Structured Video Coding

by

Patrick Campbell McLean

B.A., Engineering
Cambridge University, England
1989

Submitted to the Media Arts and Sciences Section,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1991

Signature of Author ....
Media Arts and Sciences Section,
School of Architecture and Planning
May 10, 1991

Certified by ....
Andrew Lippman
Associate Director, MIT Media Laboratory
Thesis Supervisor

Accepted by ....
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students

Structured Video Coding

by

Patrick Campbell McLean

Submitted to the Media Arts and Sciences Section,
School of Architecture and Planning
on May 10, 1991, in partial fulfillment of the
requirements for the degree of
Master of Science
at the
Massachusetts Institute of Technology

Abstract

In order to transmit a video signal over a limited bandwidth channel many techniques have been developed which can reduce the quantity of data required to represent moving image sequences. Almost all current coders use techniques based on reducing the statistical redundancy of the signal and psychophysically less significant details. Such coders are relatively successful for coding signals at bandwidths of 1 Megabit/second and higher but below this level purely signal based techniques become much less robust.

This thesis will attempt to use an analysis of the structure of the image, the components from which it is composed, in order to gain greater image compression. The input material is examined and divided into elements representing the background, the actors and any other significant independent objects. The background image is stored as a model at the receiver and animated according to estimates of the motion in the original scene. Actors and other foreground objects are segmented out of the scene, compressed and transmitted using conventional signal based methods. This technique typically leads to reductions in bandwidth of greater than 70%. In addition, enabling the computer to distinguish the components of the picture allows for forms of post-shoot modifications of the picture not possible with conventionally represented signals.

Thesis Supervisor: Andrew Lippman
Title: Associate Director, MIT Media Laboratory

The work reported herein was supported by a contract from the Movies of the Future consortium, including Apple Computer, Inc., AT&T, Eastman Kodak Co., Viacom International Inc., and Warner Brothers Inc.

Contents

Chapter 1

Introduction

1.1 Structured Video

Structure within video could be defined in many different ways depending on the perspective of the author. In this thesis the term video structure is used to refer to the content of a scene. What are the components which make up the image, what is their relationship to one another and what is the position and motion of the camera? The aims of this investigation are twofold: one is to establish whether analysing the image sequence from this perspective can aid us in creating more efficient and more usable video coding systems; the other is to explore ways in which this can help us to modify scene content or to create entirely new scenes, 'synthetic movies'.

This thesis will address the reasons for doing this, the methods that can be used to make it possible, the problems that were encountered in so doing and what can be achieved as an end result of the process.

1.2 Script to film: Local models.

The process of creating a film can be seen as a process of data expansion. The idea of a writer becomes a script which gets enacted and recorded onto film. The quantity of data in the film is greater than that in the script which in turn is greater (we guess) than the data composing the idea in the writer's head. The reason why the script holds less data but can convey as much information is because the reader is expected to possess a certain level of knowledge about the world. Take this description of a room in a play:

> "It is very large and looks formal because it is underfurnished. There are double doors at left-back... The room is painted white, walls and ceilings. There is a low wide divan, covered in rough black material, in the right back corner; a window, with dark red curtains, in the right wall; a large round, ornate mirror on the left wall; a low shelf of books under the window. ... The life of the room is concentrated around the divan. A low table by its head has a telephone, and is loaded with books and papers and a small reading light."[27]

This description leaves much unsaid. How much furniture is 'underfurnished'? What does a divan look like? What colour is the telephone? Although none of this is stated we each are able to interpret these statements and conjure up a vision of the scene. The author does not need to present us with a photograph in order for us to visualise the picture. Also, if there is an ambiguity, if a detail is not stated, it is probably either unimportant or can be derived from other details. In fact, one of the beauties of a play is that it lends itself to constant reinterpretation by each director who chooses to perform it. We also understand that the room will not change over the course of time. There is no need to present 30 images each second to describe how the scene will appear in a filming of the room. All that is required is a description of the motion of the camera.

This discussion is not very enlightening while movies are stored on film. However, the advent of computers and the digital storage of video offers us the opportunity to take a different viewpoint. The computer allows us to devise means of manipulating images which diverge from those which are traditional. It is possible to create models of reality which can be animated according to the wishes of the user. These models become the implicit knowledge that is used to expand a script into a movie. The more knowledge that can be stored locally, the less we are required to transmit. In a movie this knowledge would be local databases of different rooms which appear in the duration of the action. Also models could be stored of tables and chairs and similar commonly occurring objects. A particularly sophisticated system would be able to store and animate models of the people appearing in the movie, knowing their appearance and perhaps their characteristic walking and talking patterns. The objects at the receiver acquire some knowledge of themselves and can therefore act 'intelligently'. The job of the transmitter moves from one comparable to the animator who must draw each frame laboriously, to the director who can issue orders to actors and objects intelligent enough to acceptably portray a scene from incomplete information.

1.3 Scene Modification

The last section referred to the extremely flexible nature of the script for a play or film. This script lays down the essentials of the story but leaves many decisions up to the director. By moving from a signal-based representation of images to a model-based representation we move a step towards allowing each individual user (or a computer) to create a different and unique rendering of a scene according to their own needs or tastes.

The desire to modify a scene can arise in many situations. One is in the one-off modification of an existing movie or television show. A global change to the material would be a desire to colorize the movie, or to change the aspect ratio to match a new medium. Alternatively, more local changes might be desired to correct mistakes in the original: removing an object that should not have been present, closing a door or changing the camera lens or view direction.

A different situation is the creation of entirely artificial film clips. These could be to convey information by the use of video objects, perhaps indicating the status of a system. A remotely monitored site could generate video of its current state without the need for a camera. Sensor readings would translate into video representations, a water flow meter would translate into video of water flowing through a pipe at different speeds or even water bursting out of a hole in the pipe if that event occurred.

The availability of the people and scenes in a movie as a form of 'clip art' creates the possibility that people might also want to use these elements to create entirely new film sequences. Actors could be swapped or placed in novel situations. There are many examples in art where new pieces are formed from collages of existing objects. A model-based representation of video would encourage this practice in the film domain.

An alternative perspective is to view this as a merging of computer graphics and photography. In computer graphics images are stored as models and parameters for their animation. These models can be altered at will and provide a very compact representation of the image.

However, the rendering of photorealistic scenes is highly processor intensive and often fails to accurately imitate reality. In photography we have images that are relatively good mirrors to reality but which are data dense and hard to manipulate. The computer graphics community have been tuning their models such that more and more realistic images can be generated. This work is an approach from the opposite direction, an image processing approach to generating more and more manipulable photographic images.

Chapter 2

2-D Image Modeling

2.1 Rationale

The software work of this thesis concerned itself with a subset of the task of analysing image structure: modeling image sequences in two dimensions. The principal aim was to take material from the 1950's television show 'I Love Lucy' and encode it at low bandwidths (64Kbit/s - 300Kbit/s). The reasons for wishing to encode video at such bandwidths can be seen in two domains. One is due to the introduction of the new ISDN[1] telephone technology which has a data bandwidth of 64Kbit/s. If this becomes a ubiquitous technology it would be desirable to make it possible to receive and maybe transmit video in addition to other types of data. The other domain is perhaps more interesting, and that is to consider using CD or similar technology in a novel way. In the entirety of the 'I Love Lucy' series the number of sets used is in the low hundreds. Creating models of these sets and storing them in a database on the CD allows these models to be shared between scenes and even episodes. If the actors are subsequently encoded at a bandwidth of less than 100KBit/s then over 20 hours of 'I Love Lucy' could be stored on a single CD. "The Complete I Love Lucy" would be a small CD box set.

Transforming the image into models also allows for the exploration of picture modification. When the perseverant fan club has watched each episode twenty times they can start creating their own versions using the raw material of the sets and the actors available on the CD. In particular this work highlighted a technique for 'scene-widening', an important task required

[1] Integrated Service Digital Network - The next generation of telephone technology currently being introduced

for converting current television footage to the wider aspect ratio High Definition Television (HDTV).

2.2 Image Modeling

The basic premise for bandwidth reduction by image modeling is that the model need only be sent once, not thirty times each second. All that needs to be sent each frame is the parameters for animating the model and information about the movement of the camera in the intervening time. The extent to which this is possible is dependent on the quality and completeness of the model at the receiver. Ideally we would like a full three dimensional database of the scene. If this was the case the image could be re-rendered in any way the user chose and the data representation would be very compact. However, the disadvantage of attempting to pursue this goal is that existing footage does not contain 3-D information and so considerable effort has to be used to extract depth information from the 2-D scene, a very hard, sometimes impossible, task. The aim of this work is to concentrate on discovering how much one can achieve using only two-dimensional models.

What is meant by two-dimensional models is flat images of specific scene components such as a room, a table or a person. In some cases depth information was included, but only to the extent of specifying the relative depths of the different image planes: Lucy is behind the table but in front of the wall. Mostly the scene was simply divided into 'background' and 'foreground'. The foreground was the actors and any other moving objects and the background was the set. This is an effective model for image coding since the set is usually a rigid object which does not change over the course of a scene and so should not need to be sent more than once. The models were stored as cylindrical projections so that, once captured, a model will be valid for any camera angle or lens size so long as the pivot point of the camera does not move. If the pivot point of the camera does move then the images can be distorted but will need updating when the distortion becomes too great or too much new scene detail is revealed. The efficiency of an image coder which uses these principles will therefore be dependent on the amount of camera pivot movement and the amount of the scene which it can effectively model as a rigid body. Complicated camera movements or close-ups of faces are both very difficult to model within this framework; more static scenes where the picture area is not dominated by people are amenable to compression.

2.3 Coder Design

The way in which the low bandwidth coder was constructed was to create a single image of the background set. This image was stored locally at the receiver. The original sequence was then analysed and the actors removed from the background. In the main body of the work the actors were considered to be objects which could not be modeled and therefore were coded using conventional image coding techniques. This code was transmitted to the receiver, decoded and placed into the background set model to recreate the original sequence. Three sequences were chosen for research, one where the background was still, and two where the background was panned across by the camera. In each case the actors constituted about 20%-30% of the picture area and therefore this is the proportion of the picture that required transmission each frame.

Many different strands of image processing had to be integrated in novel ways in order to create the coder. These strands can be categorised into several domains:

- Signal-Based image compression.
- Image Compositing.
- Motion analysis.
- Foreground/Background Separation.
- Image Reconstruction.

The way in which the model is assembled is detailed in figure 2-1.

There are several parts to this. The transmitter can be divided into analysis and encoding stages. In the analysis stage the model of the background set is created, the motion of this set in the original ascertained and the segmentation mask of foreground and background determined. The encoding stage uses the segmentation mask to separate out the foreground which is then encoded using conventional signal coding techniques. This code is transmitted, along with side channels for the segmentation mask and the camera motion parameters.

At the receiver the camera motion parameters are used to animate the background, already available in local storage. The foreground data is decoded and formed into a coherent image

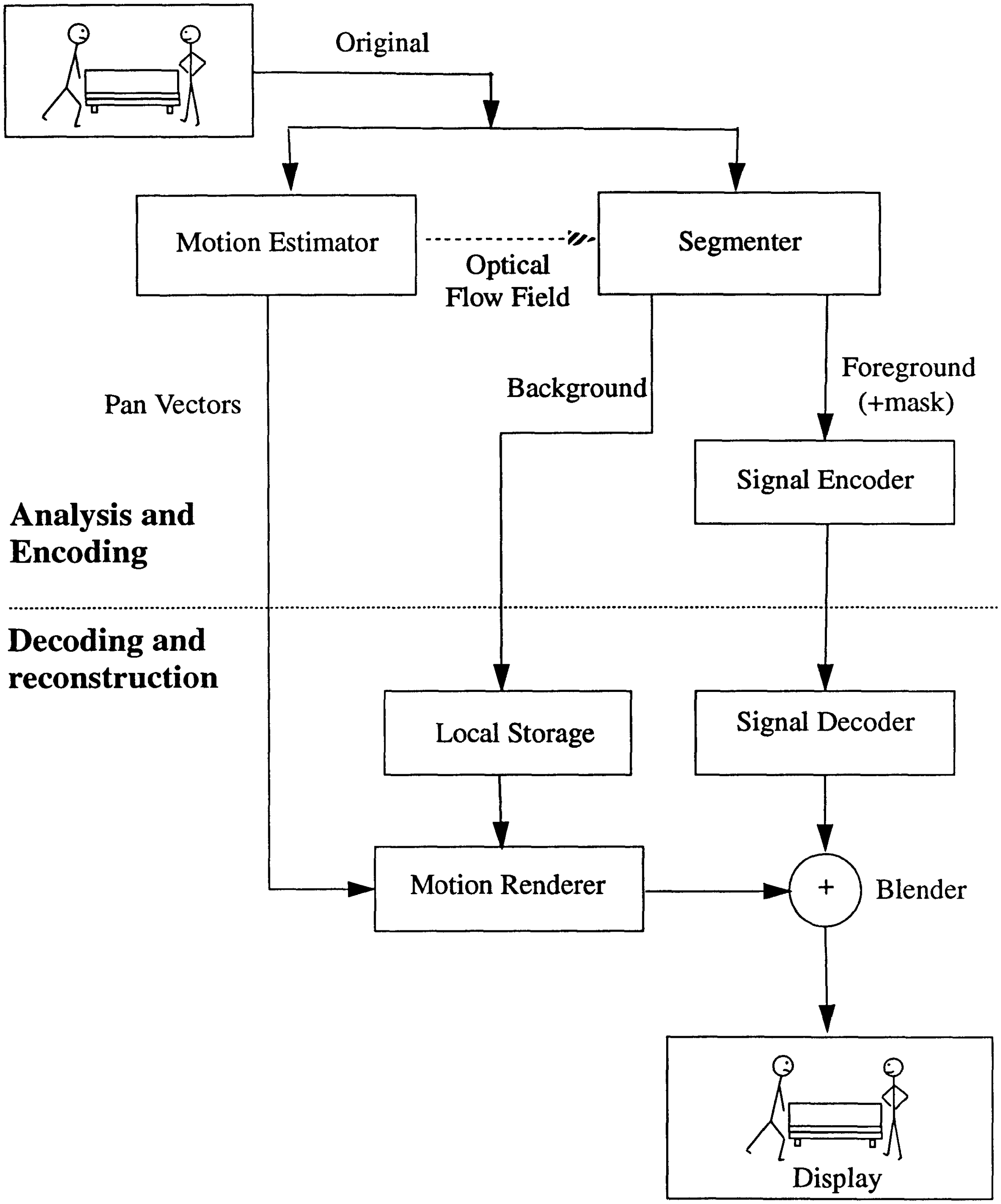

Figure 2-1: Elements of a model-based image coder

using the mask. The final stage is to blend the actors back into the animated background as seamlessly as possible. The result of this operation is the image ready for display.

