2003

Vector flow model in video estimation and effects of network congestion in low bit-rate compression standards

Balaji Ramadoss
University of South Florida

VECTOR FLOW MODEL IN VIDEO ESTIMATION AND EFFECTS OF NETWORK CONGESTION IN LOW BIT-RATE COMPRESSION STANDARDS

by

BALAJI RAMADOSS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido A. Moreno, Ph.D.
James T. Leffew, Ph.D.
Wei Qian, Ph.D.

Date of Approval:
October 16, 2003

Keywords: h.263, compression, deformable super quadrics, video segmentation, medical imaging, network behavior

Dedication

This thesis is dedicated to my parents and teachers for their continuous support and motivation.

Acknowledgments

I would like to express my sincere thanks and appreciation to Dr. Wilfrido A. Moreno for his guidance and tutelage throughout this endeavor.

I would also like to sincerely thank Dr. James T. Leffew and Dr. Wei Qian for being available with time and useful advice as committee members.

My heart-felt personal thanks to Dr. Wilfrido A. Moreno for supporting and sponsoring this research venture.

Table of Contents

List of Tables

List of Figures

VECTOR FLOW MODEL FOR VIDEO ESTIMATION AND THE EFFECTS OF NETWORK CONGESTION IN LOW BIT-RATE COMPRESSION

Balaji Ramadoss

ABSTRACT

The use of digitized information is rapidly gaining acceptance in bio-medical applications. Video compression plays an important role in the archiving and transmission of different digital diagnostic modalities. The present scheme of video compression for low bit-rate networks is not suitable for medical video sequences. The instability is the result of block artifacts resulting from the block based DCT coefficient quantization. The possibility of applying deformable motion estimation techniques to make the video compression standard (H.263) more adaptable for bio-medial applications was studied in detail. The study on the network characteristics and the behavior of various congestion control mechanisms was used to analyze the complete characteristics of existing low bit-rate video compression algorithms.

The study was conducted in three phases. The first phase involved the implementation and study of the present H.263 compression standard and its associated limitations. The second phase dealt with the analysis of an external force for active contours, which was used to obtain estimates for deformable objects. The external force, which is termed **Gradient Vector Flow**, (GVF), was computed as a diffusion of the gradient vectors as

sociated with a gray-level or binary edge map derived from the image. The mathematical aspect of a multi-scale framework based on a medial representation

for the segmentation and shape characterization of anatomical objects in medical imagery was derived in detail. The medial representations were based on a hierarchical representation of linked figural models such as protrusions, indentations, neighboring figures and included figures, which represented solid regions and their boundaries. The third phase dealt with the vital parameters for effective video streaming over the internet in the bottleneck bandwidth, which gives the upper limit for the speed of data delivery from one end point to the other in a network. If a codec attempts to send data beyond this limit all packets above the limit will be lost. On the other hand, sending under this limit will clearly result in suboptimal video quality. During this phase the packet-drop-rate, (PDR), performance of TCP(1/2) was investigated in conjunction with a few representative TCP friendly Congestion Control Protocols, (CCP).

The CCPs were TCP(1/256), SQRT(1/256) and TFRC (256), with and without self clocking. The CCPs were studied when subjected to an abrupt reduction in the available bandwidth. Additionally, the investigation studied the effect on the drop rates of the TCP-Compatible algorithms by changing the queuing scheme from Random Early Detection (RED) to DropTail.

CHAPTER 1

INTRODUCTION

"We have to do the best we can. This is our sacred human responsibility."

--Albert Einstein

There is a growing demand for data sharing, offshore development and service calls for better and accurate means of multimedia compression and transmission techniques. The development of compression algorithms has mainly focused on the general demand for entertainment multimedia and other applications, which require cost reduction. The need for high quality, error free, video streaming for bio-medial applications has evolved rapidly and is market driven. The use of digitized information is rapidly gaining acceptance in bio-medical applications. Video compression plays an important role in the archiving and transmission of different digital diagnostic modalities. Various compression schemes such as JPEG and MPEG are usually applied to telephone conferencing, cable video transmission and other non-medical applications. The JPEG and MPEG compression schemes are not suitable for medical video sequences such as angiograms. The JPEG and MPEG compression schemes suffer from instability due to block artifacts resulting from the block based DCT coefficient quantization. The image

quality degrades severely with consecutive frame processing due to the accumulation of errors across frames.

Despite rapid progress in mass-storage density, processor speeds and digital communication system performance, demand for data storage capacity and data transmission bandwidth continues to outstrip the capabilities of available technologies. To appreciate the need for compression and coding of the individual signals that constitute the multimedia experience, Table 1.1 presents data for a few typical multimedia data types and the resulting uncompressed storage space requirements. The numbers indicate the qualitative transition from simple text to full-motion video data and demonstrate the need for compression. Additionally, the data presented in Table 1.1 clearly illustrates the need for large storage spaces for the image, audio and video signals.

Table 1.1: Multimedia Data Types and Uncompressed Storage Space Requirements

Multimedia Data	Size/Duration	Bits/Pixel or Bits/Sample	Uncompressed Size
A page of text	11" x 8.5"	Varying resolution	16-32 Kbits
Telephone quality speech	1 sec	8 bps	64 Kbits
Grayscale Image	512 x 512	8 bpp	2.1 Mbits
Color Image	512 x 512	24 bpp	6.29 Mbits
Medical Image	2048 x 1680	12 bpp	41.3 Mbits
SHD Image	2048 x 2048	24 bpp	100 Mbits
Full-motion Video	640 x 480, 10 sec	24 bpp	2.21 Gbits

This research presents the development of a compression scheme for bio-medical video sequence coding based on deformable motion estimation. The present state of technology, the prevalent video compression standard, (H.263), was implemented. In

addition, the possibility of applying deformable motion estimation techniques, termed Gradient Vector Flow, (GVF), was introduced to the video compression standard H.263. The introduction of GVF made the H.263 compression standard more adaptable for bio-medial applications, which were studied in detail. The study of the network characteristics and the behavior of various congestion control mechanisms were used to analyze the complete characteristics of the existing low bit-rate video compression algorithms. The importance of this study can be more highly appreciated if the effect of congestion on a network, specifically a low bit-rate network, is understood. Compressed video communication for biomedical applications was studied in great detail. The discussion of this research is divided into an analysis of the three elements central to the need for better compression for biomedical video. The three elements are depicted in Figure 1.1.

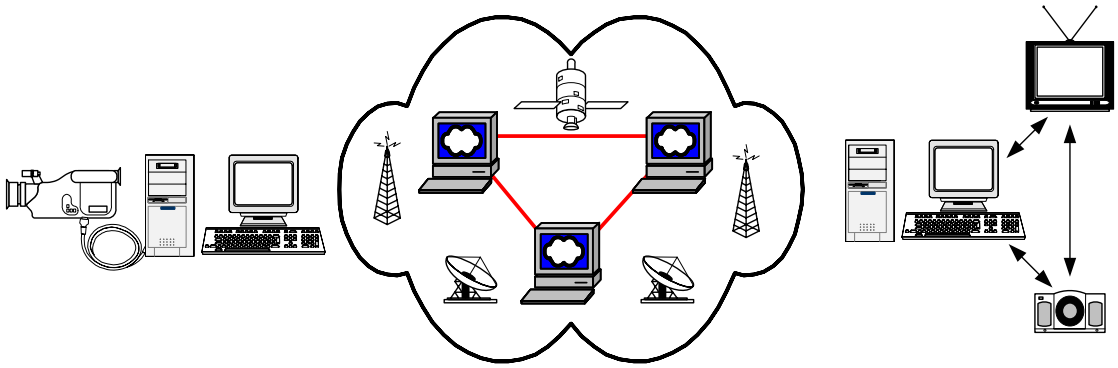

Figure 1.1: Overview of The Coder/Decoder and Network Under Study

1.1 Objectives

This research work was conducted in three phases:

- Definition and implementation of the compression scheme in H.263
- Implementation of Deformable Motion Estimation
- Behavior analysis of slowly responsive congestion control algorithms

The first part was the implementation of the existing H.263 algorithm, which included a measure achieved by its compression algorithm. The second phase dealt with the analysis of other means of motion estimation. During the second phase a Gradient Vector Flow, (GVF), model was developed for deformable motion estimation. The third phase dealt with the networks' behavior during congestion and its impact on network traffic.

CHAPTER 2

VIDEO COMPRESSION

MPEG, H.261, and H.263 are three closely related codecs for motion video. They are all international, non-proprietary standards. MPEG is an International Standard of the International Organization for Standardization, (ISO), while H.261 and H.263 are Recommendations of the International Telecommunications Union, (ITU). MPEG is intended for playback of movies from digital storage media while the other two codec is intended for teleconferencing.

MPEG is an acronym that stands for Moving Picture Experts Group, which is the name of the ISO committee that developed it. All three codecs are based on the discrete cosine transform, (DCT), predicted frames and motion estimation. A predicted frame is essentially a difference frame, which is the difference between the current input frame and the previously encoded and reconstructed frame. The difference should be small over most of the frame area except around the edges of moving objects and where new objects are introduced into the frame. A predicted frame is termed a P-frame; another name is inter-frame. A frame that is encoded independently of other frames is termed an intra-frame or I-frame.

Motion estimation estimates the translational motion of objects in the current frame relative to the previous frame. This allows the encoder to reduce the energy in the frame difference by moving pixels around in order to simulate object motion. Such action incurs a cost since the encoder must insert a small amount of motion information in the compressed data so that the decoder can reproduce the pixel shuffling exactly. Both MPEG and H.263 enhance prediction and motion estimation by using bi-directionally-predicted or B-frames. One can think of a B-frame as the average of two P-frames that use previous and future input frames as predictors for the current input frame. Generally speaking, a B-frame can be about one third the size of a P-frame. Obviously, the use of B frames implies out-of-order encoding since the encoder can only encode a B-frame after encoding the requisite previous and future frames. Note that a B-frame never predicts another B-frame; only I-frames and P-frames are used to predict B-frames.

One feature that H.261 and H.263 share that MPEG does not possess is support for variable frame rate within a video sequence. This is important in teleconferencing for two reasons. First, the bit rate in a teleconference can be very low, so an encoder must be able to lower the frame rate to maintain reasonable visual quality. Second, the encoder must be able to adjust to sudden changes in video content in real time without warning. For example, at a scene change the first compressed frame tends to be large. However, with variable-frame-rate encoding the encoder can encode the frame and then skip a few input frames before encoding the next frame. In fact, the human eye will not see motion in the video for a while after the scene change.

Historically, MPEG-1 was derived from H.261 and JPEG. H.263 is based on H.261 and MPEG-1 and adds some enhancements of its own. On the other hand, MPEG has some enhancements that are not present in H.261 or H.263. All other things being equal, at bit rates of 1 Mbps or above, MPEG-1 video will look better than the same content encoded with either H.261 or H.263. At the common rate of approximately 1.2 Mbps, with a frame resolution of 352x240 and a frame rate of 30 fps, the visual quality is comparable to or better than that from an analog VCR. When operated at the limit, these codecs can produce visual artifacts similar to those for JPEG. These include blocks and artifacts near the edges of objects. Such artifacts can be common in low bit rate teleconferencing and at unexpected scene changes. H.261 will be the most sensitive to such problems followed by H.263 and then MPEG. In low bit rate teleconferencing, 128 kbps and below, H.261 and H.263 video might look better than MPEG video since the first two codecs can vary the frame rate within a video sequence. H.263 can run at lower bit rates than H.261. H.263 can also run at higher bit rates and support larger frames, which are up to 4 times larger in each dimension. However, if an H.263 encoder does not use its optional modes, its output should be comparable to that from a similar H.261 encoder. The ability to reduce the blocks and artifacts are the most important characteristics of the algorithm developed during this research when it is used in biomedical applications.

2.1 H.263: An Example

The H.263 standard supports video compression for videoconferencing and video telephony applications. The H.263 standard is published by the International Telecommunications Union, (ITU). Figure 2.1 provides a macro-level representation of the H.263 system [1].

