is found, the Euclidean distance between the matched pair (often called the disparity of the pair) can be used as a representative depth value.

Simply, given a pixel (u_1,v_1) in the left image, a match (u_2,v_2) can be found in the right image. Once a match is found, the depth of that pixel can be estimated by simple triangulation. Correlation scores are computed by comparing a fixed window in the first image to a shifting window in the second image. Repeating this process for the whole image allows us to assign depth(z) values to each of the pixels in the left. This process allows one to generate a 3D view of the scene.

An implementation of NCC can be based on the following equation:

$$Score(m_1, m_2) = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1(u_1+i, v_1+j) - \overline{I_1(u_1, v_1)}\right]\times\left[I_2(u_2+i, v_2+j) - \overline{I_2(u_2, v_2)}\right]}{\sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1(u_1+i, v_1+j) - \overline{I_1(u_1, v_1)}\right]^2} \times \sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_2(u_2+i, v_2+j) - \overline{I_2(u_2, v_2)}\right]^2}}$$

Here, m and n indicate the size of the correlation window (also called the window size), m_1 and m_2 are the two pixels we wish to obtain a matching score for, and u_1, v_1, u_2, v_2 are the x and y coordinates of the pixels respectively. The components highlighted by arrows indicate the average intensity of the correlation windows in the first and second images in Figure 3. The conversion from RGB to luminance or intensity is:

$$I = 0.2122*R + 0.7013*G + 0.0865*B$$

where R, G and B are the red, green and blue components of each pixel and I represents the intensity conversion.

The *Score* function above returns a value between –1 (for a very weak match) and 1 (for a perfect match) depending on the strength of the match. The fact that a value is returned is useful in the case where one wants to ensure that a match is good, i.e. exceeds a certain minimum matching score. Specifying a minimum strength for a match means that it is possible that multiple matches for a pixel will be identified if the minimum strength is too low. Additional overhead must then be incurred to try and find the best match from these possibilities. A match is classified as good if the pixels neighboring (u_1,v_1) have matches occurring in the same vicinity as (u_2,v_2). A process is then followed which performs this check.

An NCC evaluation suffers from problems such as occlusions (where information in the first image is hidden because of rotation or translation) and false matches. Because the algorithm makes use of luminance