The transmission of the background image itself can occur in several ways. In the CD application the backgrounds will be placed as a database on the disc appended to the foreground data. In the case of 64Kbit/s video for transmission over a telephone, the background must either be sent in parallel with the foreground, shared between as many frames as possible prior to its first required use, or it could be downloaded in advance.

The details of the workings of each of these units is described in more detail in the following chapters. However, at this stage several issues should be considered in overall coder design. The image coding model suggested here is one which is designed to perform very efficiently with certain types of sequence. The efficiency that is gained by accurately modeling the 'I Love Lucy' sequence is offset by the losses which will arise when the coder is faced with an image which it cannot interpret. In addition, the individual components which make up this coder are tuned to their task. If an image coder is required which will efficiently tackle many different types of image sequence it will be necessary to incorporate several different types of encoding methods into the system. In such a case at least one must succeed in encoding the picture but others will often be able to encode particular sections more effectively. Depending on picture content different coder designs will perform in varying ways and the best coder and best module for the current task must be chosen. Image coding systems such as this one and those that follow it must address complex systems issues in chosing optimum encoding methods. The problem is one of adapting intelligently to changing circumstances and coordinating a semi-distributed system to choose an optimum configuration. Each unit (such as one which segments out actors or attempts to create background sets) must be able to monitor its own performance such that it can indicate to others whether its output is useful. Several alternative paths will exist to take the data from the original to the reconstructed image, some tuned to particular picture types and some more general. If all the modules in a particular valid path were active then that path would be chosen. If more than one path indicated that it could encode the picture successfully then greater weight would be attached to paths considered desirable according to current goals, for instance ones which achieved low bandwidths. A model of a multi-path image coding system is shown in figure 2-2. This figure shows a possible set of components but does not indicate the means of control. This control could either be centralised, a module would ascertain which units are functioning well and choose the path accordingly, or the system could be distributed with each unit outputting activation and suppression signals which acted together such that the most successful path under the current goals would show up with the highest energy and by that means be chosen to perform the encoding.

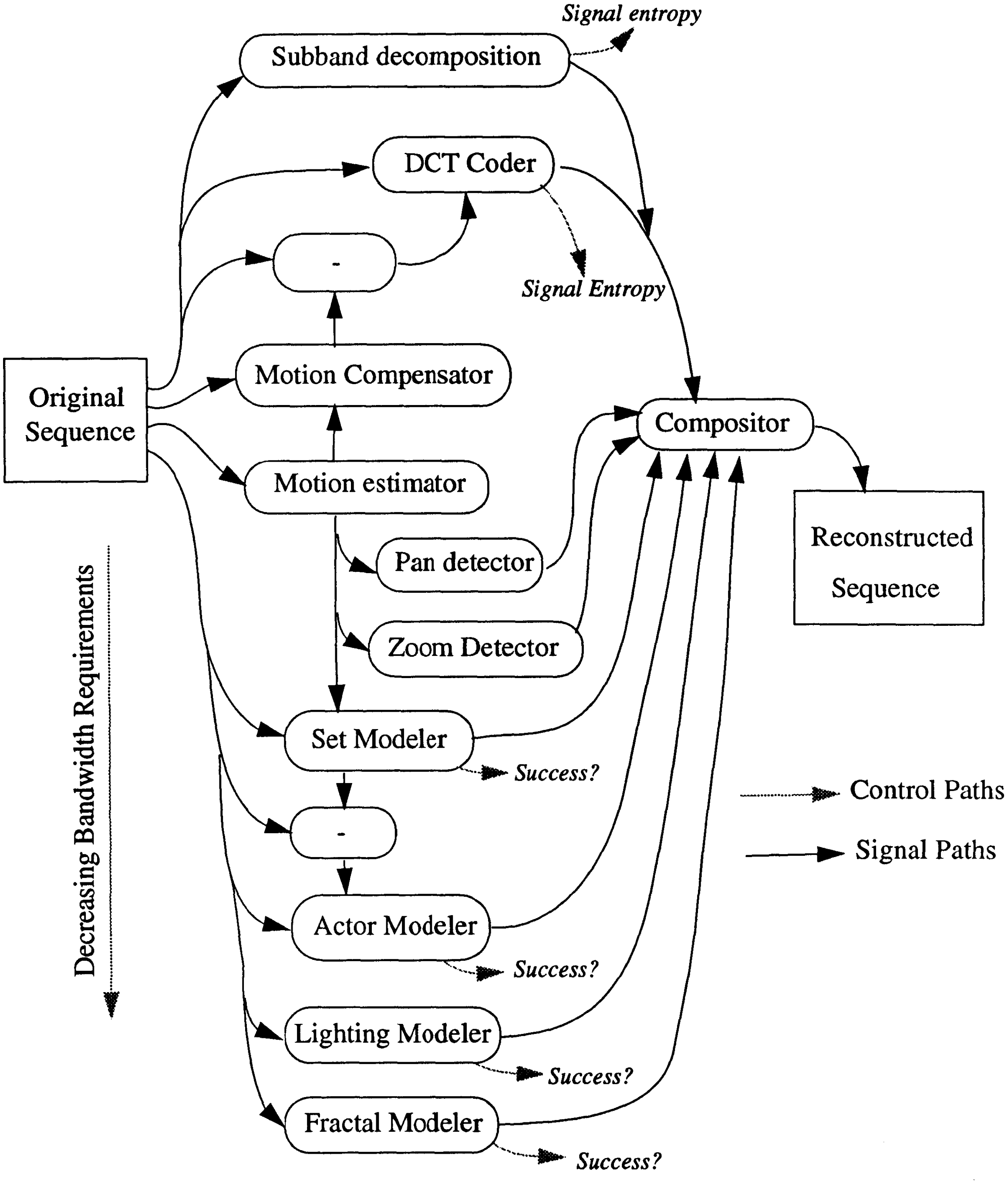

Figure 2-2: Model for a general multiple-approach image coder

Chapter 3

Signal-Based Coding

3.1 Digital Video

Ever since the mid 1960's and particularly following the advent of the FFT[1], researchers have realised the possibility of using a computer to store, analyse and manipulate video sequences. The advantages of being able to do this are similar to the advantages associated with storing and manipulating text on a computer. Essentially they fall into two categories. One is the ease of replication and distribution of digitally represented information and the other is enabling individuals and small companies or research groups to perform tasks previously only possible with expensive equipment and skilled technicians. Writing and formatting this thesis to its current presentation fifteen years ago would have been a very considerable undertaking. It is possible that in fifteen years time it will be as easy for ordinary people to edit and manipulate a video sequence as currently it is to edit this text document. It will be possible to access information in video form over networks and telephone lines, thereby considerably increasing our choice in possible viewing material. It is hard to envisage to what extent the technology will allow individuals to manipulate the content of a film (and to what extent they will wish to) but it seems reasonable to expect that when technological barriers to creativity are broken more people will see themselves as possible creators. Witness the remarkable increase in the number of people who are willing to produce documents and posters on computers for public distribution. Every Macintosh user sees him or herself as a graphic designer. Naturally, we

[1] Fast Fourier Transform, invented by Cooley and Tukey in 1965 [8]

should sometimes question whether this is a good thing...

These are the possible incentives for storing video in a digital form. However there are some significant barriers which have made digital video long in coming. The fundamental barrier is the sheer volume of data that is required to represent even a very short sequence of video. The resolution of RS-170A (component NTSC) when stored digitally is typically 640 points horizontally by 480 lines by 30 frames per second. To represent a full palette of colours about 3 bytes are required per pixel. These numbers mean that a single minute of raw digital video requires 550 Megabytes of memory, an average film over 50 Gigabytes of memory. Even by today's standards this is an enormous amount. From this has grown the field of image data compression, better known as image coding.

3.2 Signal Based Image Coding

There are many different ways in which the quantity of data required to represent an image can be reduced, all of which are based on one of two principles: either they are attempts to reduce the redundancy in the representation or attempts to exploit knowledge gained about the psychophysics of image perception in order to place less emphasis on visual detail which the eye finds less significant. Techniques which exploit redundancy do not discard any information and are therefore called lossless. The other techniques are lossy by definition, in that they discard data which is considered to be less visually important. In addition, a distinction can be made between intraframe techniques, which independently encode each frame, and interframe techniques in which the correlations between frames are exploited.

An image coder can typically be split into four elements. A predictor, a de-correlator, a quantizer and a code assigner. A particular image coder might have one or all of these components and some techniques overlap these functional boundaries; however, in most cases this distinction is a helpful tool for analysis. These elements will be discussed in the following sections.

3.2.1 Prediction

The aim of the prediction component of a coder is to reduce the energy of the signal by encoding the difference between the current value and a predicted value rather than the current value itself. If the prediction is good the values to be encoded will often be close to zero and require few bits. The prediction can be linear, for instance a particular pel might be predicted from the weighted summation of a set of preceding pels (this is known as Differential or Delta Pulse Code Modulation (DPCM)); or nonlinear, for instance a pel being adaptively predicted from the previous pel horizontally, vertically or temporally depending on which provides the lowest difference energy.

Temporal prediction can be particularly effective. Image sequences tend to have a great deal of temporal redundancy. To exploit this, coders often encode the difference between the current frame and the last frame. If the two frames are well correlated the signal energy to be coded will be considerably reduced. A refinement of this technique is *motion compensated prediction.* Motion compensated prediction is the name given to the method whereby an image is predicted not by the previous frame but by a two-dimensional warping of the previous frame designed to account for the motion of objects in the image during the interframe interval. This is a non-linear extension of DPCM. Prediction loops of this kind have a canonical form, shown in figure 3-1.

A very important part of this design is the manner in which a model of the decoder is built into the encoder. This is done so that the difference image is created from the difference between the current original and the predicted frame *constructed in the decoder*, rather than a frame predicted from last original. If the previous original were to be used for comparison quantization errors would accumulate in the image in the receiver. The signal flow can be thought of as follows. First, the current image to be encoded I(0) is compared with a motion compensated version of the previous frame $\mathrm{I}_{mc}(-1)$. This comparison results in a difference image which is encoded (by any intraframe technique) and the codes S passed on to both the channel and also to a the model of the decoder in the encoder. The decoder has several elements. First it must decode S into the difference image and then add this to the motion compensated previous frame. This is the image that will be displayed, $\mathrm{I}_d(0)$. A motion estimator compares the current input frame I(0) with the previous displayed image $\mathrm{I}_d(-1)$, and creates a set of

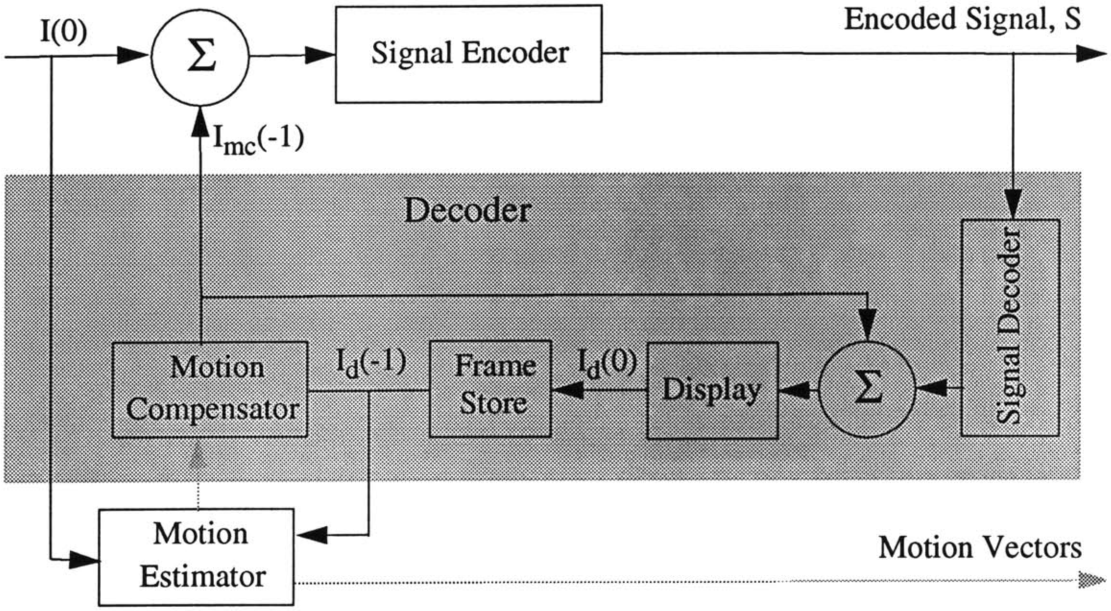

Figure 3-1: Motion Compensated Prediction Loop

motion vectors to be transmitted down a side channel. These vectors are also used to warp $\mathrm{I}_d(-1)$ to create $\mathrm{I}_{mc}(-1)$, the image used to predict the next frame. All encoders which use this type of loop must have an image with which to start the cycle. This is known as the *key frame* and is encoded using an intraframe technique.