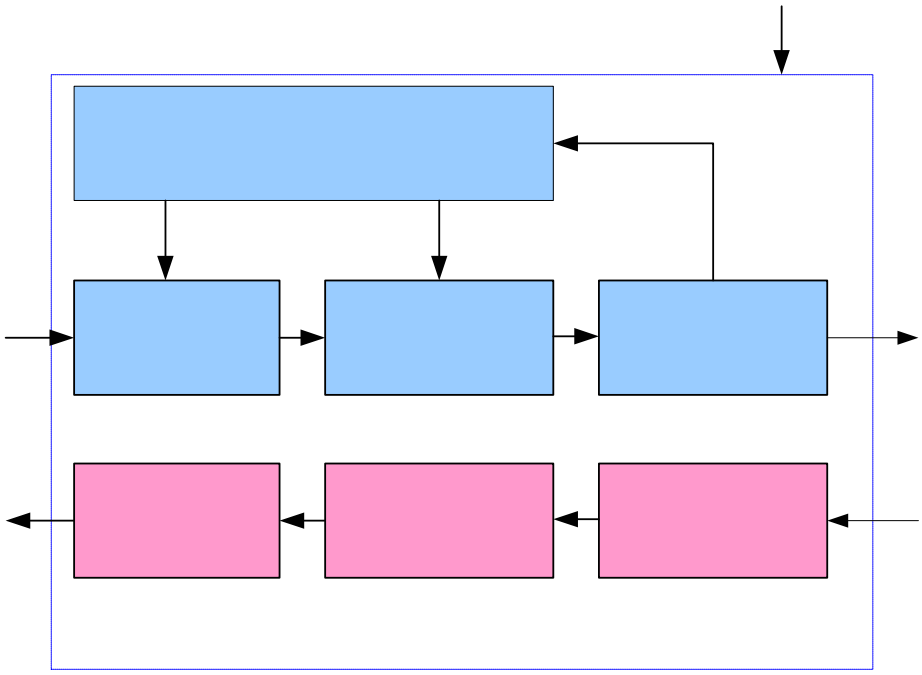

Figure 2.1: Video Coder/Decoder Module

Videoconferencing and video telephony have a wide range of applications that include:

- Desktop and room-based conferencing
- Video over the Internet and over telephone lines
- Surveillance and monitoring

- Telemedicine (medical consultation and diagnosis at a distance)
- Computer-based training and education

In each case video, and perhaps audio, information is transmitted over telecommunication links, which include networks, telephone lines, ISDN and radio. Video has a high "bandwidth" that requires many bytes of information per second. Therefore, these applications require video compression or video coding technology in order to reduce the bandwidth before transmission.

2.2 H.263 Encoder

A block diagram of the H.263 encoder is presented in Figure 2.2.

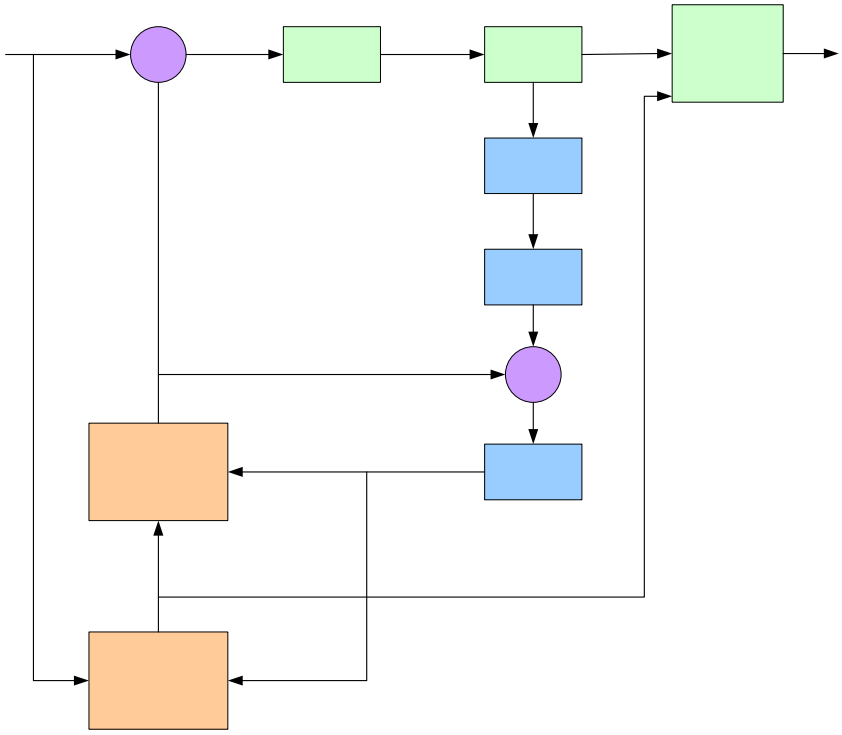

Figure 2.2: H.263 Encoder

2.2.1 Motion Estimation and Compensation

The first step in reducing the bandwidth is to subtract the previously transmitted frame from the current frame. This action leaves only the difference or residue for encoding and transmission. Therefore, areas of the frame that do not change, such as the background, are not encoded. Further reduction is achieved by attempting to estimate where areas of the previous frame will occur in the current frame and compensate for the movement, which is termed motion estimation and compensation. The motion estimation module compares each 16x16 pixel macro-block in the current frame with its surrounding area in the previous frame and attempts to find a match. The matching area is moved into the current macro-block position by the motion compensator module. Then the motion compensated macro-block is subtracted from the current macro-block. If the motion estimation and compensation process is efficient, the remaining or "residual" macro-block should only contain a small amount of information [2].

2.2.2 Discrete Cosine Transform (DCT)

The DCT transforms a block of pixel or residual values into a set of "spatial frequency" coefficients. This is analogous to transforming a time domain signal into a frequency domain signal using a Fast Fourier Transform. The DCT operates on a 2-dimensional block of pixels rather than a 1-dimensional signal and is particularly good at "compacting" the energy of the block of values into a small number of coefficients. This

means that only a few DCT coefficients are required to recreate a recognizable copy of the original block of pixels [3].

2.2.3 Quantization

In a typical block of pixels, most of the coefficients produced by the DCT are close to zero. The quantizer module reduces the precision of each coefficient so that the near-zero coefficients are set to zero and only a few significant non-zero coefficients are left. This action is performed practically by dividing each coefficient by an integer scale factor and truncating the result. It is important to realize that the quantizer "throws away" information [3].

2.2.4 Entropy Encoding

An entropy encoder, such as a Huffman encoder, replaces values that occur frequently with short binary codes and replaces values that occur infrequently with longer binary codes. The entropy encoding in H.263 is based on such a technique and is used to compress the quantized DCT coefficients. The result is a sequence of variable-length binary codes. These codes are combined with synchronization and control information such as the motion "vectors", which are required to reconstruct the motion-compensated reference frame in order to form the encoded H.263 bit stream.

2.2.5 Frame Store

The current frame must be stored so that it can be used as a reference when the next frame is encoded. Instead of simply copying the current frame into a store, the quantized coefficients are re-scaled; inverse transformed using an Inverse Discrete Cosine Transform and added to the motion-compensated reference block in order to create a reconstructed frame that is placed in a store termed the frame store. This ensures that the contents of the frame store in the encoder are identical to the contents of the frame store in the decoder. When the next frame is encoded, the motion estimator uses the contents of the frame store to determine the best matching area for motion compensation.

2.3 H.263 Decoder

A block diagram of the H.263 decoder is presented in Figure 2.3.

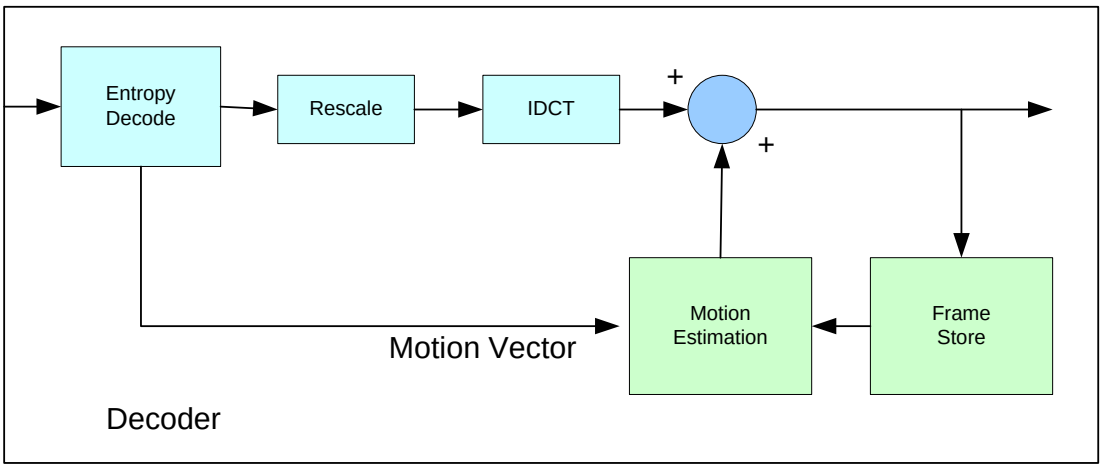

Figure 2.3: H.263 Decoder

2.3.1 Entropy Decoder

The variable-length codes that make up the H.263 bit stream are decoded in order to extract the coefficient values and motion vector information [3].

2.3.2 Rescale

Rescale is the "reverse" of quantization. During rescale the coefficients are multiplied by the same scaling factor that was used in the quantizer. However, because the quantizer discarded the fractional remainder, the rescaled coefficients are not identical to the original coefficients.

2.3.3 Inverse Discrete Cosine Transform (IDCT)

The IDCT reverses the DCT operation in order to create a block of samples. These samples typically correspond to the difference values that were produced by the motion compensator in the encoder.

2.3.4 Motion Compensation

The difference values are added to a reconstructed area from the previous frame. The motion vector information is used to pick the correct area, which is the same reference area that was used in the encoder. The result is a reconstruction of the original frame. The reconstructed frame will not be identical to the original because of the "lossy" quantization stage, which causes the image quality to be poorer than the original.

The reconstructed frame is placed in a frame store and it is used to motion-compensate the next received frame.

2.4 Block-Matching Motion Compensation

Predictive coding is widely used in video transmission, especially for low bit-rate coding. Typically, only a fraction of an image changes from frame to frame, which allows for a straightforward prediction from previous frames. Motion compensation is used as part of the predictive process. If an image sequence shows moving objects, then their motion within the scene can be measured and the information used to predict the content of frames later in the sequence [4].

2.4.1 Frame Based Block-Matching Motion Compensation

2.4.1.1 Fixed Size Block-Matching (FSBM)

The Fixed Size Block-Matching, (FSBM) technique was originally described by Jain and Jain [6]. The technique is easy to implement and widely adopted. Each image frame is divided into a fixed number of usually square blocks. For each block in the frame a search is made in the reference frame over an area of the image that allows for the maximum translation that can be used by the coder. The intent of the search is location of the best matching block that yields the least prediction error. Usually the search is conducted with the goal of minimizing either the mean square difference or the

mean absolute difference, which is easier to compute. Typical block sizes are of the order of 16x16 pixels and the maximum displacement might be plus or minus 64 pixels from a block's original position. Several search strategies are possible. Some kind of sampling mechanism is usually employed but the most straightforward approach is an exhaustive search. However, an exhaustive search is computationally demanding in terms of data throughput but algorithmically simple and relatively easy to implement in hardware. A good match during the search means that a good prediction can be made but the improvement in prediction must outweigh the cost of transmitting the motion vector. A good match requires that the whole block undergone the same translation. In addition, the block should not overlap objects in the image, including the background, that have different degrees of motion.

The choice of the block-size to use for motion compensation is always a compromise. Smaller and more numerous blocks can better represent complex motion than fewer number of large ones. This reduces the work and transmission costs of subsequent correction stages but increases the cost of the motion information itself. The problem has been investigated by Ribas-Corbera and Neuhoff [7]. They concluded that the choice of block-size could be affected not only by motion vector accuracy but also by other scene characteristics such as texture and inter-frame noise.