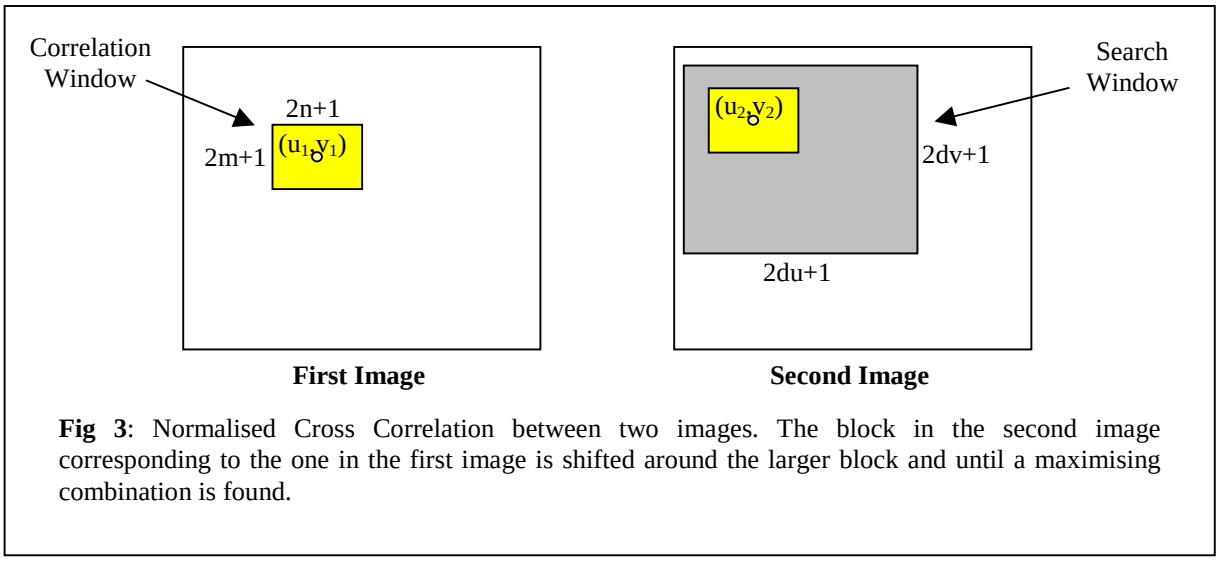

Fig 3: Normalised Cross Correlation between two images. The block in the second image corresponding to the one in the first image is shifted around the larger block and until a maximising combination is found.

ratios to determine the best possible match, false matches are often reported in cases where a pixel's colour matches the background colour.

An inherent problem with this algorithm is that as the window size increases, a point is finally reached where the accuracy of the implementation begins to decline, i.e. many matches are found. This is highlighted in Table 1, where up to a 8x8 window, the number of incorrect matches decline. With a 16x16 window size, firstly the number of non-matches (those pixels matches were not found for at that strength) increases sharply, and secondly the number of false matches begins to increase (from 2 at 8x8 to 3 at 16x16). The complexity of the algorithm is *O(n)*.

3.3 Epipolar Geometry and the Fundamental Matrix – NCC is very computationally expensive and defeats the requirements of real-time, with individual pixel matches exceeding 2 seconds on a P133 (window size = 8x8). Trying to match every pixel in the first image to a pixel in the second image is thus simply not feasible. This is where the epipolar geometry begins playing an important role. A stereoscopic pair of images are related to one another by the epipolar geometry. Given a point m_1 (u_1,v_1), its match m_2 (u_2,v_2) is constrained to lie on a line called the epipolar line of that point i.e. m_1 is related to m_2 by the following formula:

$$m_1^{\dagger} F m_2 = 0$$

where *F* is known as the fundamental matrix. This means that our search space reduces from a two-dimensional problem down to a one-dimensional case, where the search simply takes place along the epipolar line. This constraint arises because of the nature of the optical camera systems. Essentially, two projections of a single point in different images are related to one another by the fact that the projection onto each plane passes through the optical centre of the two camera. Given enough matches, we are able to

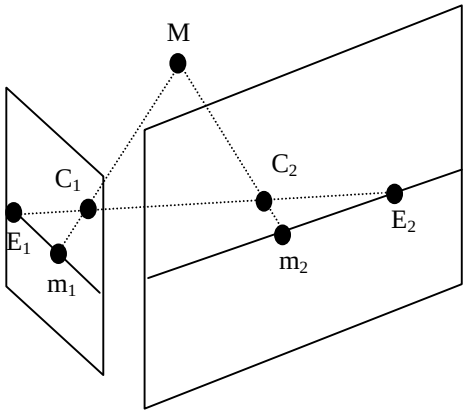

Fig 4: The relationship between two images. Given enough matches it is possible analytically determine the optical centre of the two images. M's depth thus depends on the intersection of the lines crossing through m_1,C_1,m_2 and C_2.

determine an analytical solution for F by simply plugging these values into the fundamental matrix. This is why it is very important to have a robust matching algorithm such as the one mentioned above.