This type of model is used in the image compression standards set by CCITT[2] and its successor MPEG[3]. Details of these models can be found in [16, 11]. In MPEG the original sequence is prefiltered and then divided into groups of typically 15 frames. The first frame of such a group is the key frame and then subsequent frames are either predicted frames or interpolated frames (see figure 3-2).

MPEG uses a transform coder based on the DCT (see below) as the signal encoder. This technique has been shown to produce "near vcr" quality pictures at 1.2 MBit/s and is being proposed as a standard for image coding at this bandwidth. Image coders which work at bandwidths below 1Mbit/s tend to use the same principles as MPEG to achieve their compression.

[2] Committee Consultatif International de Telephonie et Telegraph

[3] Motion Picture Experts Group

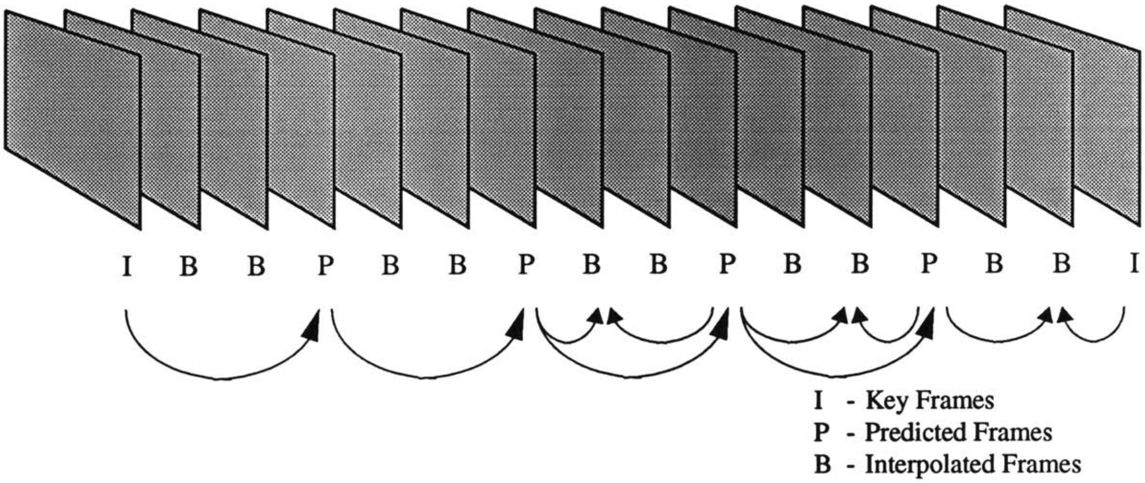

Figure 3-2: The MPEG 'Group of Frames'

The CCITT videoconferencing standards from which MPEG grew are based on bandwidths of n x 64Kbit/s (n = 1-30). An example of a coder within this framework is provided by Pereira et. al. [11]. Variants on these exist which use additional techniques such as vector quantization and conditional replenishment, for examples of these see [53, 40].

3.2.2 De-correlation: Subbands and the DCT

Images are not random signals. Usually there is a great deal of correlation between samples in the image. If this correlation can be identified then grouping together correlated pixels will lead to a concentration of signal energy. This concentration can be exploited by later stages of a coder which will allocate more bits to the energy dense parts of the picture and fewer bits to the energy sparse portions. In particular real images tend to have a concentration of signal energy in the low spatial and temporal frequencies and psychophysical evidence indicates that the human visual system is less sensitive to noise in the higher frequency components of an image over the lower frequencies [46, 14]. The combination of these two facts leads to the proposition that the higher frequency components of an image should be coded with many fewer bits than the lower frequency components. Identifying these frequency components will allow for more efficient encoding of an image. Two common methods for performing this are subband decomposition and transform coding.

In a subband decomposition a complete image is divided into its spatial and temporal frequency elements, the sub-bands of that image. The Media Laboratory has been working for several years on a hybrid coder which vector-quantizes (see below) these sub-band representations of images [5, 44, 6]. The different sub-bands are quantized in varying degrees depending on the significance attributed to those bands, and the variation in bit allocation is performed by changing the parameters of the quantizer in each band. This algorithm has been shown to code images successfully at a range of bandwidths, although the image quality at equivalent bit rates is slightly worse than MPEG coded sequences. The main advantage of the subband coding technique over the MPEG technique is that it is scalable; since it is a pyramidal representation it naturally supports image representation at multiple temporal and spatial resolutions.

In transform coding, a block of pels in the image is transformed into a different domain with the aim of minimizing the statistical dependency between the transform coefficients; this leads to many coefficients being small enough that they either need not be coded or only require few bits. Many different types of transform exist but one of the most common is the Discrete Cosine Transform, a specialisation of the fourier transform for cases of purely real signals. One example of implementing the DCT within an encoder is offered by the JPEG[4] still picture coding standard. The DCT outputs an array of coefficients that roughly represent the frequency components of the signal. In the JPEG standard three operations are subsequently performed which exploit the characteristics of the picture in the transform domain. Firstly, the matrix of coefficients is multiplied by a visual weighting matrix which amplifies the low frequencies more than the high frequencies; the aim being that subsequent bit allocation will be biased towards the more visually important coefficients. Secondly, the result is scalarly quantized to different degrees depending on the desired output bandwidth. Thirdly, the block is scanned in a manner which bunches together the predominantly zero-valued high frequency coefficients. The output of this scan can be compressed by run-length encoding to a degree which would not have been possible had the picture been represented in the spatial domain.

[4] Joint Picture Experts Group

3.2.3 Quantization

In any coder, the actual compression only occurs through the action of the quantizer and the code assigner. Prediction and decorrelation prepare the signal such that few bits are required for encoding but do not actually reduce the quantity of data by themselves. Quantization can either be scalar or vector, scalar quantization being a specialisation of vector quantization.

In scalar quantization, the number of signal values which a signal may take is reduced. For instance, luminance values in an image are often represented by an 8 bit number, allowing a total of 256 distinct possible luminance values. A scalar quantizer might limited this range to 64 values. The result would be a less faithful rendition of the image but only 6 bits would be required to represent each pixel rather than 8. Vector quantization is an extension of the idea of scalar quantization, and has been explored by many researchers [15, 9, 37]. Rather than limiting the number of discrete levels that a single pixel can take, the number of different combinations of a set of pixels is limited. This set of pixels, typically a 2x2 or 4x4 square block of pixels in the image, can be viewed as a vector in N-dimensional space where N is the number of elements. The task is to find a small set of vectors which together minimize the total error measure between themselves and the actual vectors that they are to represent. The image is then encoded as indices to the chosen vectors (blocks of pixels). The vectors themselves are stored in a 'codebook' which is transmitted along with the encoded image. Compression is gained through the fact that the codebook is only a fraction of the size of a codebook of all the possible combinations, allowing each vector to be represented by a low precision number.

3.2.4 Code Assignment

Having passed through all the preceding elements a signal emerges as a set of numbers. At this stage each number could be transmitted and decoded at the receiver, allowing faithful reproduction of the image. However, the *entropy* of the signal would not have been exploited. A signal's entropy represents the information content of that signal, and can be expressed in bits per sample. A commonly occurring sample does not carry much information and therefore has a low entropy; conversely, a rare sample contains much more information and has a high entropy. Code assignment is performed by an entropy coder which uses a knowledge of the statistics of the incoming number stream to assign codes with few bits to commonly occurring symbols and codes with more bits to rarely occurring symbols. The Huffman coder is an example of such a system. The problem with the Huffman coder is that it only exploits the entropy of individual samples, and ignores correlation between sequences of samples. Two solutions exist to this. The first is to combine Huffman coding with run-length encoding. In run-length encoding the value of a sample is encoded in parallel with the number of sequential occurrences of that sample. This is an effective method for efficiently encoding long runs of zeros for instance. The second is to use a more sophisticated entropy coder, the Arithmetic Coder, which assigns codes based on the statistics of sequences of samples rather than individual samples. These entropy coders translate the lower energy of the signal into fewer bits.

3.3 Model Based Coding

The last few years has seen the growth of a new technique known as model-based coding [41]. Model-based coding is the name used for the technique of building a model of the image at the receiver and animating this model using control information gained by image analysis performed on the input image. To date this work has concentrated on the modeling of faces for videophone and teleconferencing applications [29, 34]. In these coders, a 3-D wireframe model of a human head is stored at the receiver and the image of the sender's face is texture mapped onto the model. The sending machine tries to follow salient portions of the senders face such as several control points on the mouth, eyes, chin etc. This information is then transmitted (at very low bandwidth) and animates the computer graphic at the recipient end.

Other approaches to model based coding have been conducted at the Media Laboratory. Michael Bove [3] worked on using a 'range camera' to create images with $2\frac{1}{2}$D depth information, i.e. at every pixel not only was the luminance known but also the distance of the object from the camera. This information was used to convert the image into a 3-D database, both a more compact representation and also a representation which allowed the image to be viewed from angles different from that at which it was shot or with different lighting conditions. This is a powerful representation, but somewhat hampered by the requirement for the range camera. No such analysis could be performed on existing footage. John Watlington [54] also worked on synthetic movies, in this case from a 2-D model perspective. He developed a system known as 'VideoFinger', a variant on the Unix 'finger' command which created an artificial video sequence to represent the users logged into a computer system. Models of individual users were stored locally and animated according to information derived from the finger information.

It is systems such as these which are the closest precedents to the work in this thesis since they attempt to view the video image as consisting of objects rather than just signal levels. This approach allows for a much more compact and manipulable representation of information.

3.4 Implementation

Since the foreground objects such as Ricky and Lucy could not be adequately modeled in this work, standard signal techniques were drawn upon to compress them. Experiments were performed using models based on the JPEG and MPEG reference models, with the eventual architecture acting like a hybrid of these two. Only monochrome images were tested and issues of compressing colour information were not explored. Frames from the original were broken into groups of 12 or 24 frames. In this group the first frame was intraframe coded as the key frame. This key frame is used to provide a means of searching to a specific position within the movie without having to reconstruct too many intermediate frames to achieve this. It also is required to refresh the prediction loop; if this is not done then the energy of the difference images will rise as errors accumulate. The remaining frames were all predicted from the previous frame using a motion compensated prediction technique working at single pel accuracy. The difference image between the original and the reconstructed image was transformed using a DCT and the resulting coefficients multiplied by a visual weighting matrix and quantized. The output of the quantizer is the bit-stream. When used to code foreground images this coder works sometimes like a JPEG coder and sometimes like an MPEG coder. This is because for each block there may or not be a previous block from which to make a prediction. If there is no previous block, we are acting like JPEG and coding an ordinary picture; if there is a previous block then we are coding a difference image and therefore acting like MPEG. In addition to the above, in some instances the image was filtered and subsampled in order to achieve further reductions in bandwidth.

The MPEG type model can achieve considerable compression of the video signal. It is able to compress pictures down to approximately 0.5 bits/pel with little degradation. Further compression is achievable at the expense of gradually diminishing quality. In order to fit the foreground signals into a 64Kbit/s channel the foreground must be compressed into 0.1 bits/pel, assuming that it takes up 25% of the picture. This number is arrived at by comparing the channel capacity with the image dimensions to be transmitted:

picture size	368 x 240
frames per second	24
pels per second	368 x 240 x 24 = 2100000
75% gain from modeling	2100000 x 0.25 = 530000
Bits/pel for 64Kbit/s	64000 / 530000 = 0.1

This describes a particular situation. There are several variables involved, some of which can be controlled and others which cannot. The primary indeterminate variable is the gain from modeling which can vary anywhere between 0 and 100% depending on the nature of the image. The spatial and temporal resolution of the image to be coded and also the bits/pel of the coding are variables which can be controlled. These interact in that after a certain point it is better to degrade the image by reducing the resolution rather than decreasing the bits/pel of the MPEG encoder because the MPEG coder begins to produce extremely ugly artifacts when it is made to work under 0.1 bits/pel. If bandwidth is available, sometimes it is also beneficial to send a larger halo around the actors such that the blending can work more effectively.

Chapter 4

Scene Widening

The underlying principle in this work is the separation of the components of the scene and the creation of models for each component. In particular the background set is to be modeled and stored in the receiver. In order to do this a scene from the original is scanned by the computer. Often, in the course of shooting the stage will have been wider than what is visible in a single shot. At one moment we might be viewing the left half, at the next, the right (in fact, in 1950's TV shows there is very rarely a longshot that reveals the full width of the set, they were not constructed with enough height and support for such a shot). However, all the material has been derived from the same stage and we can create a single 2-D model of the set if certain conditions are met, most significantly that the pivot point of the camera should not move. If the camera pivot point does move then several background images must be stored, each representing a different viewpoint. When the scene is to be rendered, the closest view is chosen and transformed appropriately to model any differences in perspective. It can be seen that in such a situation there is a clear tradeoff between picture quality and bandwidth. If more bandwidth is available then many different viewpoints are modeled, if not then a smaller number of views are used to create an approximate rendering. However, there are many situations when the camera pivot does not move significantly and in these cases only a single model is required for the background and no extra distortion is added. This was the case for the material used.

We call the process of creating the single complete background set *scene widening.* This is an idea which could be useful for modern HDTV systems where the aspect ratio is greater than current television. If current television material is to be shown on an HDTV system then there is a problem of what to do with the extra width. This is the inverse to the problem of displaying movies on television today, where the original aspect ratio must be reduced. In the movies to television case, data must be cut from the image. In the television to HDTV case data must be synthesised. By modeling the set beyond the extent of what is in view in the current shot we have the data which is required for filling in those parts of the image. This will not always be the case, if we are at one of the extremes in a pan situation then we will not have extra data for beyond the edge we see. In such a case we must either offset the view centre, look to another scene from the movie where that area was seen, or perform an extrapolation. However, for much of the time the extra information required to synthesise the edges of the wider picture is readily available in the extended model of the background set.