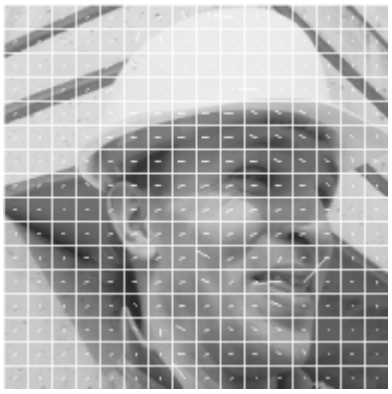

Figure 2.4: FSBM Example

The motion vectors resulting from FSBM are well correlated. Therefore, vector information can be coded differentially using variable length codes. This is performed in a number of codecs such as the ITU-T H.263 [3]. Variable length codes have also been proposed for the MPEG-4 video standard [8]. An example of the block structure generated is presented in Figure 2.4, which is a frame from the MPEG-4 test sequence known as "Foreman". Most noticeable is the stationary background that is represented by large numbers of blocks with very similar motion vectors. The short lines starting from the center of each block represent the motion vectors. Subsequently the motion vectors are variable length coded through the use of a differential 2D prediction mechanism.

2.4.1.2 Variable Size Block-Matching (VSBM)

Proposals have been presented that specify improvements to FSBM by varying the size of blocks in order to more accurately match moving areas. Such methods are known as variable size block matching, (VSBM), methods. Chan, Yu and Constantinides

have proposed a scheme that starts with relatively large blocks, which are then repeatedly divided, which is a so-called top down approach [9]. Whenever the best matching error for a block is above a specified threshold, the block is divided into four smaller blocks until the maximum number of blocks or locally minimum errors are obtained. The application of such top-down methods may generate block structures for an image that match real moving objects but it seems that an approach that more directly seeks out areas of uniform motion might be more effective.

The VSBM technique detects areas of common motion and groups them into variable sized blocks for use with a coding strategy based on the use of quad-trees [10]. Use of a quad-tree obviates the need to describe the size and position of each block explicitly. Quad-tree use requires only the tree description. The vectors for each block in the tree are identical in nature to those of the FSBM. Since the process is a grouping together of smaller blocks to form larger ones, it is generally regarded as a bottom-up technique. An example of the block structure generated is presented in Figure 2.5. Comparatively few large blocks represent the stationary background. However, the moving parts of the image are represented by smaller blocks and a larger number of motion vectors.

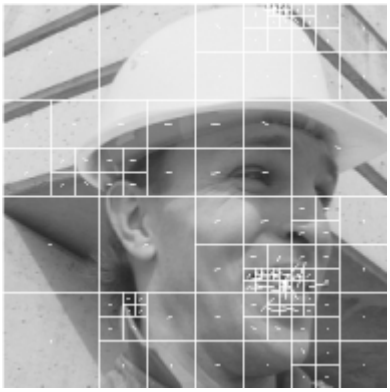

Figure 2.5: VSBM Example

For the same number of blocks per frame as FSBM, the VSBM method results in a smaller mean square error, (MSE), or better prediction. More significantly, for a similar level of MSE as FSBM, the VSBM technique can represent the inherent motion using fewer variable-sized blocks, which translates into the use of a reduced number of motion vectors. Subsequently the motion vectors are variable length coded using a quad-tree based 2d predictor mechanism [11].

Since frame-based VSBM results in a better estimate of "true" motion and more efficient coding of vector information one would expect that it can be applied to object-based systems with similar effects [12]. The expectation is true but there are two problems to overcome when using a basic block matching approach to find true motion.

- The first is the majority effect, where any small area of motion inside a block will simply be lost since the matching error for the block is determined by the majority of the block. This is an argument against the use of large block sizes.

Furthermore, a single block cannot effectively represent more than one motion. Therefore, there is always a trade-off between block size and error quality of match.

- The aperture problem is the second difficulty, which is associated with small block sizes. The fewer pixels there are to match, the more spurious matches there will be due to ambiguity. Additionally, there is little point in having the overhead of many small blocks if they all have the same vector but if they don't the vectors are unlikely to all be correct.

FSBM and VSBM were applied to the same frame of video data. Figure 2.6 presents the result of applying FSBM and Figure 2.7 presents the results of applying VSBM to the same video frame for the same quality prediction. While FSBM required 109 blocks, VSBM required only 44, which represents a saving of approximately 60 percent. Motion vectors were then variable length coded using a differential, object-based 2D prediction strategy [13].

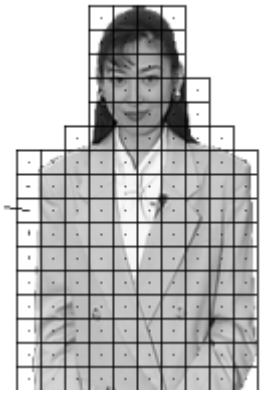

Figure 2.6: FSBM Block Structure

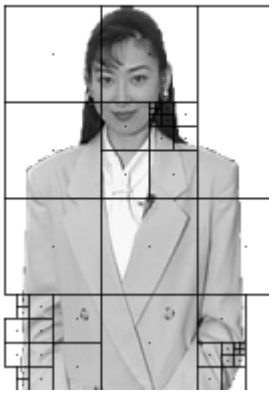

Figure 2.7: VSBM Block Structure

2.4.2 Object Based Block-Matching Motion Compensation

Evolving object-based video coding standards such as MPEG-4 permit arbitrarly-shaped objects to be encoded and decoded as separate video object planes, (VOPs). There are several motivating scenarios behind the use of objects:

- Where transmission bandwidth or decoder performance is limited, the user may be able to select some subset of all video objects, which are of particular interest.
- The user may wish to manipulate objects at the receiver such as a change in position, size and depth ordering, which may evolve strictly as a function of interest. It may be possible to replace the content of an object with material generated later or local to the receiver/display that can be used for enhanced visualization and "augmented reality".

2.5 Implementation

The H.263 algorithm was implemented and the results are tabulated in Table 2.1. A wide variety of video sequences were chosen for the analysis. The common input format chosen for the video is termed Common Intermediate Format, (CIF), and is based on a progressively scanned format with 360 x 288 pixels/frame at 30 frames/sec while the Quarter Common Intermediate Format, (QCIF), was 180 x 144. In addition, CIF compatibility was made optional but QCIF compatibility was mandatory. Therefore, all codecs had to be able to operate with QCIF. CIF is primarily for videoconferencing, while QCIF is suitable for a desktop videophone. QCIF was used, which was the mandatory requirement for the codec. Table 2.1 summarizes the results of various video formats being compressed with both QCIF and CIF formats utilized [5].

Table 2.1: Compressed Video Results

Sequence Name	File Name	Source Format	Number of Pictures	Size / KB	H.263 File / Size KB
Container	container.qcif	QCIF	300	11,138	87
Foreman	foreman.qcif	QCIF	400	14,850	658
News	news.qcif	QCIF	300	11,138	144
Silent	silent.qcif	QCIF	300	11,138	138
Mobile	mobile.cif	CIF	300	44,550	6391
Paris	paris.cif	CIF	1000	158,153	2624
Tempete	tempete.cif	CIF	260	38,610	2398

The H.263 Coder, decoder and the QCIF reader is presented in the appendix. Video communication over narrow-band channels such as the Internet, ISDN, modem or mobile communication suffers from a loss of image quality. The most annoying artifacts of block-based coders are the discontinuities at block boundaries. Many authors reduced this problem by applying smoothing operators to the block edges or by prediction of the image structure. Filtering decreases the blocking discontinuities but destroys image structures at the block edges [14]. When dealing with medical imaging application the threshold of allowed image/video loss and the quality is extremely critical. Therefore, this research investigated the feasibility of applying deformable motion and shape analysis to the H.263 algorithm for better prediction and as a method of avoiding the artifacts.

CHAPTER 3

DEFORMABLE MODEL AND SHAPE ANALYSIS

This section of the study analyses a new external force for active contours, which was used to estimate the deformable objects. The reason for this study was to analyze the feasibility of using some sort of estimation other than the motion estimation in order to reduce the block artifacts and increase the usability of the H.263 algorithm for medical applications. This study envisioned two different applications for medical video applications. The first application involved the ability to incorporate a vector flow model into the motion analysis part of video compression models. Such action involves taking the compressed video validation obtained by applying the vector flow model at both the transmitter and receiver end and comparing the numerical data for image distortion and loss. The second application involved the ability to develop a mathematical model for the deformable motion analysis and the numerical implementation of various forces and factors, which could be applied in medical imaging. The external force, which is called the **gradient vector flow** (GVF), is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image [15]. Snakes, or **active contours**, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image

data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. In this research, the focus was on parametric active contours. Parametric active contours synthesize parametric curves within an image domain and allow them to move toward the desired features, which are usually edges. Typically, the curves are drawn toward the edges by potential forces, which are defined to be the negative gradient of a potential function. Additional forces such as pressures, coupled with the potential forces comprise the **external forces.** There are also **internal forces** such as elasticity forces and bending forces that are designed to hold the curve together and to keep it from bending too much. Care must be maintained since pressures can push an active contour into boundary concavities but cannot be too strong or "weak" edges will be overwhelmed. How these forces are used in the mathematical model is also presented in the later part of the paper. Gradient vector flow, (GVF), fields are dense vector fields derived from images by minimizing certain energy functionals in a variational framework. The minimization was achieved by solving a pair of decoupled linear partial differential equations that diffuse the gradient vectors of a gray-level or binary edge map that is computed from the image.

The active contour that uses the GVF field as its external force is termed a **GVF snake**. The GVF snake is distinguished from nearly all previous snake formulations since its external forces cannot be written as the negative gradient of a potential function. As a result it could not be formulated using the standard energy minimization framework. Therefore, it was specified directly from a force balance condition.

To be more precise about the model, the research proposed a Bayesian approach incorporated from prior knowledge of the anatomical variations and the variation of the

imaging modalities. Following the deformable templates paradigm, exemplary templates were constructed to incorporate prior information about the geometry and shape of the anatomical objects under study. The infinite anatomical variability was accommodated in the Bayesian framework by defining probabilistic transformations on the templates. The segmentation problem in this paradigm involved finding the transformation **S,** of the template that maximized the posterior

$$p(S/data) \propto p(data/S)p(S) \tag{3.1}$$

where p(S) is the prior probability function that captures prior knowledge of the anatomy and its variability and $p(data/S)$ is the data likelihood function that captures the image data-to-geometry relationship [16]. For efficiency of implementation, the log-posterior given by

$$Logp(S/data) = Logp(data/S) + \log p(S) \tag{3.2}$$

was equivalently maximized up to an additive constant. The modeling approach adopted in this research for building exemplary templates and associated transformations was based on a multi-scale medial representation. The transformations defined in this framework are parameterized directly in terms of natural shape operations such as thickening and bending and their locations.

3.1 Parametric Snake Model

A traditional snake is a curve, given by

$$X(s) = [x(s).y(s)], s \in [0,1], \tag{3.3}$$

that moves through the spatial domain of the image in order to minimize the energy function

$$E = \int_0^1 \frac{1}{2}\left[\alpha\left|X'(s)\right|^2 + \beta\left|x''(s)\right|^2\right] + E_{ext}(X(s))ds \tag{3.4}$$

where α and β are weighting parameters that control snake's tension and rigidity respectively and X'(s) and X''(s) denote the first and the second order derivatives of X(s) with respect to the parameter "s". The external energy function, E_{ext}, is derived from the image so that it takes on its smaller values at the features of interest such as the boundaries. Given a gray-level image I(x,y), which is viewed as a continuous function of the position variables (x,y), the typical external energies were designed to lead an active contour toward step edges given by

$$\begin{aligned} E_{ext}^{(1)}(x,y) &= -\left|\nabla I(x.y)\right|^2 \\ E_{ext}^{(2)}(x,y) &= -\left|\nabla\left[G_\sigma(x,y) * I(x.y)\right]\right|^2 \end{aligned} \tag{3.5}$$

where $\left[G_\sigma(x,y)\right]$ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator. If the image is a line drawing that is black on white then the appropriate external energies include

$$\begin{aligned} E_{ext}^{(3)}(x.y) &= I(x,y) \\ E_{ext}^{(4)}(x,y) &= G_0(x,y) * I(x,y) \end{aligned} . \tag{3.6}$$

The definitions in equation (3.5) and equation (3.6) show that larger σ's will cause the boundaries to become blurry. However, large σ's are often necessary, in order to increase the capture range of the active contour.