The fundamental matrix essentially describes the rotation and translation of $\mathbf{m_1}$ to $\mathbf{m_2}$. The fundamental matrix has 7 degrees of freedom. Since no prior knowledge of camera properties such as focal length is assumed, the recovery of the fundamental matrix must be performed analytically. Using NCC above, we can calculate 8 random matches and thus determine a unique solution for **F**. Once the fundamental matrix has been determined, matches can be obtained quickly using the equation given in this section. This is because the epipolar equation constrains the seach for match $\mathbf{m_2}$ to the epipolar line formed by $\mathbf{m_2}$-$\mathbf{e_2}$ [Zhang et al, May 1994] Figure 4 illustrates how the epipolar geometry can be used to determine matches quickly. See [Zhang et al, May 1994] for a detailed description on how the epipolar geometry of a scene can be calculated

3.4 Convolution – To identify interesting points in the image pair, we have decided to apply some convolution filter to the source image. Applying an edge-detection convolution such as the Sobel or Laplacian, we are able to preserve the edge pixels. We then proceed to choose eight of the remaining filtered pixels and then try to find good matches for these pixels. This process simply improves the matching technique mentioned above. [Blythe, 1998]

In addition to the Sobel and Laplacian filters, we have implemented our own simple histogramming/polarisation convolution. Each pixel's red, green and blue component is checked against a threshold; if the threshold (in this case 128) is exceeded the relevant component is set to the maximum (in this case 255) or else the minimum (0). A result of this convolution applied to an image is illustrated in figure 7. This type of polarisation combined with a Laplacian convolution leads to an edge extraction algorithm which has virtually no noise (when applied to most real world images) and which generates edges represented by solid lines. This convolution combination thus generates good results for real-world images with much noise.

4. Approach

The implementation of the surface reconstruction algorithms mentioned above has been partially completed. The goal is not only to fulfil the requirements of a virtual videoconferencing system, but also to develop a full object/surface reconstruction system. Once the algorithm has been completed, work will begin with CoRgi.

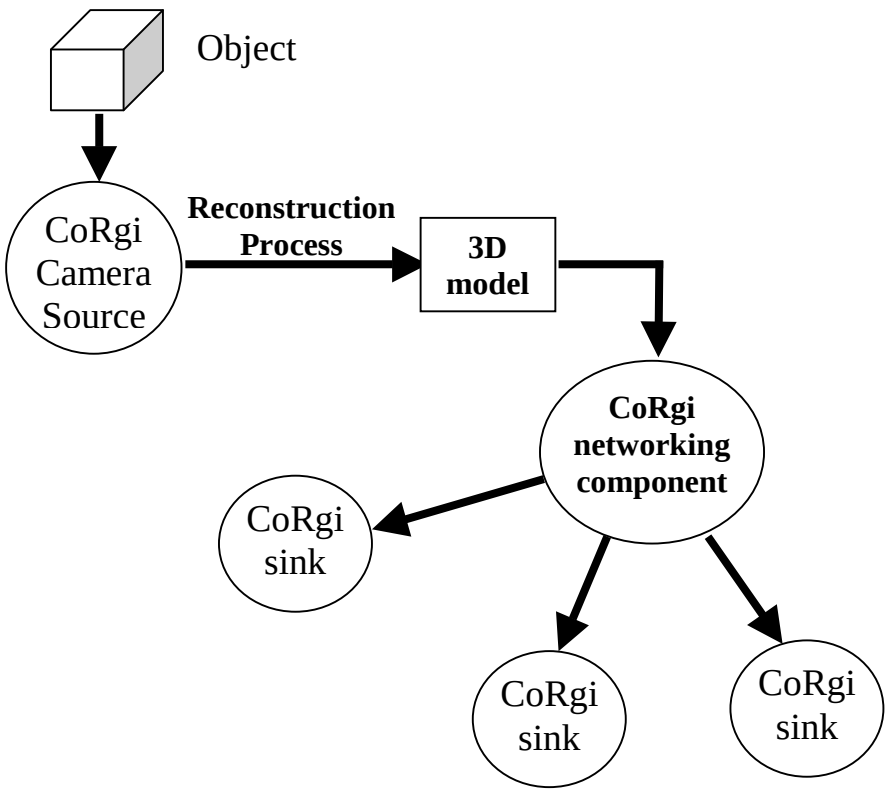

Fig 5: A graphical depiction of how CoRgi will be used as a platform for implementation of the virtual videoconferencing.