4.1 Optical Issues

An optical representation is required for the extended scene which facilitates the mapping of the input sequence into the extended scene and eliminates distortion in this mapping. Conversely it must also facilitate the rendering of the extended scene into a particular requested view. The different forms of anamorphic optical mapping and the relationships between them are discussed by Yelick [55] and optics texts such as [19, 10]. For our purposes, the aim is to find an objective representation, independent of camera position, lens type, and angle of view.

The only representation to satisfy the first requirement is a full three-dimensional model of the set with complete lighting information. However, in two dimensions it is possible to make approximations to the ideal. A significant change in camera position will reveal new areas and occlude old areas. A 2-D model cannot completely account for this, but guesses can be made based on previous views or intelligent extrapolation. The success of such a scheme depends on the number of views available, the accuracy of the transformations used to generate the new view and the nature of the extrapolation procedures. Success at this is possible, but by no means guaranteed.

Combining images which have been shot using lenses of different focal lengths and from different angles is tractable, since the image that arrives from the world at the camera does not change, only the transformations which map that image onto the film plane change. Consider first the case of changing the focal length of the lens. It is sometimes considered that a telephoto lens distorts the perspective of an image, making objects appear flatter than they really are. An image taken with a telephoto lens is identical with an enlarged portion of the center of a ordinary image. The apparent distortion is due to the fact that the size of objects in the frame make you feel as if you are close to the content of the image while the perspective lines of the content are those of a distant objects [25]. This disparity appears as a flattening of depth, but the same disparity will occur when an ordinary picture is enlarged. Therefore, if we wish to combine images taken with lenses of differing focal lengths, or if we wish to render a scene with the appearance of a shot taken with at a particular focal length, we merely need to magnify or shrink the images around their centres.

When the camera direction changes the situation becomes a little more complicated. At this stage it would be useful to introduce a model of the optics. If we assume that the distance to the scene is significantly greater than the focal length of the lens and that lens distortions are not significant then a pinhole camera is a suitable model which can be used to simplify calculations. This world, shown in figure 4-1 is a cartesian space with the origin at the focal centre of the lens, the pinhole, and the optical axis collinear with the z axis.

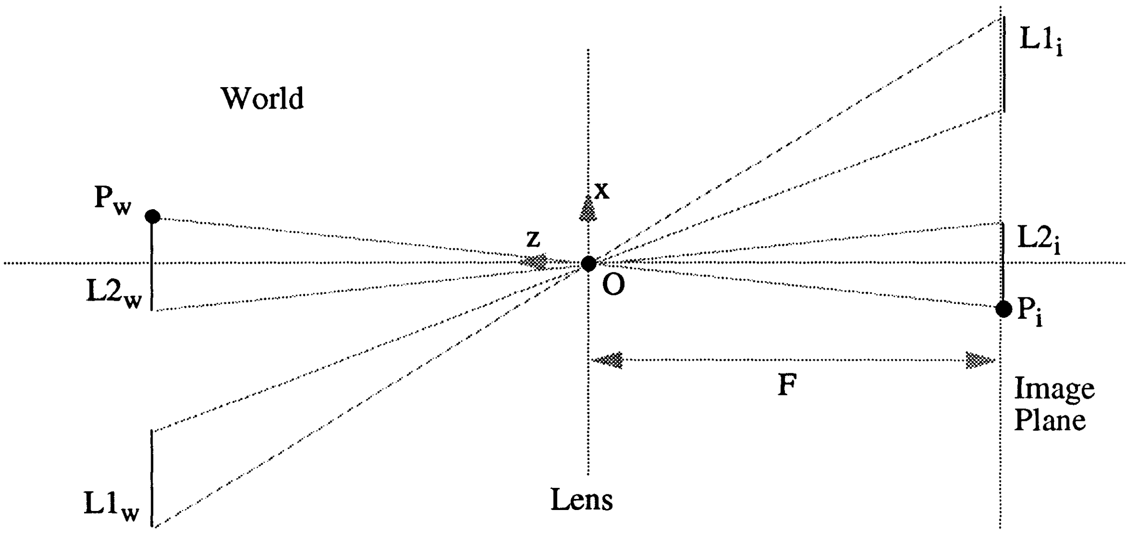

Figure 4-1: Optics of Image Capture

We view light rays as coming from a point $P_w(x_w, y_w, z_w)$, in the world, passing through the pinhole O, and landing on the image plane at $P_i(x_i, y_i, -F)$ (the image plane has a z coordinate of -F, the focal length of the lens). In the diagram this has been simplified into two dimensions, with the y dimension coming out of the page. The difference in lens focal length is reflected in the model as a difference in the distance from the lens centre to the image plane. A point P_w in the world will then be mapped onto a point P_i in the image according to:

$$P_i = \frac{F}{P_w.z} P_w$$

It is not acceptable to merely place different frames from a film side by side and paste them into one image since objects will appear different depending on whether they are viewed in the centre of the image or on its perimeter. It can be seen from the figure that the image L_i of the line L_w will have a different dimension depending on its position relative to the optical centre of the lens; $L1_w$ and $L2_w$ are of equal lengths, while the image of $L1_w$, $L1_i$, is longer than $L2_i$. We need a projection in which image dimensions are independent of camera angle. A spherical projection is an example of this. Such a projection is what would occur if a pinhole camera were to have a sphere (or more realistically, a hemisphere) as its image plane. A simplification of the spherical projection which is suitable in cases where the camera only pans horizontally is a cylindrical projection with the axis of the cylinder being vertical. This is the image representation chosen for the extended scene. Using this projection makes it possible to paste together elements from different frames with no distortion and provides a reference projection from which all views can be rendered with equal ease (or difficulty). In order to map from the planar projection of the film image to our 'objective' cylindrical projection we need to perform a transformation, and obviously we must perform the inverse on re-rendering. This nature of this transformation is best understood by analysing the geometry of the situation in figure 4-2.

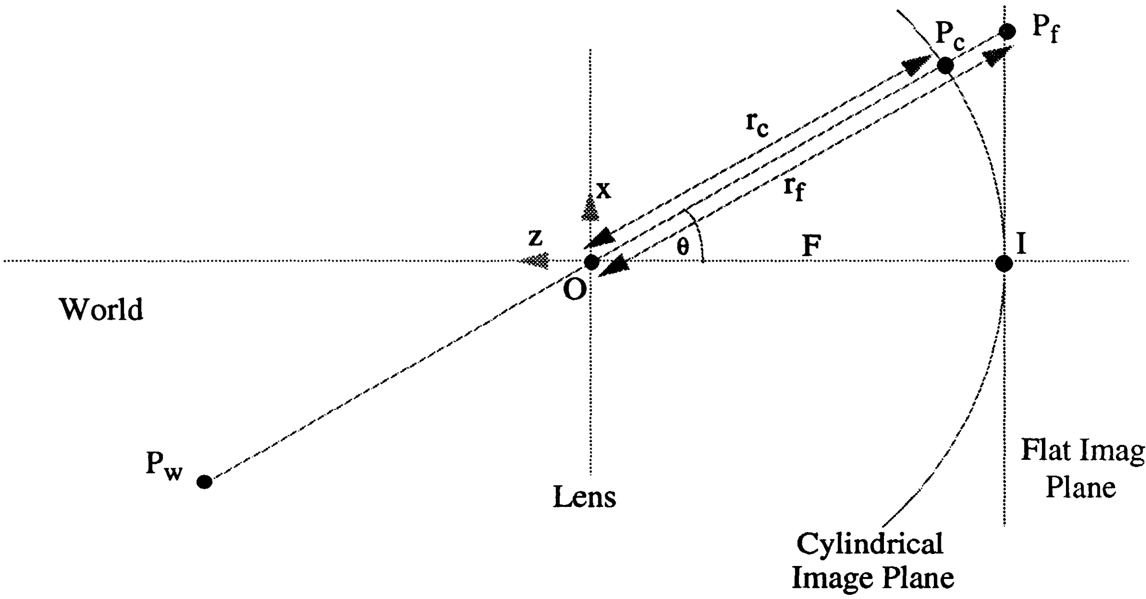

Figure 4-2: Planar and Cylindrical Projections

Two transformations will occur, one in the height of vertical features in the image and one in their width. The height of a feature in the image is proportional to the distance from the image plane to the centre of the lens. This gives us an equation for changes in height:

$$\begin{aligned} h_c &= h_f \frac{r_c}{r_f} \\ r_f &= r_c \, cos(\theta) \\ h_c &= h_f \frac{r_c}{r_c cos(\theta)} = h_f \frac{1}{cos(\theta)} \end{aligned}$$

Here h_c and h_f represent the height of a feature as it appears in the cylindrical and flat projections respectively. The width of features is proportional to their distance along the arc of the image plane:

$$
\begin{aligned}
w_c &= w_f \frac{IP_c}{IP_f} \\
IP_c &= r_c \theta \\
IP_f &= r_c \, tan(\theta) \\
w_c &= w_f \frac{r_c \theta}{r_c tan(\theta)} = w_f \frac{1}{tan(\theta)}
\end{aligned}
$$

Here, w_c and w_f represent the horizontal position of a point in cylindrical and planar projections.

Combining changes in focal length and direction of view takes a little care. If we wish to zoom in on an off-centre section of an image three steps must be performed. Firstly the image must be mapped from the flat projection to a cylindrical projection. Second the image must be mapped back to a flat image except with the centre moved to the new focus point. Now that the viewpoint has changed it becomes valid to magnify or shrink the image about its centre in order to simulate a change in lens focal length.

The above discussion avoids issues of resolution and aliasing. Whenever an image is resized, enlargement necessarily entails a loss of image resolution (or rather no gain of resolution with the increased size of the image) and reduction runs the risk of aliasing. Several approaches to the issue of arbitrary scaling of images exist in the literature, for instance [45, 24]. The algorithms employed in the transformations above used an averaging filter for reductions and bilinear interpolation for enlargements. The issue of loss of resolution in enlargement cannot be countered but aliasing was minimised by the filtering technique of the downsampler.

4.2 Automatic Scene Widening

To perform automatic background extraction three problems need to be addressed: compensation for camera motion and zoom, elimination of camera noise and removal of the actors. Compensation for camera movement was performed by analysing the input sequence and extracting a single pan vector for each frame. Details of this can be found in section 5.3. Work was not performed on compensating for camera zoom. Removing camera noise is relatively straightforward since we can assume that it is random and uncorrelated. If this is true then averaging a large number of frames will lead to the content constructively contributing to the average while the noise will randomly add and subtract intensity leaving a very small net value in the final image. Removing the actors is more complicated since they may be uncorrelated but are not random. A form of non-linear averaging is required which employs the heuristic that pixels which do not change significantly over time are background pixels. This has been explored in such applications as traffic detection where vehicles must be distinguished from the background [47]. The most stable pixels are given strong confidence measures which are propagated to neighboring pixels. The confidence measures of surrounding pixels is used to choose in cases where there there is less certainty about the value for the current pixel. A problem is in distinguishing differences due to camera noise, which should be averaged out, and differences due to occlusion, where no averaging should take place since we do not wish the occluding object to contribute to the intensity at that point. The most difficult areas are those where an object occludes the background for a long period of time. In such cases it is extremely difficult to judge what the background pixel value should be. What is needed in such cases is a means of distinguishing spatially that certain objects might belong to certain categories, for instance a 'people detector'. Currently this is not a feasible task.

An automatic scene extender was created which performed well for the majority of the image but failed in cases where a particular person was stationary for a long period of time. For instance, figure 4-3 shows the results of running the algorithm on two Lucy living room scenes. The first is taken from a sequence in which Ricky moves very little, revealing no background in a significant area above the icebox. In the second the majority of the image is sharp and free from noise with the exception of the area to the right of the sofa where Ethel was standing for a long period of time and a similar region above the left side of the sofa.