A snake that minimizes E must satisfy the Euler equation

$$\alpha X''(s) - \beta X''''(s) - \nabla E_{ext} = 0 . \tag{3.7}$$

This can be viewed as a force balance equation given by

$$F_{\text{int}} + F_{ext}^{(p)} = 0 \tag{3.8}$$

where

$$F_{\text{int}} = \alpha x''(s) - \beta x''''(s) \quad \text{and} \quad F_{ext}^{(p)} = -\nabla E_{ext} \, . \tag{3.9}$$

The internal force, F_{int}, discourages stretching and bending while the external potential force, $F_{ext}^{(p)}$,pulls the snake toward the edges of the desired image [17].

3.2 Example and Behavior of Traditional Snakes

An example of the behavior of a traditional snake is presented in this section. Figure 3.1 presents a 64 x 64-pixel line drawing of a U-shaped object, shown in gray, that has a boundary concavity at the top.

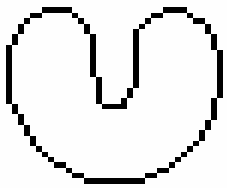

Figure 3.1: U Shaped Object

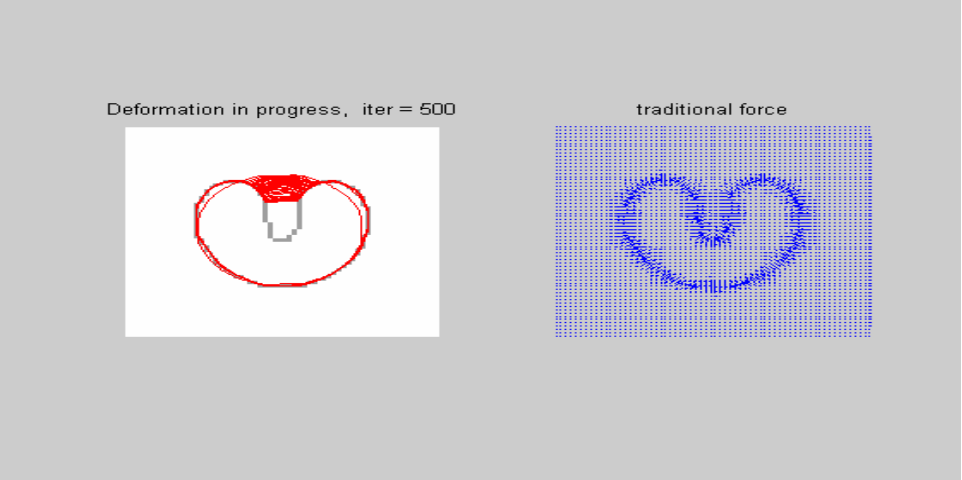

Figure 3.2: Iterations and Convergence

Figure 3.2 presents a sequence of curves, shown in red, that depict the iterative progression of the solution by a traditional snake ($\alpha = 0.6, \beta = 0.0$) initialized outside the object but within the capture range of the potential force field. The potential force field is given by

$$F_{ext}^{(p)} = -\nabla E_{ext}^{(4)} \tag{3.10}$$

where the pixels, shown in blue, are also presented in Figure 3.2. Note that the final solution in Figure 3.2 solves the Euler equations of the snake formulation but remains split across the concave region [18]. The reason for the poor convergence of this snake is revealed in a close-up of the external force field within the boundary concavity. Although the external forces correctly point toward the object boundary, within the boundary concavity the forces point horizontally in opposite directions. Therefore, the active contour is pulled apart toward each of the "fingers" of the U-shape but is not made to progress downward into the concavity. There is no choice of the α, β parameters that will correct this problem. Another key problem with traditional snake formulations is the

problem of limited capture range, which can be understood by examining Figure 3.2. In Figure3.2 the magnitude of the external forces dies out quite rapidly away from the object boundary. Increasing α in

$$E_{ext}^{(4)}(x,y) = G_0(x,y) * I(x,y) \tag{3.11}$$

will increase this range but the boundary localization will become less accurate and distinct, which will ultimately obliterate the concavity itself when σ becomes too large. In Figure 3.2 the snake fails to converge exactly on the object and has a lot of disparity in finding the edges. These are the usual problems found with the traditional snake method. Therefore, the distance potential forces do not solve the problem of convergence to boundary concavities.

Chapter 4

GRADIENT VECTOR FLOW

4.1 Snake: Introduction

The overall approach started with the force balance condition

$$(F_{\text{int}} + F_{ext}^{(p)} = 0) \tag{4.1}$$

as a starting point for designing a snake. A new static external force field

$$F_{ext}^{g} = v(x, y), \tag{4.2}$$

which was termed the **gradient vector flow,** (GVF), field was developed. In order to obtain the corresponding dynamic snake equation, the potential force $-\nabla E_{ext}$ was replaced in

$$X_t(s,t) = \alpha X''(s,t) - \beta x''''(s,t) - \nabla E_{ext} \tag{4.3}$$

with v(x,y), which yielded

$$X_t(s,t) = \alpha X''(s,t) - \beta X''''(s,t) + V . \tag{4.4}$$

The parametric curve that solves equation (4.4) is termed a **GVF snake**. Equation (4.4) was solved numerically by through a process of discretization and iteration, which is a process identical to that utilized to produce a traditional snake. Although the final configuration of a GVF snake satisfied the force-balance equation, equation (4.1), the

equation does not, in general, represent the Euler equations of the energy minimization problem where

$$E = \int_0^1 \frac{1}{2}\left[\alpha\left|X'(s)\right|^2 + \beta\left|x''(s)\right|^2\right] + E_{ext}(X(s))ds \tag{4.5}$$

since v(x,y) will not, in general, be a non-rotational field. However, the resulting loss of the optimality property was well-compensated by the significantly improved performance of the GVF snake [19].

4.1.1 Edge Map

An **edge map** was derived from the image *I(x,y)*, which had the property that it was larger near the image edges. Any gray-level or binary edge map could have been used. For example,

$$f(x,y) = -E_{ext}^{(i)}(x,y) \tag{4.6}$$

where i=1, 2, 3, or 4 could have been used. Three general properties of edge maps are important in the present context. First, the gradient of an edge map, ∇f , has vectors pointing toward the edges and become normal to the edges at the edges. Second, these vectors generally have large magnitudes only in the immediate vicinity of the edges. Third, in homogeneous regions, where I(x,y) is nearly constant, ∇f is nearly zero.

These properties affect the behavior of a traditional snake when the gradient of an edge map is used as an external force. The first property causes a snake initialized close to the edge to converge to a stable configuration near the edge, which is a highly desirable property. However, the second property, in general, causes the capture range to be very small. The third property causes homogeneous regions to have no external forces

whatsoever. The second and third properties are undesirable. Therefore, the approach adopted in this research was to keep the highly desirable property of the gradients near the edges but to extend the gradient map farther away from the edges and into homogeneous regions using a computational diffusion process. This approach produced an important benefit. The inherent competition of the diffusion process created vectors that pointed into boundary concavities.

4.2 Mathematical Model

A gradient vector flow field was defined as the vector field

$$V(x,y) = [u(x,y).v(x,y)] \tag{4.7}$$

with the purpose of minimizing the energy functional

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 \, dxdy \,. \tag{4.8}$$

This variational formulation of ε provided the benefit of making the result smooth when there was no data. In particular, when $|\nabla f|$ is small the energy is dominated by the sum of the squares of the partial derivatives of the vector field, which produces a slowly varying field. However, when $|\nabla f|$ is large the second term of the integrand dominates the integrand, which can then be easily be minimized by setting v equal to $|\nabla f|$.

Minimization of the integrand in ε produces the desired effect of keeping v approximately equal to the gradient of the edge map when it is large but forces the field to be slowly varying in homogeneous regions. The parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand. The parameter μ should be set according to the amount of noise present in the image. As the

noise increases μ increases. The smoothing term, which is the first term within the integrand of ε, where ε is repeated as equation (4.9) for convenience,

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 \, dxdy \tag{4.9}$$

is the same term as found in the classical formulation of optical flow. It has recently been shown that this term produces an equal penalty on the divergence and curl of the vector field. Therefore, the vector field resulting from this minimization can be expected to be neither entirely non-rotational nor entirely solenoid. Using the **calculus of variations**, it can be shown that the GVF field can be found by solving the following Euler equations. The Euler equations are given by

$$\begin{aligned} \mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \\ \mu\nabla^2 v - (v - f_x)(f_x^2 + f_y^2) = 0 \end{aligned} \tag{4.10}$$

where ∇^2 is the Laplacian operator. The Euler equations provided the motivation for the GVF formulation utilized in this research. In a homogeneous region, where *I(x, y)* is constant, the second term in each equation is zero since the gradient of *f(x, y)* is zero. Therefore, within such a region, *u* and *v* are each determined by Laplace's equation and the resulting GVF field is interpolated from the region's boundary, which indicates the existence of a kind of competition among the boundary vectors. This explains why GVF yields vectors that point into boundary concavities [20].

4.3 Numerical Implementation

The Euler equations can be solved if *u* and *v* are treated as functions of time. Solving the Euler equations for *u* and *v* yields

$$\begin{aligned} u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - \left[u(x,y,t) - f_x(x,y)\right]\left[f_x(x,y)^2 + f_y(x,y)^2\right] \\ v_t(x,y,t) &= \mu\nabla^2 v(x,y,t) - \left[v(x,y,t) - f_y(x,y)\right]\left[f_x(x,y)^2 + f_y(x,y)^2\right] \end{aligned} . \quad (4.11)$$

The steady-state solutions of these linear parabolic equations are the desired solution of the Euler equations, equation (4.10). The expressions in equation (4.11) are decoupled. Therefore, they can be solved as separate scalar partial differential equations in *u* and *v*. The expressions in equation (4.11) are known as generalized diffusion equations. For convenience, they are rewritten as presented in equation (4.12) as

$$\begin{aligned} u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - b(x,y)u(x,y,t) + c^1(x,y) \\ v_t(x,y,t) &= \mu\nabla^2 v(x,y,t) - b(x,y)v(x,y,t) + c^2(x,y) \end{aligned} \quad (4.12)$$

where

$$\begin{aligned} b(x,y) &= f_x(x,y)^2 + f_y(x,y)^2 \\ c^1(x,y) &= b(x,y) f_x(x,y) \\ c^2(x,y) &= b(x,y) f_y(x,y) \end{aligned} . \quad (4.13)$$

Any digital image gradient operator can be used to calculate f_x and f_y. In the examples presented in this thesis a simple central difference was used. The coefficients b(x,y), $c^1(x,y)$ and $c^2(x,y)$ can be computed and fixed for the entire iterative process. In order to set up the iterative solution, let the indices i, j and n correspond to x, y, and t, respectively and let the spacing between pixels be Δx and Δy and the time step for each iteration be Δt . The required partial derivatives can be approximated as

$$
\begin{aligned}
u_t &= \frac{1}{\nabla t}(u_{i,j}^{n+1} - u_{i,j}^{n}) \\
v_t &= \frac{1}{\nabla t}(v_{i,j}^{n+1} - v_{i,j}^{n}) \\
\nabla^2 u &= \frac{1}{\nabla x \nabla y}(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} + 4u_{i,j}) \\
\nabla^2 v &= \frac{1}{\nabla x \nabla y}(v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i,j-1} + 4v_{i,j})
\end{aligned}
. \tag{4.14}
$$

Substituting these approximations into equation (4.13) yields an iterative solution for the GVF that is given by

$$
\begin{aligned}
u_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)u_{i,j}^{n} + r(u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}) + c_{i,j}^{1}\Delta t \\
v_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)v_{i,j}^{n} + r(v_{i+1,j}^{n} + v_{i,j+1}^{n} + v_{i-1,j}^{n} + v_{i,j-1}^{n} - 4v_{i,j}^{n}) + c_{i,j}^{2}\Delta t
\end{aligned}
\tag{4.15}
$$

where

$$
r = \frac{\mu \Delta t}{\Delta x \Delta y}. \tag{4.16}
$$

Convergence of the above iterative process is guaranteed since it is a standard procedure from the theory of numerical methods. Provided that b(x,y), $c^1(x,y)$ and $c^2(x,y)$ are bounded, the results in equation (4.15) are stable whenever the Courant–Friedrichs–Lewy step-size restriction of $r \le 1/4$ is maintained. Since Δx and Δy and μ are normally fixed, use of the definition of r in equation (4.16) imposes a restriction on the time-step must be maintained in order to guarantee convergence of the GVF. The restriction on the time step demands that

$$
\Delta t \le \frac{\Delta x \Delta y}{4\mu}. \tag{4.17}
$$

4.4 MATLAB Results

The GVF field was simulated in MATLAB. Some predetermined shapes were used as test diagrams. Figure 4.1 presents the test images and the traditional forces acting on them.