The CoRgi virtual reality system under development at Rhodes University is an object–oriented component based platform supporting the notion of sources, processors and sinks. The implemented reconstruction algorithm will be encapsulated in the form of a source, which will then, using a networking component, transmit these changes to the various sinks on the other side of the network for rendering. This process is illustrated in figure 5.

Issues such as using freeform deformation or simplified versions thereof to model natural facial changes will also be investigated. Freeform deformation will be investigated as a possibility of creating natural transitions between changes in the transmitted models.

A similar approach has been followed by [Escher and Thalmann, 1997], in which they make use of hardware which is prohibitively expensive, but the final result being a distributed system which both performs facial modeling on one SGI, but at the same time uses another machine to evaluate speech.. The result of our work will hopefully be a system that will work on "real" platforms. Once the implementation has been completed, virtual videoconferencing will be compared to standard videoconferencing and the feasibility of such a system classified. What follows below is a discussion of the CoRgi networking component. Since the networking component is still under development, much of what is mentioned below still has to be implemented.

4.1 CoRgi Networking Component

The main idea behind the network component is to provide a solid networking implementation without the need of traffic optimisation from the user or programmer. The networking component will "intelligently" decide on traffic shaping and traffic types by itself, and try to optimise the network usage while keeping as good a level of service to the clients as possible.

It consists of a number of components:

- channels: A Channel is a stream of network data, TCP and UDP channels are the currently implemented channel types.
- networkComponent: A networkComponent is one up the chain, it currently handles one channel, with an upcoming version handling any number of channels (up to 256) and stores information about the reliability wanted on the channel, and other attributes to provide useful information to the rest of the system about the network. If one links a networkComponent by itself, it will broadcast any information it is sent to all channels.
- networkServerComponent: The networkServer makes decisions on behalf of the various components it' s compiled into. This component will use the information given to it by the networkComponent, and the info given to it by the component, to decide on how to send the data, where to send it, and how much to send.

5. Results

The results we have obtained are very promising. Figure 6 illustrates NCC in action. The white dots on the right indicate matches for the triplet blue lines on the left. The and bottom blue lines indicate the size of the

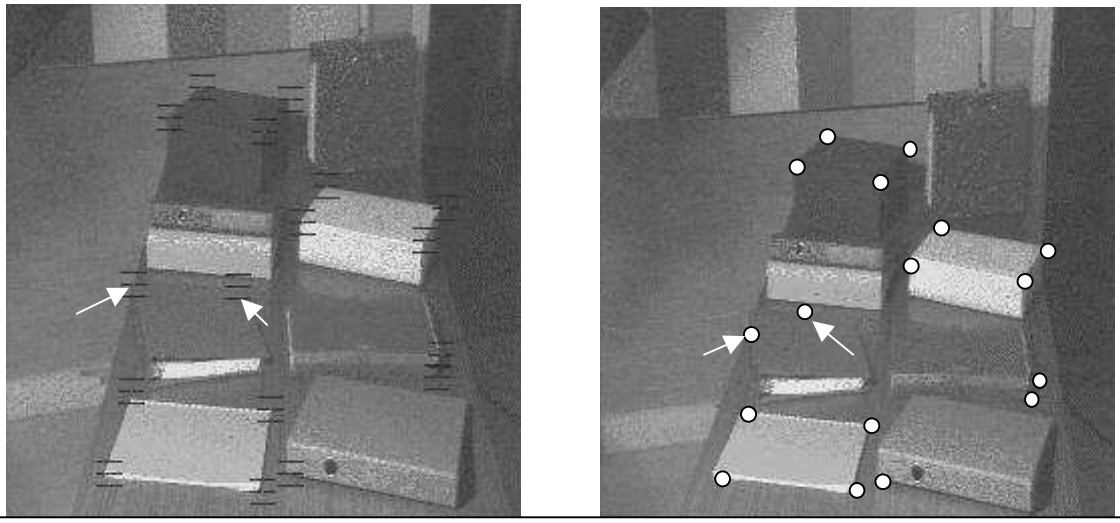

Fig 6: A stereoscopic pair of images of the same scene, courtesy of INRIA. The blue lines depicted in the left image indicate points for which matches have been found in the right image. The white arrows indicate false or incorrect matches. The window size in this case is 8x8.