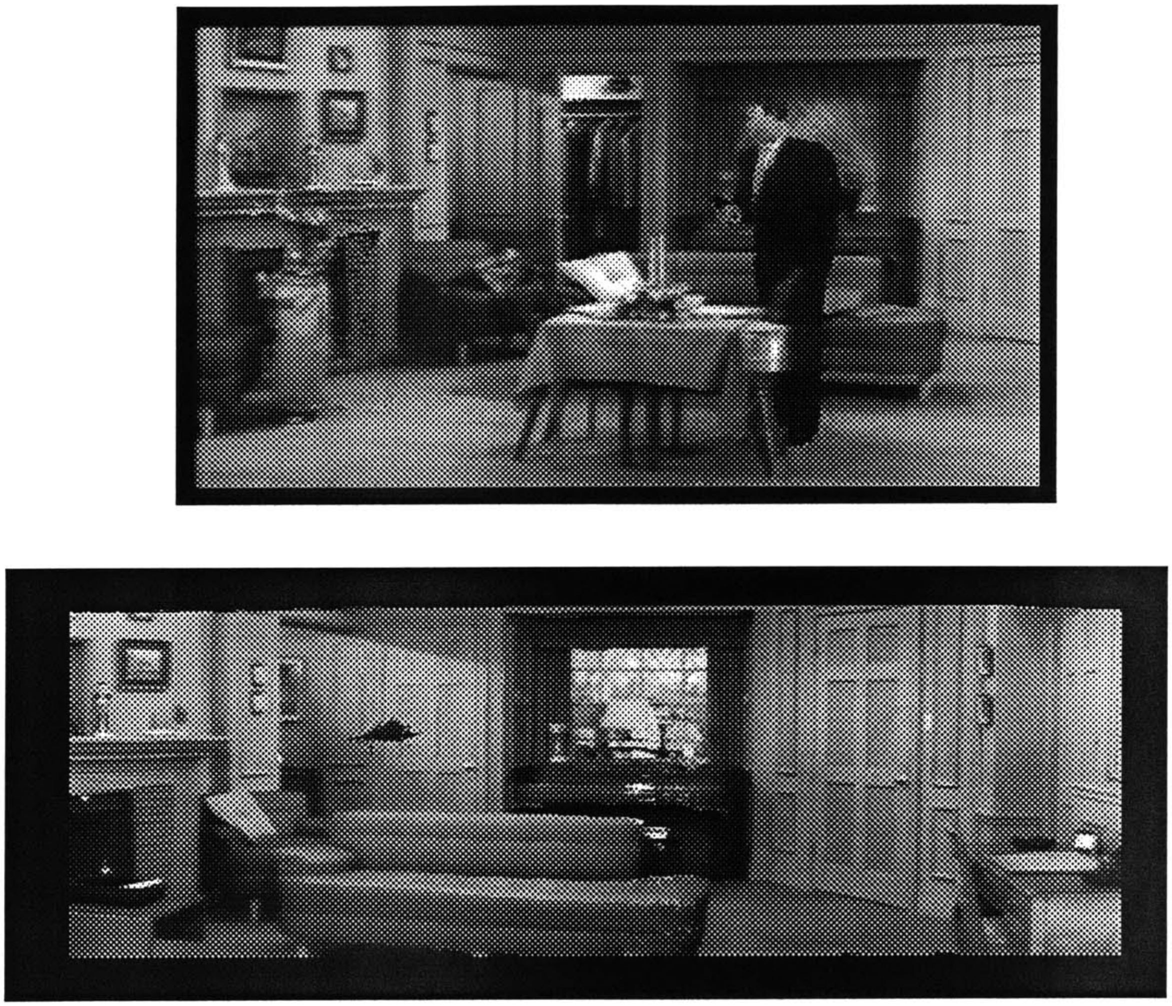

Figure 4-3: Automatic Scene Widening. The problems are in the areas where people never moved

4.3 Manual Scene Widening

Although the automatic scene widener worked well for the majority of the image, the problem of actors who do not move remaining in the image is a serious one. The segmentation software relies on having a background scene without any actors and will fail if this is not the case. Due to this, the scene widening had to be manually assisted. A tool was built which presented images from a sequence (predistorted into a cylindrical projection) to the user. The user can cut any part of any frame out and paste them together in a single background image (see figure 4-4). In addition to this, the same tool can be used to artificially create detail for the areas where no detail is revealed. This is done by extending spatially adjacent areas. Although the result of this shows obvious artifacts, it allows the segmentation software to succeed in these areas and in the reconstruction the region will normally be occluded (see figure 4-5).

In a typical sequence the final image will have components drawn from ten to twenty source frames. An image created in this way tends to show edge artifacts where elements from different frames have been placed adjacent to one-another. In order to deal with this problem, the tool was modified such that it also created a history list of the actions of the user. After a full picture was manually created the computer made a second pass, performing the same operations as the user but blending each piece together according to the multi-scale blending algorithm described in chapter 7.

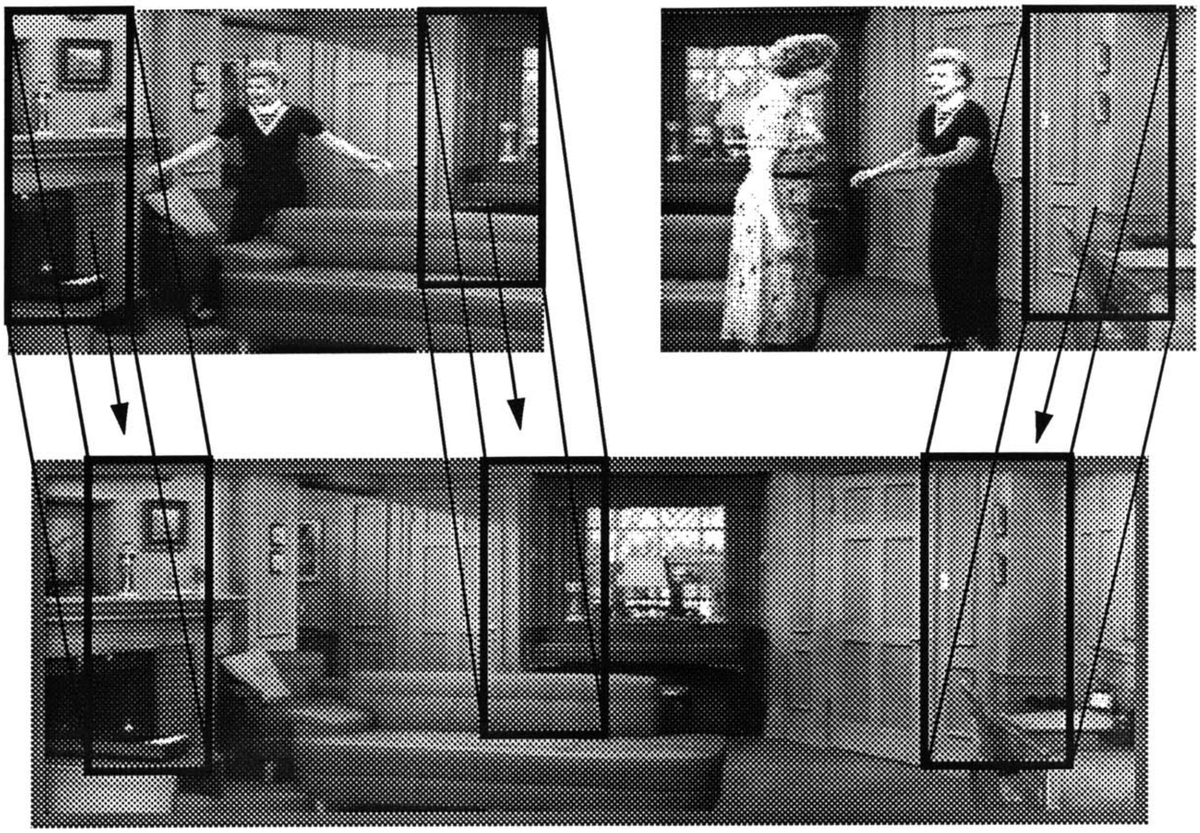

Figure 4-4: Manual Scene Widening - Cut and Paste!

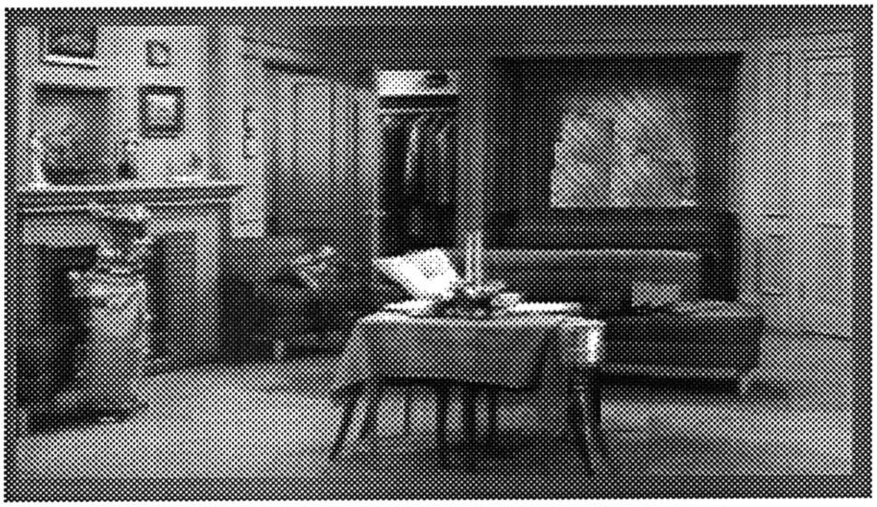

Figure 4-5: Manually extended living room scene. The area above the ice box is entirely artificially created since Ricky never revealed the background behind him

Chapter 5

Motion

An estimate of the motion of the objects in the scene is very important for several parts of the thesis. Firstly, the motion of the background set must be found so that the model can be animated correctly at the receiver. This animated scene is also required for use as an actor-free background for the segmentation algorithm. Also, in some instances the segmentation of the actors from the background relied on using the difference in their motion from that of the background. Several different methods for motion estimation and analysis were investigated, which will be discussed below.

5.1 Motion Estimation Techniques

A motion estimator is attempting to determine the *optical flow field* of an image sequence. This is the projection of the three dimensional motion of the scene onto the image plane, resulting in a 2-D field of 2-D vectors, each one indicating the translational motion in the image plane of the object at that point.

If we make the assumption that lighting is not directional and temporally and spatially uniform (a strong assumption, but one that is often an acceptable approximation in small regions) the optical flow field can be defined by the optical flow equation:

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Expanding the right hand side using the Taylor series and dropping the high order terms we get:

$$\frac{\partial I}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial I}{\partial y}\frac{\partial y}{\partial t}=-\frac{dI}{dt}$$

$$\left[I_x' I_y'\right]\left[\begin{array}{c} V_x \\ V_y \end{array}\right]=-I_t'$$

Intensity gradients x *2-D velocities = Frame difference*

This is a single equation with two unknowns and therefore underconstrained. This problem is known as the aperture problem and heuristics must be applied to obtain a solution, a common one being assuming that the motion field is smooth. There are several classes of technique for finding a solution:

- Gradient techniques. These search for a solution to the above equation by determining the local spatial and temporal gradients.
- Energy based techniques. These are biologically based models which use filters tuned to detect oriented motion energy.
- Token matching techniques. These attempt to trace the motion of key features in the image such as edges.
- Block matching techniques. These compare two images on a blockwise basis looking for the set of local displacements which will maximise the correlation between the images.
- Frequency Domain Correspondence. These search for phase changes in the frequency domain to indicate motion in the spatial domain.

A number of factors are important to consider in comparing these methods. What is the density of the motion fields they produce? How well do they perform within and at the edges of moving objects? What is the complexity of the algorithm and could it be realised in real-time hardware (important for an image coder)?

5.1.1 Image Gradients

A popular method class of motion analysis techniques is based on the computation of image gradients in space and time. The earliest such implementation was performed by Limb and Murphy [28] which computed the sum of the frame difference signal of the whole image and divided this by the element difference where this was non-zero (moving areas). The frame difference is an approximation to the temporal derivative of the image $\frac{\partial I}{\partial t}$ and the element difference is an approximation to the spatial derivative $\frac{\partial I}{\partial x}$. Applying the chain rule leads to the equation for determining image velocity $\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}\frac{\partial x}{\partial t}$. This is somewhat simplistic since it assumes only a single moving object within the frame but the central idea is common to all gradient based techniques. The method was extended to deal with multiple motions by Fennema and Thompson [7]. The problem here is that using only the first derivatives one can only compute the component of the 2-D velocity parallel to the image gradient. They reformulated the problem, showing that the derivatives constrain the velocity to lie in a cosine curve in polar velocity space in which one axis represents direction and the other magnitude. If many image points belong to the same object then their velocity constraint lines will intersect and uniquely specify both components. The intersections were found by drawing these curves into an accumulator array and searching for peaks. This is the fault of this method in that peak finding only assigns a single vector for each object, a result that will be inaccurate in all but cases of simple translational motion.

Horn and Schunk [22] formed an elegant and concise solution to the incompleteness of the gradient equation and described clearly when the algorithm could be expected to work. They made two main assumptions. The first is that $\frac{dI}{dt} = 0$, the brightness of any point in the world does not change. The second is a continuity constraint that the image intensities must be differentiable in both space and time. Both of these assumptions are problematic, the first because lighting conditions will change and the second because discontinuities are common in images at the borders of objects. In addition to highlighting these necessary assumptions they proposed a smoothness constraint that guaranteed a solution to the matching problem. Their technique was iterative and may not have been the most efficient approach but it formalised several important issues in gradient based techniques and indicated clearly when they would fail.

Image gradients can be approximated using finite difference methods such as the above as long as the signal is bandlimited and noiseless. However, real images contain significant noise levels which are enhanced by simple differentiation schemes such as Horn's. A variant on the above techniques was developed by Martinez [32] and Krause [26] which tackles some of these problems. Theirs is a parametric approach; a region of the image is approximated by a polynomial of a much lower order than the degrees of freedom of that region. This polynomial is of the form:

$$I(x,y,t) \approx \tilde{I}(x,y,t) = \sum_{i=1}^{N} S_i \phi_i(x,y,t)$$

I are the real image intensities while $\tilde{I}$ are the intensities as approximated by the model. The signal model is defined by the basis functions ϕ_i which Martinez proposed as a set of three-dimensional polynomials:

$$\begin{array}{lllllllll} \phi_1 & = & 1 & \phi_2 & = & x & \phi_3 & = & y \\ \phi_4 & = & t & \phi_5 & = & x^2 & \phi_6 & = & y^2 \\ \phi_7 & = & xy & \phi_8 & = & xt & \phi_9 & = & yt \end{array}$$

The matrix of coefficients S is calculated by minimising the error between the actual intensity values I and those predicted by the model $\tilde{I}$. The desired gradients of $\tilde{I}$ can then be calculated from the symbolic differential of ϕ and the coefficients S_i.

$$\frac{\partial \tilde{I}}{\partial x} = \sum_{i=1}^{9} S_i \frac{\partial \phi_i}{\partial x}\bigg|_{(x=0,y=0,t=\frac{1}{2})} = S_2 + \frac{1}{2} S_8$$

$\frac{\partial \tilde{I}}{\partial y}$ and $\frac{\partial \tilde{I}}{\partial t}$ can be found in a similar fashion. Typically a small spatiotemporal region (a 5x5 block x 2 frames) is modeled resulting in an overconstrained set of linear equations: 50 samples to estimate 9 parameters. Because the system is overconstrained in this way, its noise immunity is considerably increased over other techniques. In addition, Krause showed that the computations could be effectively implemented in hardware.