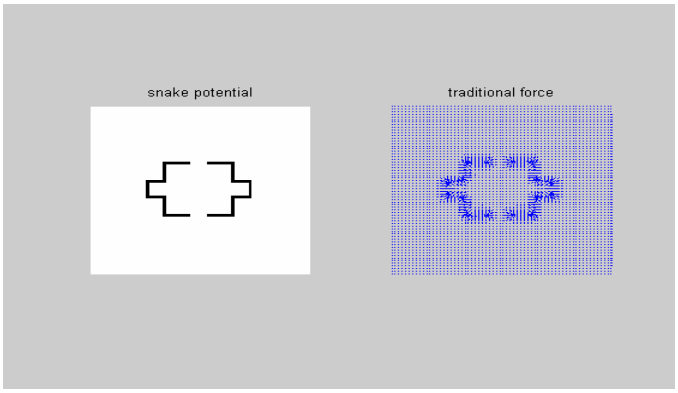

Figure 4.1: Test Image With its Traditional Forces

Iterations for computing the object shape with the traditional force method performed on the test image. Figure 4.2 illustrates the process followed in identifying the deformed test image.

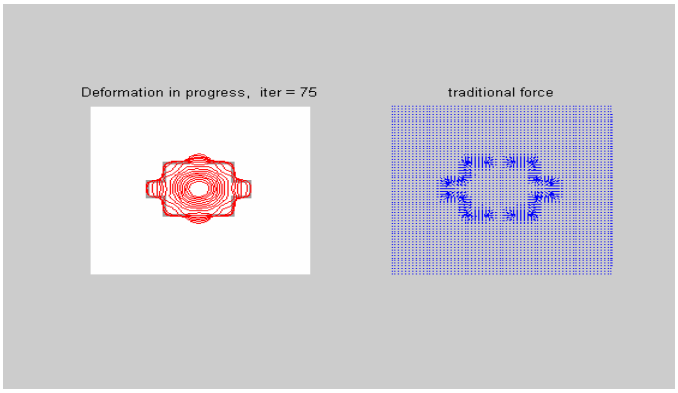

Figure 4.2: Iterations

The experimental results obtained by use of the GVF method on the U shaped object is presented in the Figure 4.3.

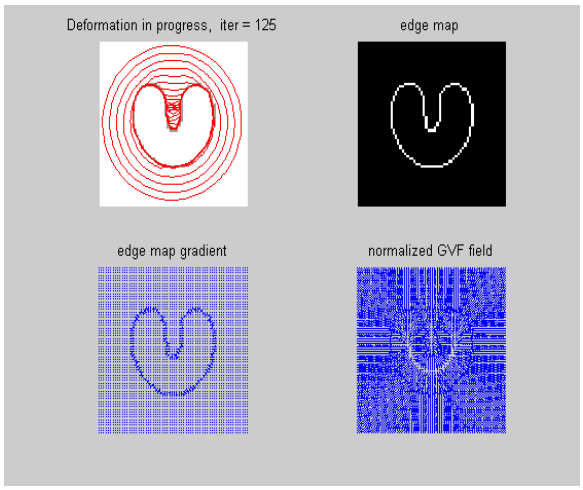

Figure 4.3: U Shaped Test Object Undergoing GVF Iterations

The improved GVF iteration method with different parameters was applied to the block image. The results obtained are presented in Figure 4.4.

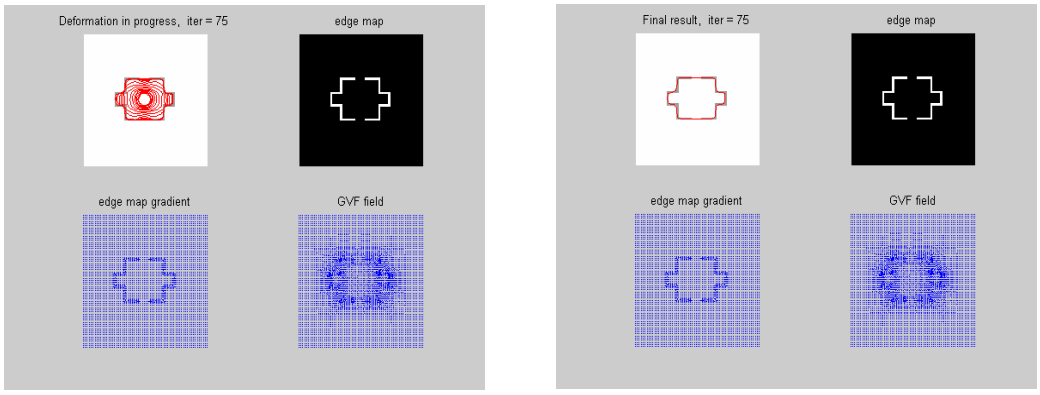

Figure 4.4: Iterations on the Block Test Image and its Convergence

CHAPTER 5

MEDICAL APPLICATIONS

5.1 Deformable Super Quadrics

Deformable super quadrics are three-dimensional, (3D), constructs that are used for describing closed-surface shapes. They can be used for both 3D graphics and computer vision. Super quadrics use one consistent representation to describe many different objects such as cubes, spheres and head shapes. The super quadric can be fitted to an existing set of data points, which unifies geometric modeling and physical modeling. They also provide for quick point-to-point distance and surface distance calculations. Super quadrics are represented mathematically by

$$\begin{aligned} h(\omega) &= \begin{bmatrix} \cos^{e2}(\omega) \\ \sin^{e2}(\omega) \end{bmatrix}; -\pi \le \omega \le \pi \\ m(\eta) &= \begin{bmatrix} \cos^{e1}(\omega) \\ \sin^{e1}(\omega) \end{bmatrix}; -\pi/2 \le \eta \le \pi/2 \end{aligned} \quad . \tag{5.1}$$

The functions presented in equation (5.10) are two-dimensional, (2D), parametric curves. The function *h* is a horizontal curve whose values traverse a full circle. The function *m* modulates *h* and yields values that traverse a half circle. The *e1* parameter changes the

scale of *h* while the parameter *e2* raises/lowers *h*. The parameters *e1* and *e2* control the "squareness" of the surface.

The basic functions, given in equation (5.1) yield a representation for a 3D surface that is given by the cross product of *m* and that is given by

$$r = m \otimes h \,. \tag{5.2}$$

The cross product yields

$$r(\eta, \omega) = \begin{bmatrix} r \\ s \\ t \end{bmatrix} = \begin{bmatrix} \cos^{e1}(\omega)\cos^{e2}(\omega) \\ \cos^{e1}(\omega)\sin^{e2}(\omega) \\ \sin^{e1}(\eta) \end{bmatrix} \tag{5.3}$$

where the ω and η parameters can be eliminated by using the relation

$$\sin^2 r + \cos^2 r = 1 \,. \tag{5.3}$$

The super quadric can be transformed through the use of

$$X = R\, D_m\, r + b \tag{5.4}$$

where

- X is the 3x1 resulting point,
- R is a 3x3 "modal deformation matrix",
- r is the 3x1 point to be transformed,
- b is a 3x1 translation vector.

The transformation allows placement of the super quadric anywhere in space with many possible shapes. The Modal Deformation Matrix is given by

$$D_m = \begin{bmatrix} d_{00} & d_{01} & d_{02} \\ d_{10} & d_{11} & d_{12} \\ d_{20} & d_{21} & d_{22} \end{bmatrix} \tag{5.5}$$

where each entry is a polynomial. The Surface Normal to a super quadric is given by

$$\begin{aligned} nh(\omega) &= \begin{bmatrix} \cos^{2-e2}(\omega) \\ \sin^{2-e2}(\omega) \end{bmatrix} \\ nm(\eta) &= \begin{bmatrix} \cos^{2-e1}(\omega) \\ \sin^{2-e1}(\omega) \end{bmatrix} \end{aligned} \tag{5.6}$$

The cross product of equation (5.6) is given by

$$n(\eta,\omega) = \begin{bmatrix} 1/r \cos^2(\eta)\cos^2(\omega) \\ 1/s \cos^2(\eta)\sin^2(\omega) \\ 1/t \sin^2(\eta) \end{bmatrix}. \tag{5.7}$$

Using a "Finite Element Model", instead of a continuous surface, forces the use of a discrete set of points across the surface. The physical forces acting on the points are modeled by

$$\text{Global coordinates: } U = [\, U \; V \; W \,]^T \tag{5.8}$$

and

$$\text{Local coordinates: } u = [u \; v \; w]^T. \tag{5.9}$$

In addition to the super quadric, the differential equations for describing the associated physics are given by

$$\text{Body forces: } f^B = \begin{bmatrix} f_x^B & f_y^B & f_z^B \end{bmatrix}^T, \tag{5.10}$$

$$\text{Surface forces: } f^s = \begin{bmatrix} f_x^s & f_y^s & f_z^s \end{bmatrix}^T, \tag{5.11}$$

Attachment forces: $f^l = \begin{bmatrix} f_x^l & f_y^l & f_z^l \end{bmatrix}^T$ (5.12)

Strains: $\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & \varepsilon_{xy} & \varepsilon_{xz} & \varepsilon_{yz} \end{bmatrix}^T$, (5.13)

and

Stresses: $\tau = \begin{bmatrix} \tau_{xx} & \tau_{yy} & \tau_{zz} & \tau_{xy} & \tau_{yz} & \tau_{xz} \end{bmatrix}^T$. (5.14)

Stiffness modeling is possible through the use of the functions

$$\begin{aligned} R^B &= \sum_M \int_{v^{(m)}} H^{(m)T} f^{(m)B} dv^{(m)} \\ R^s &= \sum_M \int_{v^{(m)}} H^{(m)T} f^{(m)s} dv^{(m)} \\ R^I &= \sum_M \int_{v^{(m)}} H^{(m)T} f^{(m)I} dv^{(m)} \\ R^c &= F \end{aligned} \quad . \tag{5.15}$$

Inertia and Damping Modeling was performed so that the final solution for the deformation energy equation would be given by

$$F_I(t) + F_D(t) + F_E(t) = R(t), \tag{5.16}$$

which mathematically expresses the fact that Inertia plus Damping plus Elasticity equals the Response. Equation (5.16) is a second order differential equation and is solved by using either of two methods. The solution methods are termed

- Direct Integration Methods
- Mode Superposition Method.

5.2 Medial Representation of Objects

M-reps, the medial representation used, are based on a hierarchical representation of linked figural models such as protrusions, indentations, neighboring figures and included figures, which represent solid regions and their boundaries simultaneously. The linked collections of figural components imply a fuzzy or probabilistically described boundary position with a width-proportional tolerance. At small scale the figural boundaries are made precise by displacing a dense sampling of the m-rep implied boundary. A model for a single figure is made from a net, also termed a mesh or a chain, of medial atoms. Each atom describes a position and a width. In addition, each atom a local figural frame that provides figural directions and an object angle between opposing corresponding positions, which are medial involutes on the implied boundary. A figure can be expressed as a sequence of over-scale medial nets, which implies successively refined or smaller tolerance versions of the figural boundary. At the final stage a dense displacement field is defined on the boundary of the medially implied object that accommodates the fine-scale perturbations of the object boundary.