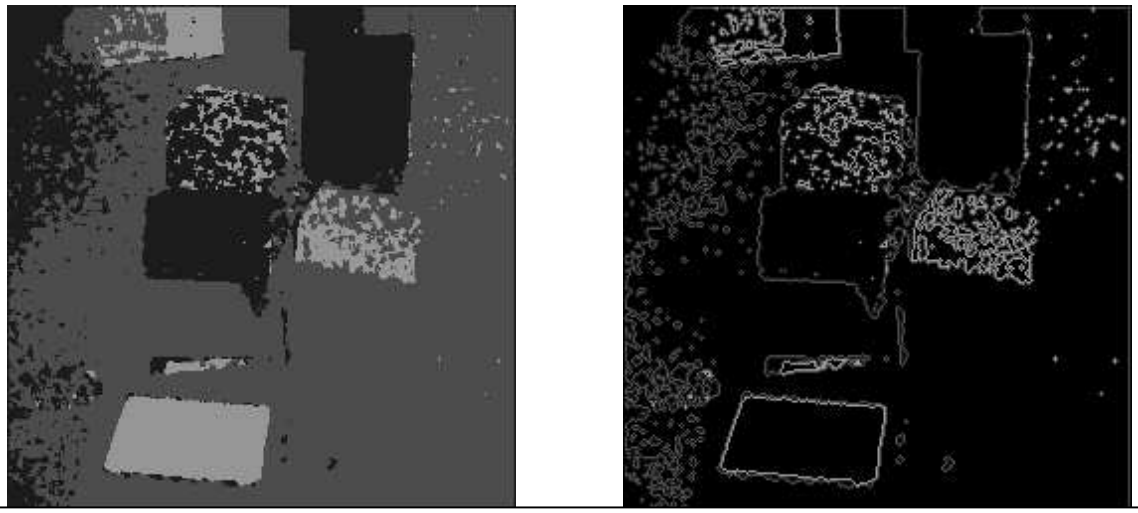

Fig 7: Convolutions applied to the left image in figure 6. The image on the left was first polarized and a mean convolution is then applied to the result. The Laplacian convolution is then applied to the left image, the result being discernable edges with constant lines. This image is subsequently used as input for the matching algorithm.

correlation window, while the central pixel in the middle line indicates the pixel we are searching for in the right image.

Size of window	No match	correct	incorrect	time
2x2	0	5	13	6. 24
4x4	1	14	3	22. 192
6x6	1	15	2	50. 385
8x8	1	15	2	88. 626
16x16	8	7	3	356. 428

Table 1: Results from applying NCC to a pair of images. The algorithm is asked to match 18 points with a minimum strength of 0. 8.

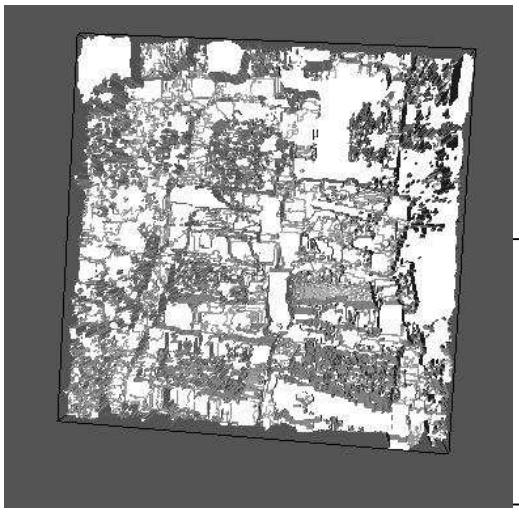

Fig 8: Normalised Cross Correlation of Figure 1. A match has been found for each pixel in the left image and the Euclidean distance between the matches used as a depth value. The flat areas (pointed to by the arrows) indicate unmatched pixels. The resulting image took 12 hours to process on 1 dual processor SGI Octane.