5.1.2 Energy Based Models

Recent progress in neurophysiology and perception has spawned a number of techniques attempting to replicate the organisation of the eye into spatio-temporal frequency bands. Adelson and Bergen [2] look at motion as orientation in the three dimensional x-y-t space and propose filters that are sensitive to these orientations. The first stage consists of linear filters oriented in space and tuned in frequency. These are organised in quadrature pairs and their outputs summed to provide a measure of motion energy in a single direction. These can be further combined as opponents to produce energy detectors which respond positively to motion in one direction and negatively to motion in the opposite direction. Heeger [20] demonstrated a motion energy model which combines the outputs of twelve motion-sensitive Gabor filters in each spatial frequency band. He also formulates a measure of image-flow uncertainty which he postulates can be used to recognize ambiguity due to the aperture problem.

5.1.3 Token Matching

Token matching was first proposed by Marr and Poggio [30] with regard to stereo matching, a computationally similar problem to motion estimation. Marr considered the human visual system to be operating on the "$2\frac{1}{2}$-D sketch" occupied by image tokens such as oriented lines and blobs. Ullman [52] also argued against using raw image data rather than tokens as the elements for correspondence (however, he and Marr later presented a model closer to the gradient and energy models described below [31]). Such schemes have not been continued in their basic form but rather integrated into grey level matching schemes either by performing block matching on the edge image or using edge data as a confidence measure in the motion estimator.

5.1.4 Block matching

Block matching is the most common technique for estimating the optical flow field in image coding, mostly because computationally it is relatively simple. Current techniques for determining this field usually make three assumptions:

1. All motion is assumed to be 2-D translation in a plane parallel to the plane of the camera.

2. Lighting is both spatially and temporally uniform.

3. There is no revealed or occluded area.

These assumptions do not hold over lengthy periods of time but are accurate enough on a frame to frame basis to justify their use.

Block matching techniques were first proposed for image coding by Jain and Jain [23], and are well described by Netravali and Haskell [38]. The current image is analyzed on a block by block basis against the preceding image with the aim of finding which block in the preceding image most closely resembles the current block. The block of pels is shifted across an area of the previous image and a difference measure taken at each displacement. The motion vector is derived as the displacement shown to yield the lowest difference measure. There are several error measures which can be used, for instance the correlation coefficient, mean squared difference or absolute difference. These three are in decreasing order of computational complexity and accuracy. Experimentally it has been observed that the choice of method does not have a significant effect on the amount of searching or the final estimate for motion so the absolute difference value is usually chosen for reasons of simplicity.

A naive scan method is to scan all the blocks within a given range. However the amount of computation required can be considerably reduced by using methods which assume a uniform image gradient as an aid to finding the vector. Three common schemes exist: 2-D logarithmic search, N-step search and conjugate search. The motion estimator used for this work employed a 4-step search whose operation is as follows. Eight pels in a square of side 2N around the search center are tested in the first step (see figure 5-1). In the second step eight pels are similarly searched but centered on the minimum of the previous step and with a narrower search width. This process is repeated until the search width reaches zero. The vector is chosen as the minimum point of the final scan.

The accuracy of the above methods is limited to a single pel. It is possible to perform fractional-pel motion estimation by creating a grid in the search frame which contains interpolated values at intervals of the accuracy of the desired result. To do this requires a considerable increase in computation time and has been shown by Girod [12] to exhibit decreasing marginal returns; half pel accuracy estimation is an improvement over full pel accuracy but at a quarter pel accuracy little further gain can be seen.

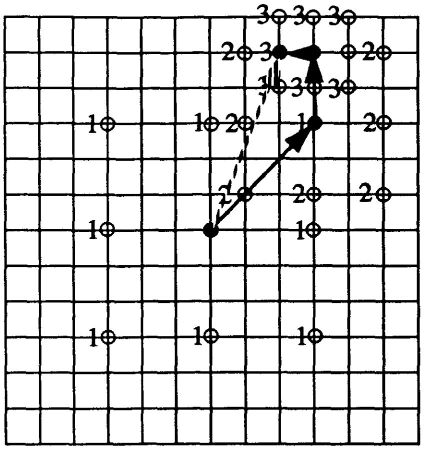

Figure 5-1: Block Matching. N-Step Search Pattern

5.1.5 Frequency domain techniques

Block matching is a localized operation. A disadvantage of this is that only a single vector is produced for any single block in the image and displacement is hard to measure at discontinuities in the optical flow field. An alternative approach to estimating displacement vectors discussed by Girod [13] is to estimate the probability density function (PDF) of the displacement vectors. The PDF can convey important information about the global motion: if the image contains only a single moving object, the PDF will display only one impulse, multiple moving objects are seen as multiple peaks. The PDF of the displacement vectors can be derived from the spectra of two images using the relation:

$$P(\omega_x, \omega_y) = \frac{\Phi_{10}(\omega_x, \omega_y)}{\sqrt{\Phi_{00}(\omega_x, \omega_y)\Phi_{11}(\omega_x, \omega_y)}}$$

Φ_{00} and Φ_{11} are the fourier transforms of images 0 and 1. Φ_{10} is the cross spectrum of the two images, the fourier transform of one is multiplied by the complex conjugate of the transform of the other. Essentially, $P(\omega_x, \omega_y)$ represents the difference in phase between the two images. In a situation where there is solely translation within an image then there will be no change in the overall energy of the signal, merely a change in phase. The output probability density function will indicate how many objects are moving within the frame and with what velocities. However, in order to find out *where* these objects are in the frame it is necessary to return to a localized technique such as block matching.

5.2 Implementation

The motion estimation for this thesis was performed by an N-step search block matching method. The reason for this was that it is computationally straightforward and proved to provide adequate results for the estimation of the motion of the background, the most important measure required for this work. It is not surprising that it was successful since only a very coarse estimate of motion is being searched for. However when motion was used to segment the images into individual elements (as described below) the performance was considerably poorer and this may in part be due to the inaccuracy of the output of the motion estimator at finer scales and at the edges of the moving objects. Greater success might be gained by implementing some of the other motion estimation techniques described above and analysing their relative abilities to identify motion boundaries,

5.3 Extracting Camera Pan

Once an approximation to the optical flow of the image was available attempts were made to derive the motions of the various objects in the set from this. Most important is the extraction of a single pan figure for the predominant motion, assumed to belong to the set. Since the motion was derived from a localised technique, it had to be scanned in order to create the histogram of the probability density function. Each coordinate of the histogram represents a direction of motion, and the height of the histogram at that point represents the number of vectors in the image judged to be moving at that velocity (see figure 5-2).

The histogram is analysed for peaks by using an algorithm which uses the analogy of a draining lake. The water level is iteratively dropped and as each new peak appears it is checked to see if it is a distinct peak or whether it is part of an already exposed peak. The process is stopped when a predetermined number of peaks have been found or when the water level drops below a threshold.

Performing this analysis on histograms from the Lucy sequence derived by block matching proved to find the predominant motion of the set with some accuracy but not other parts of the

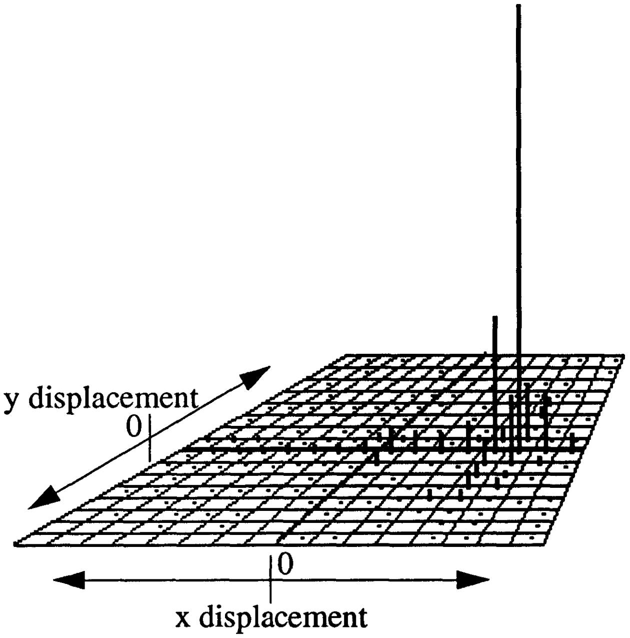

Figure 5-2: 2-D Histogram of motion vectors

scene. People moving in the foreground do not conform well enough to the assumption of rigid body translation to create a coherent enough peak in the histogram to identify them accurately by this method.

Chapter 6

Segmentation

In order that models can be created of the individual components in the scene it is necessary that these components be identified. This is the segmentation problem. A great deal of research has been performed in machine vision to tackle this problem, some of which will be discussed in relation to how it might assist this work. However, much of the difficulty of segmentation has been sidestepped by extracting the background set by methods described earlier. Once we have an image of the set without any actors then we only need to compare the original against that background and where we are confident that the two differ we describe those components as the actors. This is a much simpler task than performing an analysis within the frame and attempting to group together areas with similar characteristics, the basis of most segmentation algorithms. However, if machine vision techniques are used in parallel with the background comparison this makes the segmentation more robust and better able to deal with some ambiguous situations which exist for background comparison alone.

What are the characteristics of a good image segmentation? Regions should represent consistency in some characteristic, either in a signal sense representing perhaps uniform texture, or in a content sense representing uniform 'Ricky-ness'. Regions should be contiguous without small holes and their boundaries should be simple, unragged and spatially accurate. Achieving these aims is difficult because strictly uniform regions will tend to be ragged and dotted with small holes. Also, sometimes separate items will have similar characteristics causing a merging of regions which are distinct. Image segmentation techniques tend to be heuristic in nature and vary in the compromises they make and properties they emphasise.

The segmentation masks needed for this work must have several characteristics. A spatially accurate division of the image is the most important. In addition, assumptions about the nature of the expected masks can be a great help, such as assuming that the edges should be smooth and the mask should not contain internal holes. Sometimes these requirements conflict. If Lucy has her hand on her hips then there will be a hole between her elbow and her body. In such a situation the requirement for spatial accuracy conflicts with the requirement for no holes. Another important point concerns what is to be considered a component. Ideally we wish to be able to distinguish each independent item in a scene. However, is the bottle of champagne in Ricky's hand a part of Ricky, or independent from him? I guess we should ask. For the initial segmentations this point bore little relevance since the only distinctions made were between 'background' and 'foreground'. However more sophisticated versions of the coder ought to have a more complete understanding of the scene. Since it is difficult to define a homogeneity measure for 'Ricky-ness' or 'Lucy-ness', attempts at reaching these goals were tackled mostly in image rather than measurement space.

6.1 Machine Vision approaches to segmentation

6.1.1 Clustering and Region Growing

Haralick and Shapiro [18] have classified segmentation techniques and distinguished between techniques that look to find clusters in the measurement space and those that look to find regions in the image space. There is an interplay between these two domains which can be exploited in different ways. An emphasis on clustering in measurement space will result in more uniform and homogenous segmentations but will also display unwanted spatial characteristics such as holes and ragged edges. An emphasis on the spatial domain will conversely lead to smoother regions but which contain more widely varying data.

The most straightforward method for clustering in measurement space is by creating a histogram of some measurable feature of the image, such as grey level. Groups are searched for in the histogram, divided and then this division is mapped back into the image space. This method works in cases where the image consists of a small set of distinct objects but will fail in more complex cases because the measurement characteristics of objects will sometimes overlap and important spatial distinctness relationships are being ignored which might otherwise disambiguate the overlapping data.

To take advantage of spatial relationships in the image a technique called *region growing* is used. In single linkage region growing schemes, at each pixel every neighbouring pixel is searched and joined to the current pixel if its properties match a similarity criterion. The image segments are the complete set of pixels belonging to the same component. This scheme is attractive due to its simplicity but is very susceptible to creating false merges due to the localisation of its operations. Some hybrid techniques have been developed which make the measurement value of a pixel be dependent on its neighbours such that they represent some aspect of their region rather than solely their own value. In particular an edge operator is passed over the image and whether a pixel is considered an edge or otherwise affects the way in which it propagates its value to others. Like many other techniques the performance of this varies with the type of image being analysed and the quality of the edge operator. Centroid linkage region growing is another technique where a table of all current segments is stored and updated as the image is scanned in a regular manner. If the new pixel value matches a known adjacent segment, it is added to that segment and the segment value is updated, if it does not a new segment is created. This can be a memory intensive operation but takes a more global view of each segment and has the advantage of being able to place boundaries in areas of weak gradients since it is not comparing pixel values against their neighbours but against the average for the whole region. A different technique again for spatially based segmentation, first suggested by Robertson [51], is the split and merge algorithm. Initially the whole image is considered to be a single segment. This segment is recursively split if the region exceeds a homogeneity threshold. A last pass is then made merging regions which are similar but were never tested by the region-splitting. Lastly, it is possible to determine image segments by combining clustering in measurement space with spatial region growing. Haralick and Kelly [17] suggest that this should be done by first mapping all the peaks from the measurement space histogram into the image domain. These points are then propagated using spatial region growing while ensuring that it does not grow into a region belonging to another peak. Aach et. al. [1] have a similar technique which first implements an object detection scheme in measurement space and then performs a contour relaxation in image space to obtain a better mask. A similar technique is also used by Ohlander et. al. [39].