5.3 Single-Figure Description via M-Rep

The medial representation used was based on the medial framework. In this framework a geometrical object is represented as a set of connected continuous medial manifolds. These medial manifolds are formed by the centers of all spheres, circles in two-dimensions, which are interior to the object and tangent to the object's boundary at two or more points. The medial description is defined by the centers of the inscribed spheres and by the associated scalar field of their radii. Each continuous segment of the

medial manifold represents a medial figure. This research focused on objects that could be represented by a single medial figure. In two-dimensions, there at two basic types of medial figural segments and each type possesses medial manifolds of dimension zero and one. Figural segments with a single point represent the degenerate case of circular objects and are termed zero-dimensional. In three-dimensions there are three basic types of medially defined figural segments with corresponding medial manifolds of dimension zero, one and two. Figural segments with 2D medial manifolds *M* represent slab-like segments. Tube-like segments, where the medial manifold *M* is a 1D space curve, and spherical segments, where the medial manifold *M* consists of a single point, are degenerate cases. Examples of slab like and tubular shapes are depicted in Figure 5.1 [21].

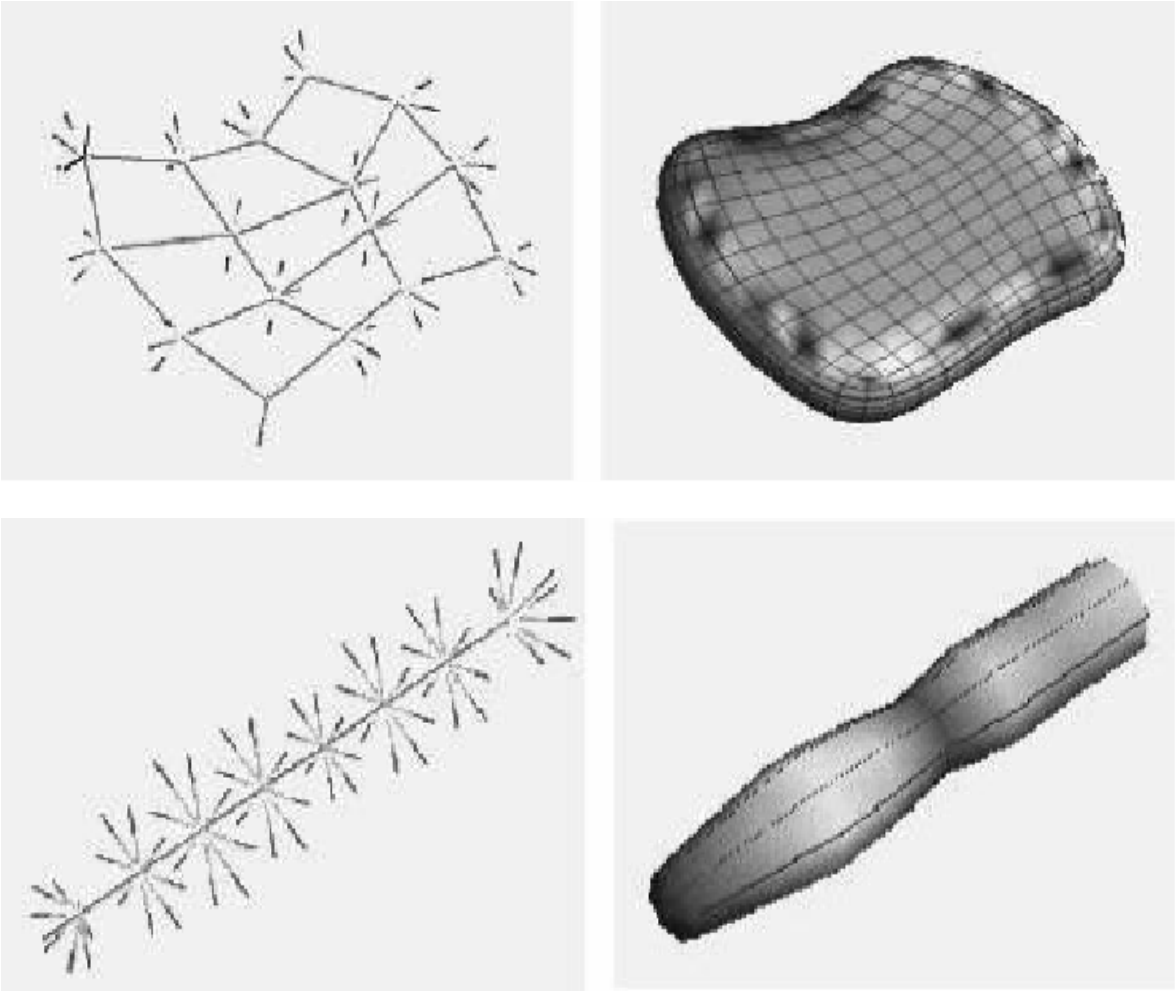

Figure 5.1: Examples of Slab Like and Tubular Figures

5.4 Discretizing Figural Segments

The purpose performing a discrete transformation on the figural segments is to provide a a very accurate model or representation of the atomic structure of the tissue and cells to which deformable motion segmentation is easily applied. To illustrate, the structures presented in Figure 5.2 and Figure 5.3, which were obtained for mathematical analysis and also modeling need to be discritized. The structure presented in Figure 5.2

represents a four-tuple defined by position x, width r, vector b tangent to the medial axis and the object angle.

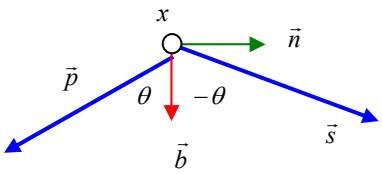

Figure 5.2: 2D Medial *M* Represents a Double Tangency of a Circle To The Boundary

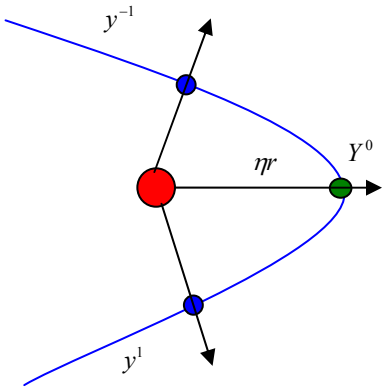

Figure 5.3: An End Atom is a Medial Atom With an Additional Component η

A 2D medial atom carries first-order geometric information at a point on the 1D medial manifold. A zeroth-order description consists of the position x and the radius r of the inscribed circle that is centered at x. A first-order description includes the unit spatial tangent b of the medial manifold at x and captures the first-order width information by the *object angle* θ, which describes the change in radius along the medial axis by the Blum relationship

$$\cos\theta = -dr/ds, \tag{5.17}$$

for the arc length on the medial manifold. The places where the inscribed circle centered at *x* touches the two halves of the boundary, indexed by 1,-1, are defined as y^{-1}, y^{1} with respective normals n^{1}, n^{-1} given by

$$\begin{aligned} n^{1} &= R(\theta)b \\ n^{1} &= R(-\theta)b \\ y^{1} &= x + rn^{1} \\ y^{-1} &= x + rn^{-1} \end{aligned} \tag{5.18}$$

where $R(\theta)$ is the rotation matrix. These parameters and coordinates are illustrated in Figure 5.3.

At the crest a medial atom is introduced at the ends of the medial chains to insure robust sampling. These medial atoms include an extra parameter that defines the position of the crest point on the object boundary that is given by

$$y^{0} = x + r\eta b, \eta \geq 1.0 \tag{5.19}$$

where $\eta = 1.0$ represents a circular end cap and $\eta \triangleright 1.0$ represents increasing elongation. The two opposing boundary points implied by the medial atom are given by y^{1} and y^{-1}, which are the respective normals given by

$$\begin{aligned} n^{1} &= R_{(b,n)}(\theta)b \\ n^{-1} &= R_{(b,n)}(-\theta)b \\ y^{1} &= x + rn^{1} \\ y^{-1} &= x + rn^{-1} \end{aligned} \tag{5.20}$$

where $R_{(b,n)}(\theta)$ is a rotation by θ degrees in the (*b,n*) plane. As in two dimensions, for stability at the crest, medial atoms on the boundary of the medial manifold also include an extra elongation parameter η that determines the crest position. , This representation *x* gives the central location of the solid section of the figure that is being represented by

the atom *m* in both two-dimensions and three-dimensions. The scalar *r* gives the local scale and size of the solid section of the figure that is being represented by the atom. The object angle θ and the direction also define the gradient of the scalar field *r* via the relationship

$$\nabla r = -b\cos\theta\,. \tag{5.21}$$

The scalar field also provides a local ruler for the precise statistical analysis of the object. For example the application of the geometric active contour model algorithm to an echocardiograph image sequence. Two consecutive video frames were taken where the object under study had undergone deformation. Then the algorithm developed was manipulated in order to determine if it was possible to capture the edges of the deformed object. The two frames that were considered are presented in the Figure 5.4.

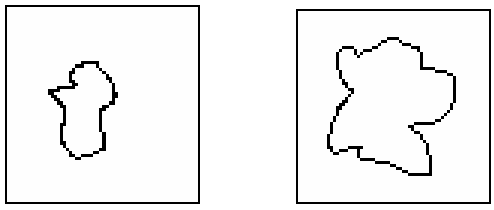

Figure 5.4: Two Consecutive Frames Where The Image Has Undergone Changes

The GVF algorithm was applied to the two frames presented in Figure 5.4 with predetermined iteration parameters. Figure 5.5 presents the iteration convergence for the first frame.

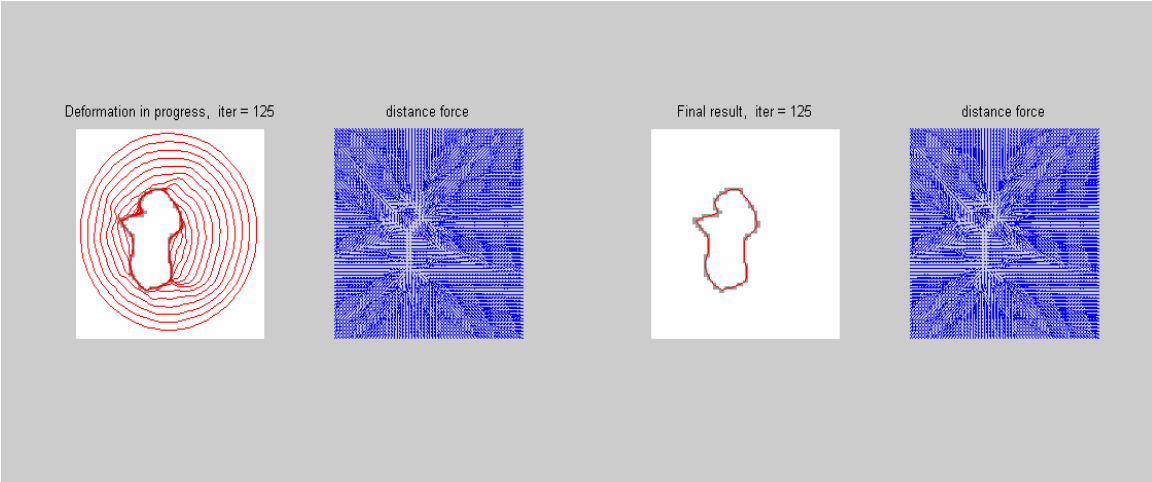

Figure 5.5: Iterations and Convergence of Frame 1

The same algorithm was applied to the second frame and the iterations are presented in Figure 5.6.