Figure 7 shows the same images as those displayed in Figure 6, but with convolutions applied to them. The left image illustrates out histogramming/polarisation algorithm, while the right image has had a Laplacian convolution applied to it. This image illustrates how this combination of convolutions generates edge information represented by solid lines.

Figure 8 illustrates the implementation of our "brute force method"; a match has been found for every pixel in the left image in the right image. This is done using NCC. The images used to do this are the books images from Figure 6. The left image from that figure is used as the source while the right image is used to find the matching pixels. The Euclidean distance for the matches is then mapped onto a depth value (z) for each of the pixels in the source image; if no match is found, the depth is set to –1. This usually occurs because the user has set the minimum matching score too high (in our case 0.8). The arrows in this Figure indicate such areas.

Table 1 illustrates some timings obtained when using NCC to match pixels.

Figure 9 illustrates the convex object extraction algorithm applied to a dolphin. In order to test this algorithm we have taken snapshots of a VRML dolphin model at different angles of rotation (10^o). The reconstructed dolphin appears below.

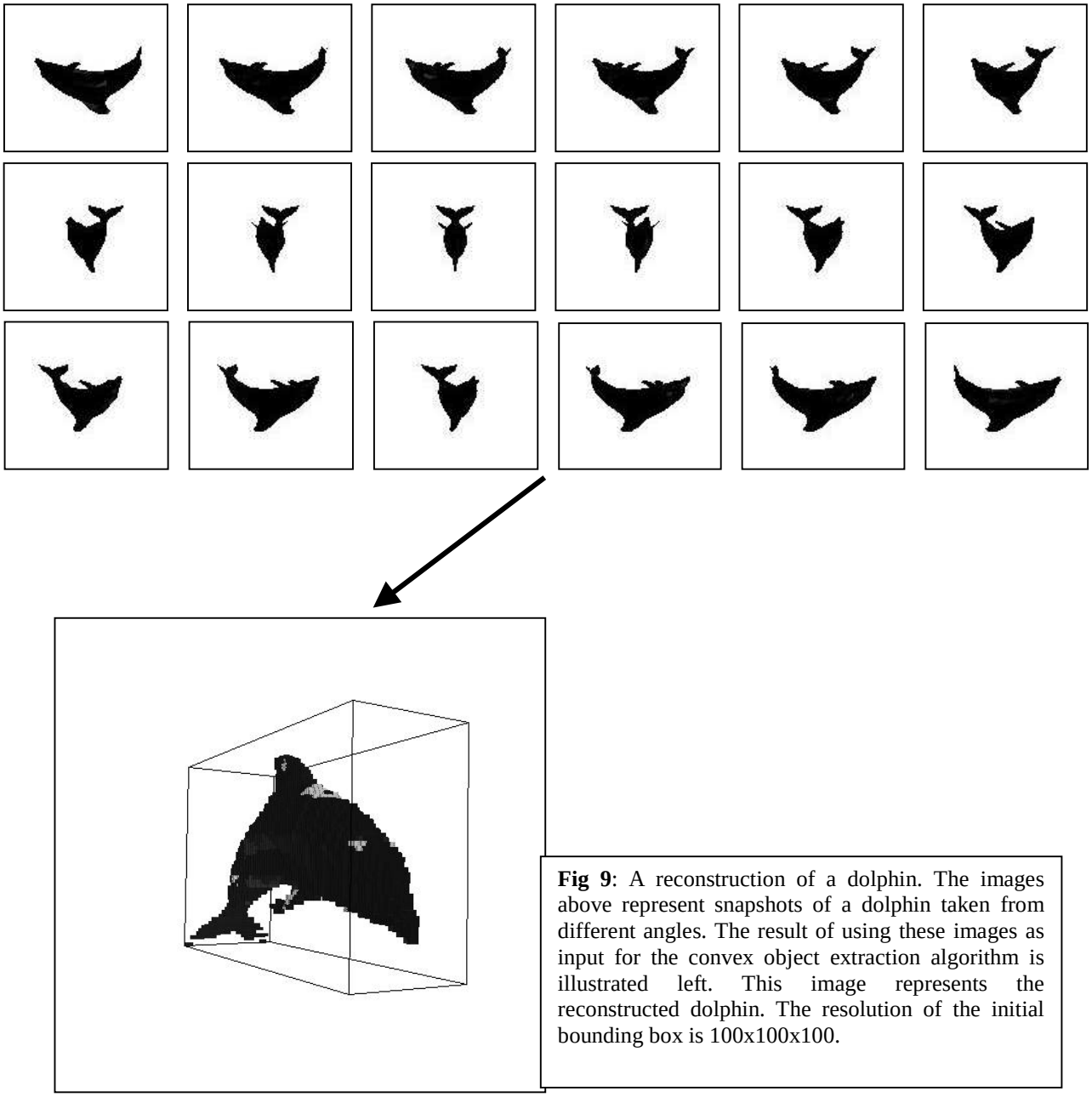

Fig 9: A reconstruction of a dolphin. The images above represent snapshots of a dolphin taken from different angles. The result of using these images as input for the convex object extraction algorithm is illustrated left. This image represents the reconstructed dolphin. The resolution of the initial bounding box is 100x100x100.

6. Conclusions

Two surface reconstruction techniques have been discussed, namely convex object extraction, and the epipolar geometry along with Normalised Cross Correlation.

Convex object extraction suffers from the fact that it is unable to properly determine depth information from a number of different angles. No concavities on the object's surface are modeled. The results indicate an algorithm which on the one hand is able to easily determine the general shape of any object, while on the other is not ideal for facial modeling (because of the human face's concave nature).