6.1.2 Detecting Motion Boundaries

All of the above techniques are based on still image analysis. Another set of segmentation methods look to the task of identifying objects by their motion, i.e. segmenting the optical flow field. Here the assumption is that moving rigid bodies will show up as regions of distinct velocity in an optical flow field. The methods for detecting boundaries can be classed as those which detect boundaries before, simultaneously with, or after the calculation of the optical flow field.

There is a important reason for detecting motion boundaries before computing the optical flow field. This relates to a paradox inherent in current methods for computing visual motion, between intensity based methods and block-matching methods. Intensity-based methods have to integrate the local motion measurements due to the aperture problem (see section on motion) and this is typically performed by assuming the flow field varies smoothly. This assumption is true everywhere but at motion boundaries such that errors are highest at the very locations we are attempting to find. Block-matching schemes produce sparse flow fields. These flow fields must be interpolated such that edge detectors can be used to locate the motion boundaries. This process, however, can smooth over the fields to such an extent that the boundaries are no longer evident. This is the dilemma: in order to detect boundaries with edge detectors we need an error free and dense motion field; however, to compute such a field we require a knowledge of the boundaries. Spoerri and Ullman [48] suggest methods for the early detection of motion boundaries to solve this problem. One is *bimodality tests*, where a local histogram is created for each point. If this histogram displays two peaks then it indicates that the pixel is on a motion boundary. Another is the *Kolmogorov-Smirnov* test, which uses a statistical test based on the fact that the populations of motions are different on each side of the boundary. A third method, the *Dynamic Occlusion* method, is based on the fact that motion boundaries tend to be areas where background area is occluded and revealed regularly. Other methods have been proposed, such as that of Reichardt and Poggio [43] where direction selective movement detectors inhibit flicker detectors when the same movement appears in the center and surround of the movement detectors. Flicker detectors with significant activity then indicate the presence of motion boundaries. Hildreth [21] uses the flow component in the direction of the intensity gradient, the *normal flow component* to detect motion boundaries. This is based on the fact that if two adjacent objects have different motions then their normal flow components will change in sign or magnitude across the boundary.

Several techniques exist to detect discontinuities after the computation of the flow field. Nakayama and Loomis [36] detect motion boundaries using a center-surround operator. Potter [42] uses region growing techniques to group features of similar velocity, making the assumption of translational motion. Thompson et. al. [50] show that object boundaries can give rise to discontinuities in the image flow field and these could be detected as zero-crossings in the Laplacian of the components of the flow field. Terzopoulos [49] detects discontinuities in sparse surface representations by marking locations where the thin plate used to interpolate between the sparse data points has an inflection point and its gradient is above threshold.

6.1.3 Segmentation by comparison against the background set

Machine vision approaches to segmentation such as those described above tackle the segmentation problem with few clues as to what is being searched for. However, if a model of the background set exists and we are trying to extract foreground objects then the task is made considerably simpler. The task becomes one of performing a comparison between two images and extracting those parts which differ. The problems which arise in performing this task are threefold. One is that the original image sequence has considerable time-varying camera noise such that individual pixel values in the original will rarely exactly match their partners in the set image. The second is that the background set construction process does not guarantee the creation of a perfectly accurate model such that some image features will be misplaced or contain pasting artifacts. The third problem is in many ways more serious, this is the problem that variations in the intensity values in the picture arise not only from objects moving within the set but also from their shadows. If we are searching for significant differences between our set image and our current image then shadows will often show up as being distinct objects.

6.2 Segmentation results

Two methods were explored to perform the segmentation: analysis of correlation measures between the current frame and the stored background image and analysis of the motion field of an image. The output of both analyses are extremely noisy, so work was also performed on filters to reduce the noise in the segmentations. When combined, the output of the segmenting software and the noise reduction software almost always would catch the actors, the main problem being that it would also catch much of the background surrounding them.

6.2.1 By motion

One method of segmenting out foreground from background is by looking for regions of distinct velocity in the image and making the assumption that each of these belongs to a different object. The original image sequence was analysed and a sparse motion field output using a block-matching motion estimation technique. A histogram was created from the motion vectors and this histogram searched for peaks. These peaks were mapped back into the image space and the results of this fed into the noise filters described below. The results of this process had many problems. The most obvious is that when the actors are not moving they become invisible, something of an undesirable trait. Also, people do not fit into a rigid body model and so their motion fields do not form into distinct areas. They can however be identified when compared with a non-random field, such as a still background. In such cases the actors were easily identifiable and formed clear segmentation masks. However, there are enough instances where this technique is incapable of accurate identification that it was not considered to be useful as a general tool.

6.2.2 By correlation

The main technique used to create the segmentation mask which distinguishes between background and foreground is the measure of the correlation between blocks in the original image and blocks in the actor-less 2-D background model. Once a single image of the background exists and is animated according to the analysis of the motion in the original, two sequences are available for comparison. These two sequences ought to be identical in every way except for the fact that the reanimated sequence has no actors. Extracting the actors is then performed by the process of comparing these two and where they differ we assume that we are seeing the foreground image. In reality, of course, many problems arise. The animated background contains lighting errors and distortion errors and the original contains noise. Simply subtracting the two images from one another creates an image with significant energy in parts of the background. However, if rather than calculating absolute differences we calculate the correlation coefficient between the two images on a blockwise basis the ability to distinguish between genuine differences and noise is considerably increased. The correlation coefficient r is the normalised covariance between two images, defined as follows:

$$\begin{aligned} r &= \frac{C}{\sigma_0 \sigma_1} \\ C &= E\{(x_{0i} - \eta_0)(x_{1i} - \eta_1)\} \end{aligned}$$

where η_b is the average value of block b

σ_b is the standard deviation of block b

x_{bi} are the pixel values in block b

This equation returns a value representing the difference between two blocks but which is independent of the strength of the values being compared; the judged difference between two images will not be affected by their lighting conditions. The output of this stage yields a segmentation such as that in figure 6-1. The actors are captured but so is a great deal of the background. This creates the requirement for post-filtering the masks (see below).

6.2.3 Combining Multiple Resolutions

Both the motion based segmentation and correlation based segmentation experiments were performed using different values of block size and threshold. The more significant of these two variables is undoubtedly the block size. If a small block size is chosen then a high resolution mask is created but the image contains a great deal of noise where the algorithm has been mistaken in its decision. The reason for this is that with a small blocksize it is much more probable for two images to randomly correlate than with a much larger blocksize. At the larger blocksizes the noise could be completely eliminated but only a very crude segmentation mask

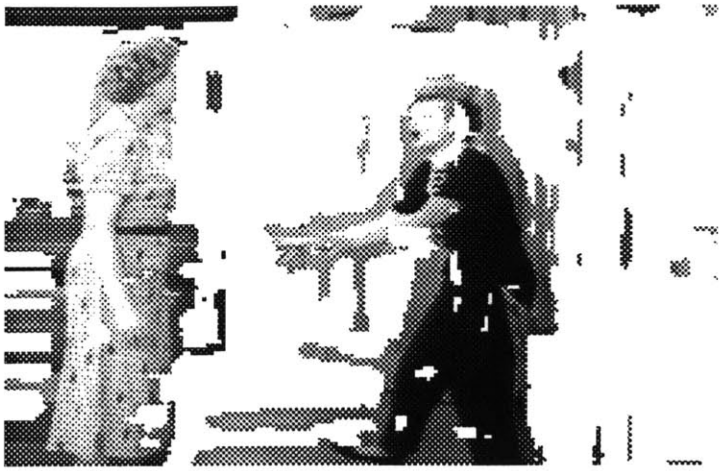

Figure 6-1: Segmentation output of the Correlation Stage

would be created with poor spatial accuracy. Obviously what is needed is somehow to combine the noise immunity of the large block size method with the high resolution of the small block size method. Two different solutions were explored.

In the first the image was analysed at a range of blocksizes, starting with the largest. At each iteration the image was only analysed in the vicinity of the regions indicated by the previous pass. For instance, the image would be analysed looking at correlations on a 32x32 blocksize grid. As was previously mentioned this produces a crude but noise free image. The next pass will look at a 16x16 grid, but only in the vicinity of the areas marked by the previous pass. This carries on down to 4x4 or 2x2 by which time a mask has been created which combines the characteristics of the noise immunity of the high blocksize correlation measure with the resolution of the low blocksize measure.

In the second all measurements are done at large blocksizes (16x16, 32x32) but rather than tiling the image as is done by the approaches previously mentioned the block slides along a small number of pixels at a time, typically 2. This approach is highly intensive on processing power and provides reasonable results, except that it has the property of integrating the mask at a boundary. A smooth mask is created but it surrounds all foreground objects with a 'halo' with a width equal to half the blocksize.

6.2.4 Filtering the masks

Whatever the technique used to make the initial segmentation decisions, the resulting mask tends to have several undesirable properties which can be considered as noise. Several heuristics were applied to reduce this noise and these are listed below:

- Removal of regions below an area threshold. This only allows large objects to be considered as viable segmentations. This is very effective at extracting the people but means that small objects will always be ignored.
- Flooding of holes surrounded by larger regions. This creates much cleaner masks by assuming that random holes were created by noise rather than correlation. Usually this is very effective but there are situations where real objects contain holes, the filter is a disadvantage at such times.
- Applying a time constraint that only allows the mask to move a limited distance from frame to frame (a momentum constraint). This is very useful in supressing temporal noise and is almost always consistent with the behaviour in the real image. Its only disadvantage is that if it is initially fed with a bad mask then it will perpetuate the initial errors. Manually touching up the key frames will stop this from occurring.
- Removal of apparent artifacts of the analysis methods, e.g. thin vertical and horizontal lines.

The application of these filters was essential in creating usable masks for the coder, and in most cases the filters led to a net improvement of the mask. This can be seen by comparing figure 6-2 which is an example of the foreground as seen after filtering with the same figure before filtering in figure 6-1 earlier.

6.3 Encoding the segmentation

The reason for segmenting the image is to reduce the bandwidth in a video encoder. For this reason it is important to consider how the segmentation information is to be encoded in such a way as to add minimal extra bandwidth to the bitstream. One method might be to encode

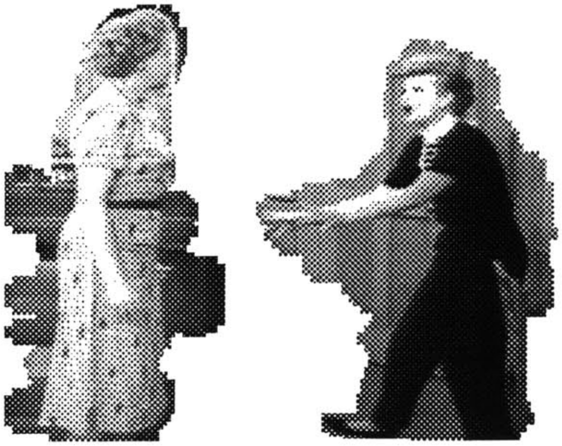

Figure 6-2: Segmentation output after filtering

the segmentation implicitly within the codes for image intensities as formulated by Mok [35]. Another is to merely code the segmentation mask at a much lower resolution than the image. This makes sense if the foreground is being encoded by a fixed-block size block based encoder such as JPEG. Only one bit is required for each 8x8 block to indicate whether it is being transmitted or otherwise. This code itself can be run length encoded to reduce bandwidth further, adding very few bits to the data stream.

Chapter 7

Reconstruction

7.1 Issues

The encoder has divided the picture into parts. The receiver has the job of putting these parts back together again. There are two issues which make this problematic. The first concerns placement, the second lighting. The estimates of motion in the original will not be completely accurate. In scenes where there is a large amount of conflicting motion, the motion estimates might be several pixels in error and even in the best cases there is a limit to the accuracy of the estimator (one pixel in the case of the implementation for this thesis). It is possible to perform motion estimates at sub-pixel accuracies; this was experimented with and gave results that were an improvement over single pixel estimation (at a much greater computational cost) but other errors tend to render the measurement accuracy less significant. The problem with lighting is that the single model of the background set which is constructed does not contain changing lighting information. As the actors walk about the set they cast shadows and this is not accounted for. If the foreground image of the actors carries around a halo of the background, as is usually the case, then the lighting conditions in this halo will not be consistent with the lighting conditions in the static image of the set. The result of this is a very visible discontinuity surrounding the edge of the foreground image when it is placed against the background. In the case where an actor is being placed against a background from which he or she was not derived, performing a good blend is even more important.

7.2 Placement

The placement problem was tackled by using the halo of background surrounding the actors to our advantage. This halo contains useful registration information which can help to indicate to the software from where in the background the actor was actually taken such that she or he can be placed back in that same spot. A strip was taken around the outside of the foreground image and this strip was moved in a search around the vicinity indicated by the motion estimate. The actual point of placement was chosen to be the point where the correlation of the foreground strip with the background was maximised. Using this method had variable results for two reasons. The first is that the strip only roughly represented parts of the foreground image that were actually background halo. If a more accurate mask could be made around the actor then the assumption that the surrounds represent background would be more valid. The main problem with this method is that random erroneous placements will lead to the actor jumping about in the scene in a most unnatural manner. An attempt at a solution to this would be to take a more global approach. The current software checks the position of the actor at every frame. A better approach would be to average results over a number of frames and place a continuity constraint on the motion.