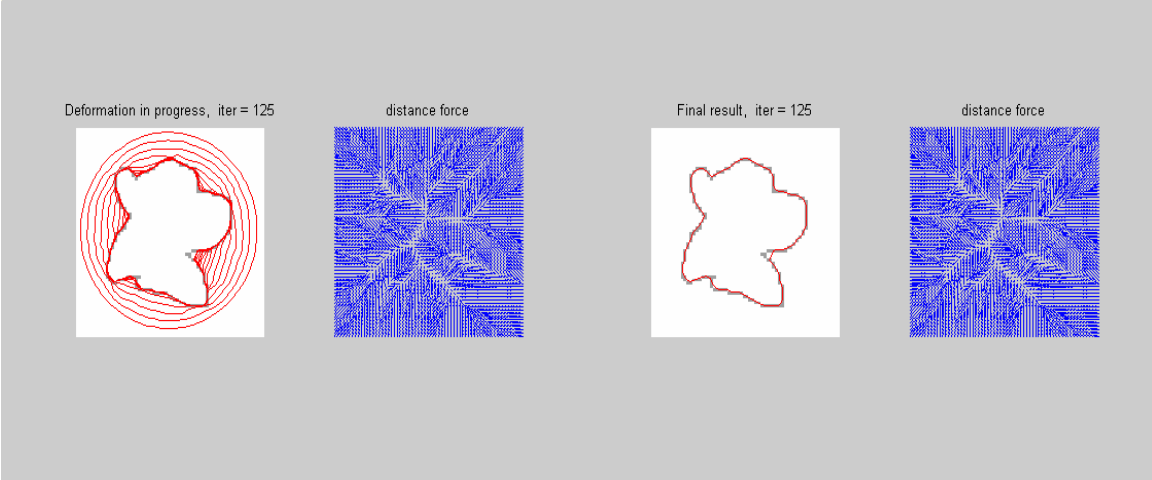

Figure 5.6: Iterations and Convergence of Frame 2

With the known parameters such as the iterations and the other constant values determined in the mathematical model this algorithm can be used to validate the received image/frame with the transmitted image/frame.

CHAPTER 6

COMPRESSED VIDEO OVER NETWORKS

This section deals with the study and behavior of the network with compressed video. After analyzing the various parameters that affect video compression, the parameter that was chosen for detailed analysis was network congestion and the behavior of the network with various congestion control mechanisms [22].

6.1 Effect of Congestion

A vital parameter for effective video streaming over the internet is the bottleneck bandwidth, which gives the upper limit to the speed of a network in delivering data from one end point to the other. If a codec tries to send data at a rate above the upper limit all extra packets will be lost. However, sending data at a rate under the upper limit will clearly result in suboptimal video quality. Although accurate estimation of this time varying limit is difficult, techniques are reported that adapt their rates to a measured bottleneck bandwidth. The bursty nature of Internet traffic is the main obstacle in estimating the bottleneck bandwidth. Another problem associated with rate adjustment at the transmitter relates to a multicast application where each receiver may experience a different bottleneck bandwidth. Rate control also poses a problem with the stored media.

Changing the storage rate increases the computational load for the server and affects the compressed video stream, which requires real time trans-coding.

6.1.1 Congestion Mechanisms: Environment of the Study

A single bottleneck dumbbell network topology was studied. The network topology is presented in Figure 6.1. All simulations were performed with the ns-2.1 simulator [23].

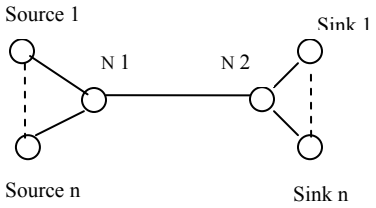

Figure 6.1: Network Topology

The simulations examined the response of slowcc algorithms to a sudden drop in congestion by removing a competing CBR flow. The RED queue had a Queue size of 2.5 times the bandwidth-delay product. The *min-thresh* and *max-thresh* were set to 0.25 and 1.25 times the bandwidth delay product respectively. Round trip time was set to 50ms. These parameters were set in the simulation code, which is given by

```
if {$queue=="RED"}{
$redq set thresh_ [expr 0.25*$rate_w_mb*50/8.0]
$redq set maxthresh_ [expr 1.25*$rate_w_mb*50/8.0]
}.
```

The DropTail queue had a queue size of one bandwidth delay product, which is shown in the code by

if {$queue=="DropTail"} {

*set limit [expr $rate_w_mb*50/8.0]*

$redq set thresh_ $limit

$redq set maxthresh_ *$limit*

} [24].

The competing CBR flow was a square on/off waveform that was created for the purpose of creating a dynamic scenario.

Binomial congestion control, (BCC), algorithms are defined using four parameters. Alteration of the parameters provides different BCC versions [25]. The SQRT version of the BCC algorithm was used during simulation. The congestion control mechanism of the TFRC used a throughput equation to establish the allowed sending rate as a function of the loss event rate and the round-trip time. In order to compete fairly with the TCP the TFRC used the TCP throughput equation, which approximately described the TCP's sending rate as a function of the loss event rate, round-trip time and packet size. This research used the TRFC with both self-clocking and no clocking. TRFC with self-clocking imposed a limit on the amount that the sending rate could exceed the receive rate, even in the absence of loss. TFRC without self-clocking limited the senders sending rate to at most twice the rate that the receiver in the previous round trip received data.

6.2 Congestion Control Mechanisms

This section presents an analysis of the simulation results and plots of the data obtained from the study on drop rates for the queuing mechanisms considered. In addition, the results are presented for the long-term fairness of different TCP versions and the variation in the TFRC behavior for TCP(Vegas).

6.2.1. Drop Rate Analysis

The behavior of different slowcc algorithms that were subjected to a sudden reduction in the available bandwidth was examined. Twenty, (20), long-lived slowcc flows where the bandwidth was controlled by an ON/OFF square wave CBR flow for 150 seconds were used. The 150-second cycle was restarted after a delay of 30 seconds and the cycle repeated. The results illustrating drop rates achieved for a slowcc algorithm using the RED queuing scheme for TCP(SACK) are presented in Figure 6.2.

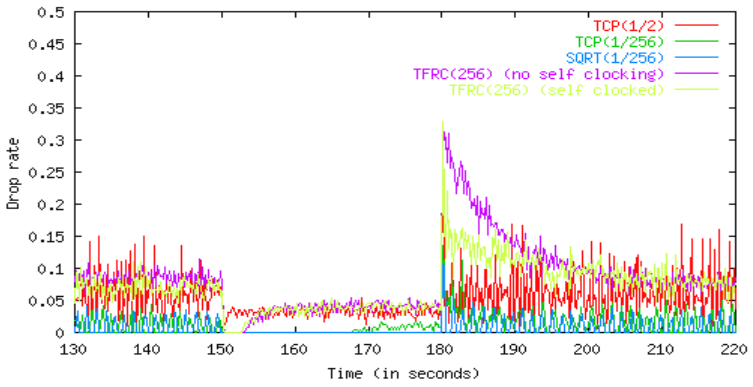

Figure 6.2: Drop Rate For Slowcc Algorithm Using The RED Queuing Scheme TCP (SACK)

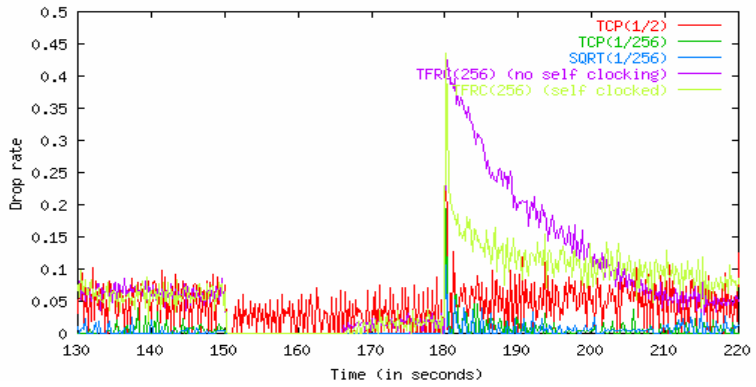

Figure 6.3: Drop Rate For Slowcc Algorithm Using The DropTail Queuing Scheme

TCP(SACK)

The behaviors of the different slowcc versions using RED queuing have been demonstrated. The behaviors obtained using DropTail queuing instead of RED queuing are presented in Figure 6.3. Significant behavioral differences were obtained for the two queuing schemes. The reasons for the differing results from these two queuing schemes can be discerned from a discussion of the basic structure of the schemes. DropTail is a basic first-in, first-out queuing technique where the first packet in the queue is the first packet that is processed. When queues become full congestion occurs and incoming packets are dropped. DropTail relies on end systems to control congestion via congestion control mechanisms. RED is an active queue management scheme that provides a mechanism for congestion avoidance. RED, unlike DropTail, uses statistical methods to drop packets in a "probabilistic" way before queues overflow. Dropping packets in a probabilistic way slows a source down enough to keep the queue steady and reduces the number of packets that would be lost when a queue overflows and a sender is transmitting at a high rate.

6.2.2 Observations

The general trend is that TCP(1/2) has a maximum average drop rate. This can be explained by the fact that TCP is the quickest in increasing its sending rate in order to utilize the new bandwidth that is created by the removal of the CBR flow. Once the limit is reached TCP(1/2) starts dropping packets multiplicatively. For DropTail, even though all the algorithms are TCP friendly, when CBR is removed a lot of packet drops occur since DropTail has no way of informing the senders of impending full queues. When RED is used drop rates fall by almost half of the average drop rate when CBR flow is removed. This can be attributed to the way RED handles queues. TCP(1/256) and SQRT (1/256) produce very low drop rates in the presence or in the absence of CBR flow and drop almost no packets. The important characteristic observed in the first 20 seconds after the removal of the CBR flow was that only TCP(1/2) lost packets. During the remaining 10 seconds both TFRC(256)s lost a few packets. However, when RED was used the removal of the CBR source caused the loss of a few packets for all the slowcc with the exception of SQRT(1/256). Once the CBR was reintroduced there was a sudden increase in the drop rates for all the slowcc mechanisms. However, the TRFC versions, both self-clocked and non-self-clocked, had more prominent peaks. TFRC(Self-clocking) automatically slows down the source when the network becomes congested and acknowledgments are delayed. The self-clocked version of TRFC(256) converged and stabilized faster than the non-self-clocked version. In any event the TFRC's had longer convergence times than the algorithms.

A similar performance analysis was also conducted with all the different TCP versions with the slowcc mechanisms using RED queuing. The most significant change was

observed with TCP(Vegas). SQRT(1/256) displayed a much higher drop rate when compared to TCP(SACK). The drop rate actually reached zero but only for a small interval of time immediately after the CBR flow was removed. These results are presented in Figure 6.4.

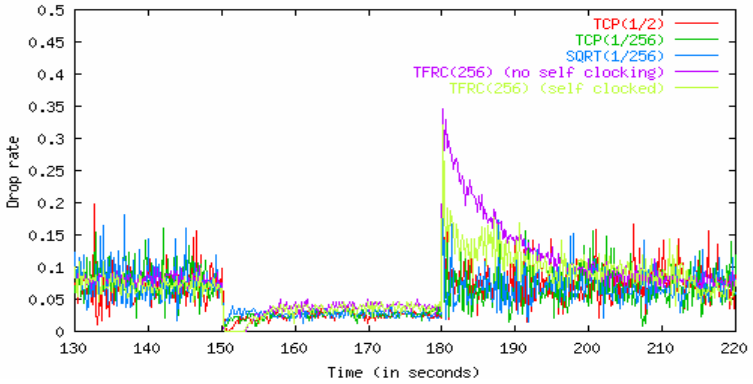

Figure 6.4: Drop Rate For Slowcc Algorithm Using The RED Queuing Scheme TCP (Vegas)

6.3 Behavior Analysis of Slowly Responsive Congestion Control Algorithms

6.3.1 Long Term Fairness

The scenario considered for studying the long term fairness was a network topology with ten long lived flows where five were TCP and the other five were either TCP(1/8), TFRC or SQRT(1/2). The bandwidth was divided in a ratio of 3:1 between the two flows and CBR. The overall bandwidth utilization was observed to be high when the CBR flow period was short. However, when the period of CBR flow was increased TCP performed better than the competing TCP(1/8), TFRC or SQRT(1/2).