Normalised Cross Correlation (NCC) offers a more complex, yet more elegant solution to the reconstruction process. Pixels in a stereo pair of images are matched. Using the Euclidean distance between these matches, it is possible to determine the depth of that part of the object relative to the camera. The matching process is quite robust, yet very slow.

To match the interesting pixels in an image we apply convolutions (filters) to the images. The filter combination that seems to generate the best output is our own polarisation/histogramming algorithm

followed by a Laplacian convolution. The solid lines remaining in the convoluted image represent our "interesting pixels".

Epipolar geometry promises improvements in speed for NCC by reducing the search space the NCC algorithm uses from a two-dimensional block to a one-dimensional line. Because not all the information about the camera is available, we propose using an analytical method to find a solution to this problem.

7. Future Work

The aim of this project is to minimise bandwidth utilisation by reconstructing a human face and transmitting the 3D model of the face to the sink. This will be done with a virtual reality system. The goal will be to finally develop a real-time virtual videoconferencing system that will run on "real" machines.

8. References

[Niem and Buschmann,1994] W. Niem and R. Buschmann - Automatic Modelling of 3D Natural Objects from Multiple Views. *Institut for Theoretische Nachrichteotecknick und Informationsverbeitung. Universitat// Hannover, Hannover, Germany.* Hamburg 1994

[Zhang *et al*, May 1994] Z. Zhang, R. Deriche, O. Faguaras and Q. Luong - A Robust Technique for Matching Two Uncalibrated Images through the recovery of the Unknown Epipolar Geometry. *Inria Sophia Antipolis Research Institute.* May 1994

[Blythe, 1998] Blythe, D. ***Image Processing*** *http://www.sgi.com /Technology/OpenGL/advanced98/notes/node167.html*. 1998

[Eschet and Thalmann, 1997] Escher, M. and Thalmann, N. M. - ***Automatic 3D cloning and Real-Time Animation of a Human Face***. MIRALab, University of Geneva. 1997