6.3.2 TCP(RENO)

When TCP and TFRC are competing with shorter CBR flows, TFRC acquires more bandwidth than TCP. However, TCP eventually takes more bandwidth than TFRC when competing fairly. This result is illustrated in Figure 6.5. When TCP competed with the other two flows the bandwidth utilization characteristic remained similar to that for TCP(SACK). The results are presented in Figure 6.6, Figure 6.6 and Figure 6.7.

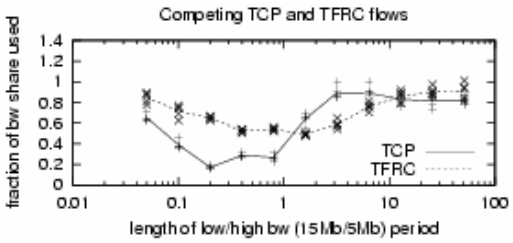

Figure 6.5: Throughput of TCP and TFRC

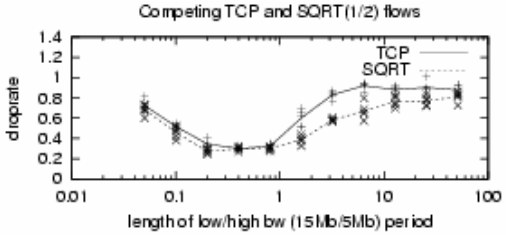

Figure 6.6: Throughput of TCP and SQRT

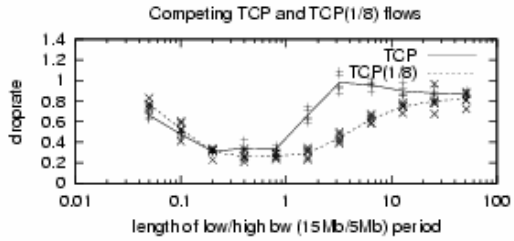

Figure 6.7: Throughput of TCP and TCP(1/8)

6.3.3 TCP(NEW RENO)

With the use of TCP(New Reno), when TCP(1/8) competed with TCP, to a great extent, TCP(1/8) lost its bandwidth margin, which was not the case with TCP(Sack). This comparison is presented in Figure 6.8.

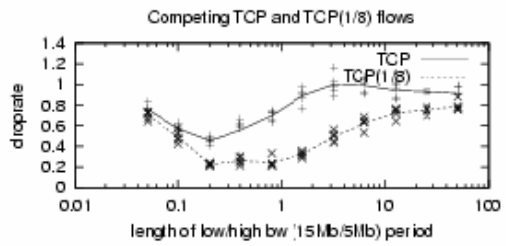

Figure 6.8: Throughput of TCP and TCP(1/8)

When TCP competed with the other two flows the bandwidth utilization characteristic remained similar to that for TCP(SACK). The results are presented in Figure 6.9 and Figure 6.10.

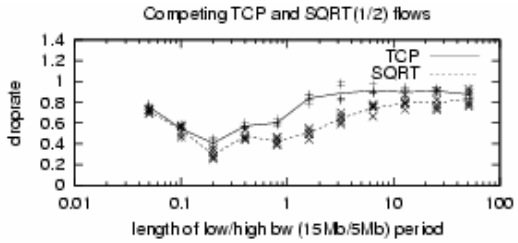

Figure 6.9: Throughput of TCP and SQRT

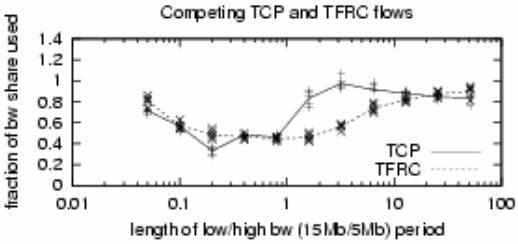

Figure 6.10: Throughput of TCP and TFRC

TFRC or SQRT competing with TCP demonstrated little change in characteristics from those observed for TCP(SACK).

6.3.4 TCP(VEGAS)

When compared to all other TCP protocols, a major difference was observed in bandwidth sharing when TCP and TFRC was competing. The comparison is presented in Figure 6.11. In TCP(Vegas), from the very beginning, TFRC obtained a larger share of the bandwidth than TCP, which was a condition that continued throughout the simulation. In contrast to the TCP(Reno) algorithm, which induces congestion in order to learn the available network capacity, a TCP(Vegas) source anticipates the onset of congestion by monitoring the difference between the rate it is expecting to see and the rate it is actually realizing. TCP(Vegas)' strategy adjusts the source's sending rate or window size in an attempt to keep a small number of packets buffered in the routers along the path [26]. A comparison of the TCP and TFRC results is presented in Figure 6.11.

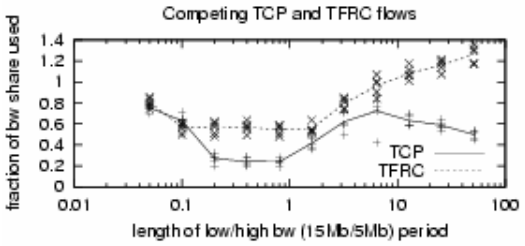

Figure 6.11: Throughput of TCP and TFRC

A bandwidth utilization pattern similar to TCP(SACK) was observed when TCP completed with the other two flows. This observation is presented in Figures 6.12 and 6.13.

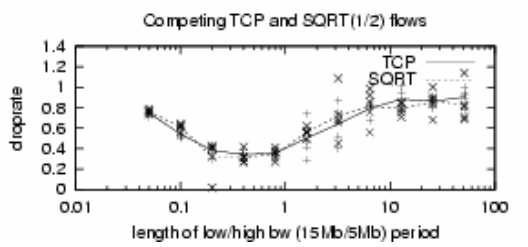

Figure 6.12: Throughput of TCP and SQRT

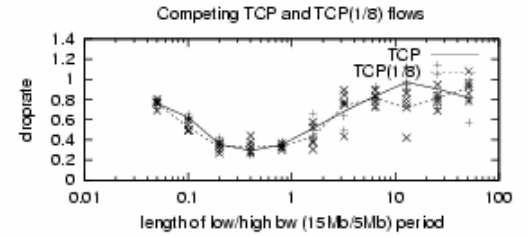

Figure 6.13: Throughput of TCP and TCP(1/8)

CHAPTER 7

RESULTS AND FUTURE WORK

7.1 Conclusions

With the rapid growth of video traffic in the Internet it is in the interest and the need of the market to have better video coding algorithms for medical imaging applications. This research addressed the problem with the introduction of a new external force model algorithm for the active contours and deformable surfaces. The algorithm was tested and the results plotted. The algorithm was implemented in MATLAB and the action of the gradient vectors was presented. This research analyzed the statistical derivations for deformable model generation associated with the Anatomical model and the deformable physical forces model that incorporates such parameters as elasticity and inertial characteristics. An algorithm implementation that includes all these techniques would be able to estimate the changes a deformable object undergoes.

The performances of different slowly responsive congestion control algorithms were investigated. These algorithms were subjected to dynamically changing traffic condition using a Drop Tail queuing scheme for the bottleneck link and compared to the RED queuing scheme. It was found that the drop rate characteristic exhibited by TCP

was much worse in DropTail as compared to that in RED. However, a few slowcc algorithms have reduced drop rates for shorter periods in DropTail than in RED.

7.2 Possible Future Work

The possibility and scope for future study is tremendous. This research, although important and successful, did little but highlight the immense need for further research. One very important area that was highlighted by this research is the need to develop the complete integration of deformable model segmentation and incorporate it into the H.263 algorithm. Additionally, study of transport modification, which takes into consideration the congestion parameter and the mechanisms used for congestion control, seems to be especially lucrative and important.

REFERENCES

[1] http://www.4i2i.com/h263_video_codec.htm

[2] Motion Search Performance using the H.263 Encoder by Alice Yu, http://ise0.stanford.edu/class/ee392c/demo/yu/paper.html

[3] [H.263] ITU-T Recommendation H.263 - Video coding for low bit rate Communication

[4] Block Matching and Compensation by Graham R. Martin, Computer Science Department, University Of Warwick, J.P. Muleller (1998) http://www.dcs.wareick.ac.uk/research/mcg

[5] http://kbs.cs.tu-berlin.de/~stewe/vceg/sequences.htm, The TML Project WEB-Page and Archive

[6] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. Commun., Vol. COM-29, No. 12, pp. 1799-1808, Dec.,1981

[7] J. Ribas-Corbera and D.L. Neuhoff, "On the optimal block size for block-based motion compensated video coders", SPIE Proceedings of Visual Communications and Image Processing, Vol. 3024, pp 1132-1143, February 1997

[8] MPEG-4 Video Group, "MPEG-4 Video Verification Model Version 10.0", ISO/IEC/JTC1/SC19/WG11 Document MPEG98/N1992, San Jose, February 1998

[9] M.H. Chan, Y.B. Yu, A.G. Constantinides, "Variable size block matching motion compensation with applications to video coding", IEE Proceedings, Vol 137, Pt. 1, No. 4, August 1990

[10] G.R. Martin, R.A. Packwood and I. Rhee, "Variable size block matching estimation with minimal error", SPIE Conference on Digital Video Compression: Algorithms and Technologies 1996, San Jose, USA, Vol. 2668, pp 324-333, February 1996

[11] G.R. Martin & R.A. Packwood & M.K. Steliaros, "Reduced entropy motion compensation using variable sized blocks", SPIE Proceedings of Visual Communications and Image Processing, Vol. 3024, February 1997, pp 293-302

[12] R.A. Packwood, M.K. Steliaros and G.R. Martin, "Variable size block matching motion compensation for object-based video coding", IEE 6th International Conference on Image Processing & its Applications, Dublin, Ireland, July 1997, pp 56-60

[13] M.K. Steliaros & G.R. Martin & R.A. Packwood, "Locally-accurate motion estimation for object- based video coding", SPIE Proceedings of Visual Communications and Image Processing, Vol. 3309, January 1998, pp 306-316.

[14] A Method for Reducing Block Artifacts by Interpolating Block Borders by Stathis Panis , Guillaume Stamm, and Robert Kutka Siemens AG, Dept. ZT IK 2, 81730 Munich, Germany Ecole des Mines de Saint Etienne, France

[15] Snakes, Shapes, and Gradient Vector Flow Chenyang Xu, Student Member, IEEE, and Jerry L. Prince, Senior Member, IEEE (Used their MATLAB Code Modules)

[16] Deformable Model-Based Shape and Motion Analysis from Images using Motion Residual Error Douglas DeCarlo and Dimitris Metaxas

[17] P. Zerfass, C. D. Werner, F. B. Sachse, and O. Dössel, "Deformation of surface nets for interactive segmentation of tomographic data", to appear in Proc. 34, Jahrestagung DGBMT, 2000

[18] C.Nikou, F. Heitz, and J. P. Armspach. Multimodal image registration using statistically constrained deformable multimodels. In Proceedings of the IEEE International Conference on Image Processing (ICIP'98)

[19] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,”. Vis., 1987

[20] D. Terzopoulos and K. Fleischer, “Deformable models,” Vis. Comput, 1988

[21] T. McInerney and D. Terzopoulos, “A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis,” Comput. Med. Imag. Graph

[22] S. Floyd and K. Fall, “Promoting the use of End-to-End Congestion Control in the Internet”, IEEE/ACM transactions on Networking, 7(4) August 1999

[23] Jacobson, “Congestion Avoidance and Control”, Proceedings of ACM/SIGCOMM August 1988

[24] D. Bansal, H. Balakrishnan, S. Floyd and S. Shenker, “Dynamic Behavior of Slowly Responsive Congestion Control Algorithms”, SIGCOMM 2001

[25] D.Bansal and H.Balakrishnan, “Binomial Congestion Control Algorithms”, Proceedings of the Conference on Computer Communications IEEE Infocom, April 2001

[26] S.H.Low, L.L. Peterson and L Wang, “Understanding TCP Vegas: A Duality Model”

BIBLIOGRAPHY

Compressed Video Communication by Abdul H.Sadka, Wiley Publications 2002

Compressed Video over a Network, Edited by Ming-Ting Sun and Amy R. Reibman, Signal Processing and Communication Series

Digital Video Processing by A. Murat Tekalp, Prentice Hall Signal Processing Series