7.3 Blending

The lighting problem displays itself as discontinuities in the recreated frame at the edges of the replaced actor. This is a blending problem. There are several approaches which one can take to blend one picture into another which vary in their complexity. The simplest is to use a weighted average of the two images in a transition region. The question that arises is what is a good width for a transition region. If the width is too small the boundary is distinct as a discontinuity in the DC value of the image. If the width is too large then small features will appear doubly exposed. What is required is a transition width of the same order as the feature size of the objects being merged. In an average scene there will be objects at many scales being merged, making a single transition width fail. One answer to this, and the technique used, is to use a method developed by Burt and Adelson [4] in which they divide the image into a laplacian pyramid. Forming a laplacian pyramid is a means of bandpass filtering an image; the image is divided into its components, each of which span one octave of spatial frequency. This fact can be exploited for the purposes of blending by merging the two images separately at each level of the pyramid. The transition width is kept constant for each image but the fact that the low frequency images are at a lower resolution than the higher frequency images means that they have a much wider effective transition width. This technique allows one to match transition width to the spatial frequency of the feature, creating a final blend with neither of the problems mentioned above concerning a blend using a single weighting function.

An example of a scene reconstructed in this way is shown in figure 7-1.

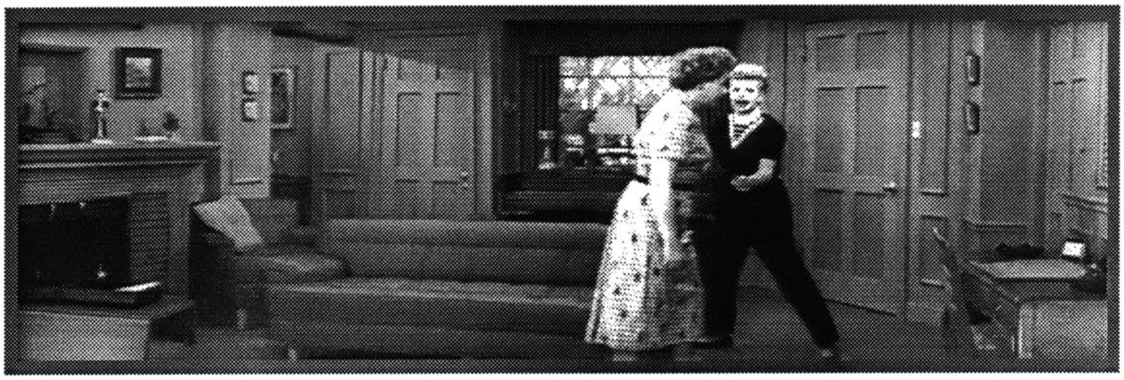

Figure 7-1: Reconstructed Widened Scene

7.4 Scene Modification

In addition to performing an accurate reconstruction of the original, several sequences were created which explored issues of picture modification. Three areas were approached: modifying the camera parameters; modifying the background set and changing the motions of the actors.

Two camera parameter variables were altered, one being the pan and the other the zoom. If the background set is large in comparison with the picture area, then there is the flexibility for altering the camera path or enlarging the picture. In particular, this alteration would be important at the edges of a pan across a widened set where the extents of the pan will need to be limited in order that the picture area is always filled. The addition of different zooms would be an artistic judgement. A sequence refilmed in this way is shown in figure 7-2.

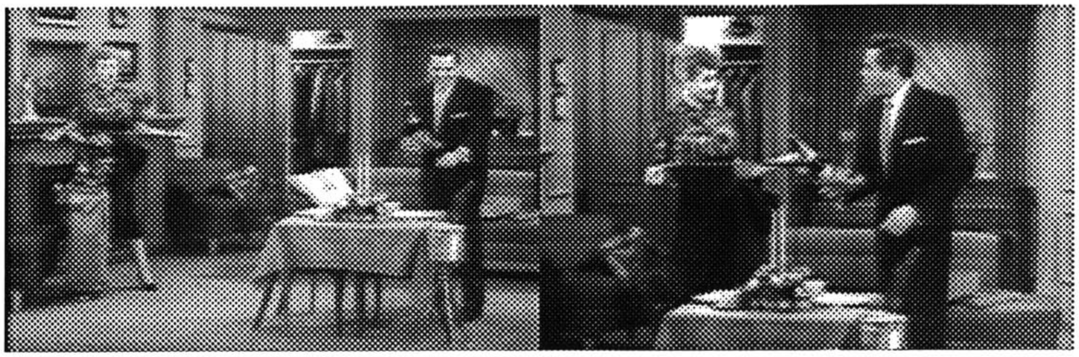

Figure 7-2: Sequence with zoom added

Changing detail in the background set might be desirable for a number of reasons. The issue explored here was in changing details. In one of the scenes a door at the back of the set is left open. This might have resulted from a mistake during the filming and if this was the case then being able to close the door without a reshoot would be desirable. All that is required is that the one shot of the background set needs to be retouched, rather than every frame of the sequence. A similar situation would occur if the film was to be colorized. A problem that arises from this is that any background surrounding the actors when they pass in front of the altered scene will become invalid. This can be solved in one of two ways, either segmenting the actors out accurately so that they carry no background with them, or placing an artificial halo around them which will blend more cleanly with the new background. An example of a sequence retouched in this way is shown in figure 7-3.

Changing the action of the actors is the hardest task in scene modification, and cannot currently be performed very gracefully. However, several ideas were considered concerning this problem. First the issue must be divided up into analysis and synthesis. The computer must be able to ascertain who is performing what, transmit that information and then synthesise a similar scene at the receiver. The analysis problem is extremely hard with current tools. All that could be ascertained were details such as how many people were in the scene and roughly where they were placed. This information was interpreted and an attempt made at ascertaining a single clean path. In some previous work [33], scenes were shot of actors performing various 'quanta' of action. For instance walking was divided into single paces and a continuous walk

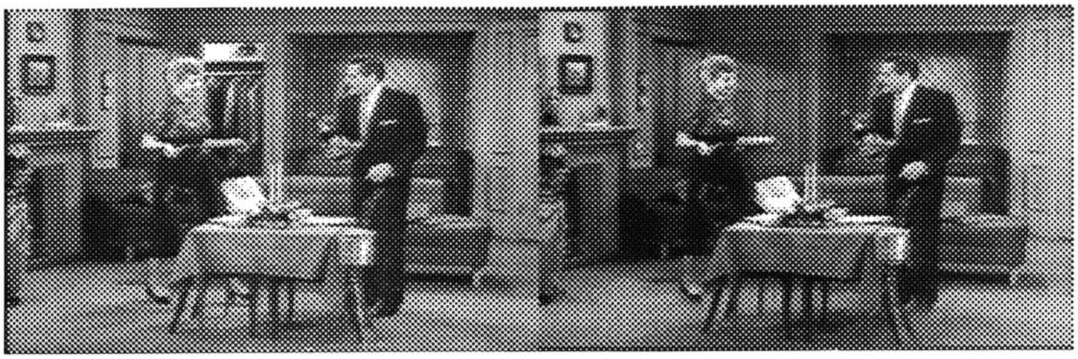

Figure 7-3: Modified sequence with closet door closed

made up by concatenating paces. A basis set of actions was built up for each individual which could subsequently be used to simulate larger actions. This approach was considered for the current work; however, the task of animating human action cannot be properly tackled in a 2-D model. The way forward is to create 3-D models of people which can then be conformed with the people in the original sequence, i.e. heights and proportions changed appropriately, face texture mapped on and clothes added. Such work was beyond the scope of this thesis.

Chapter 8

Results, Discussion and Conclusions

8.1 Results of the software work

An image coding model was designed and implemented in software which utilised the idea of creating a 2-D model of the background set to improve the efficiency of compression. This model was shown to be able to achieve bandwidths of down to 64Kbit/s with the material chosen. The resulting video showed considerable artifacts for which there is no easily quantifiable measure. Since the picture is being treated from a semantic point of view signal-based measures of quality are not appropriate. At very low bandwidths the most apparent artifacts were due to the coarse MPEG-style coding of the foreground. As the bandwidth was increased, the foreground image gained in quality and the artifacts at the boundary of the foreground and background images became more apparent. Tapes were created which demonstrated sequences coded at 64Kbit/s and 370Kbit/s. Other sequences were also animated which demonstrated picture modification, and in particular scene-widening (and some of its problems).

8.2 Discussion

This work raised many issues which are interesting to consider in order to gain an insight into the viability of applying methods such as those presented in this thesis to image coding. These will be discussed under the headings of low bandwidth video, scene-widening and object-based motion compensation.

8.2.1 Low Bandwidth Video

The coder was shown to be able to encode the chosen material down to 64Kbit/s. A considerable proportion of the bandwidth reduction was achieved by the MPEG coder and sacrifices in resolution. The gain resulting from the foreground/background separation in the instances chosen was to reduce the bandwidth to approximately $\frac{1}{5}$ of what it would otherwise have been. A judgement of the technique should address its robustness in dealing with a variety of source material and the nature of the artifacts it produces.

In order for bandwidth gains to be achieved, a reasonably substantial portion of the picture must be susceptible to modeling. This means that it must satisfy several requirements. One is that the background must be relatively static. A generalisation of this would be to make a distinction between material shot indoors and outdoors. In an indoor scene the model of a static set in which the actors move is usually relatively valid. In an outdoor situation this is likely to fail for several reasons, either the background contains moving objects such as cars, people or trees waving in the wind, or the camera might be in constant motion such as when it is tracking a person walking about a town. In the tracking case a great deal of new detail is being revealed each frame, rendering the gains of storing a background model insignificant. In such a case however, it might be possible to improve the system by creating a more dynamic model such as that discussed in the section on object-based motion compensation below. Another requirement is that the modeled part of the picture must make up a significant portion of the picture area. It gains us little to know how the background appears if a close-up of two heads talking take up 80% of the frame and we cannot model the heads. A third requirement relates to the temporal validity of the model. If a model of the set remains valid for a sequence of several seconds or more then the overhead of transmitting the model becomes insignificant. However, if the camera pivot is in rapid motion then the current model of the set will become rapidly redundant. Obviously, even if a set is valid for only two frames rather than one then bandwidth will have been reduced, but often the artifacts will not justify the gain. The break-even point is dependent on the application and on the nature of the complete movie. If the movie is mostly modelable then the coder can send high-bandwidth portions in advance during a low bandwidth section of the movie thereby smoothing fluctuations in bandwidth requirements. This technique makes considerable bandwidth fluctuation inevitable, which would disqualify it from some applications.

The artifacts of utilising this technique are interesting to consider. One set of artifacts arises from the compression of the foreground material. It is interesting to note that this technique increases the point at which MPEG-type artifacts become truly unbearable. The reason for this is because the artifacts do not take up the entire picture space. The actors look very dirty while the background is clean. The resulting picture is ugly, but considerably better than the sequence where the entire image had been coded using MPEG at the same foreground bits/pel. The latter is very hard to watch since the entire picture contains high temporal and spatial noise. Modeling the background gives the eye a rest from most of that noise. A second artifact relates to the pasting of the actors back into the modeled background. The multi-resolution blend performs a very good job of removing many of the most obvious problems associated with this. However, sometimes lighting differences between model and foreground are too considerable to be blended away and become apparent. In order to deal with this greater attention needs to be placed on the lighting issue. The beginnings of an approach towards this would be to attempt to identify which differences in luminance in the scene could be attributed to objects and which to shadows. If this could be done then shadow information could be simulated in the model. To transmit this shadow information would require more bandwidth, so it might be considered an 'optional extra', sent if bandwidth permitted to increase the quality of the picture. However, this work on shadows was beyond the scope of this thesis.

8.2.2 Scene Widening

The work presented here showed a viable technique for performing scene widening for such applications as HDTV. The investigation also revealed some of the less obvious problems that are associated with attempting this. In a similar manner to the discussion above the technique is dependent on the ability to model the scene, however it is more flexible in many ways. The pitfalls of scene widening can be considered as those being associated with the set and those associated with the people.

The first issue concerns rendering parts of the set not currently in view. We can think of several situations: the middle of a pan or zoom, the edges of a pan or zoom, or a still shot. The middle of the pan or zoom is relatively trivial situation; all that is required is a presentation of a wider view of the extended set. The edges of a pan can be dealt with by offsetting the center of view such that the extra area is not created equally on both sides but rather at the edge where data is known. The most difficult problem is at the widest zoom or the still shot. Here, no information is available of the surrounds of the picture. Two options are available, either a search for the required material from a different part of the movie, or in the last resort computer graphics could be used to render a guess at what would be at the edges of the set.

The second issue concerns modeling out of view people. Take the situation of the close up of two heads. It is possible that this could be shot in such a way that the backs of their heads would be out of view. If the heads can be modeled then the situation is solved by resorting to the model to fill in the missing portions. If the heads cannot be modeled, manual methods must be resorted to in order to fill in the missing pieces. A more complicated problem arises if one individual is known to be in the room but is temporarily off-camera. In such a situation we not only do not know what they look like but we also do not know what they are doing. That must be up to the creativity of the scene widener...

8.2.3 Object-based motion compensation

When a motion-compensated predictive coder is made to operate at low bandwidths the artifacts that appear are ugly and counter-intuitive. The edges of objects become ragged and this raggedness changes each frame creating temporal as well as spatial noise. The problem is that the objects which are being motion compensated are small blocks of the picture and often blocks from parts of the same object will have been judged to move in different directions. An approach towards a solution to this problem lies in the modeling work investigated in this thesis. If the motion-estimator and compensator could analyse the picture in terms of determining the motion of the objects within the picture rather than the blocks within the picture the resulting image would not degrade so badly at low bandwidths. The boundaries and the internal detail of each object would remain contiguous. The distortions which would arise would be in relative placement and in differences due to object motion which is not-modeled, in most cases all non-translational motion. An observation of such pictures indicates that such artifacts are more visually acceptable than artifacts arising from ordinary motion compensation at the same bandwidth.

8.3 Conclusions

A novel form of image coder was created which addressed the issue of using the structure of a video sequence to aid in the coding process. Many problems were successfully tackled, while others require further investigation and development. In particular improvements could be made in the segmentation of the actors, addressing lighting issues and in temporal noise reduction. Progress in these areas will lead to a more robust model and improved final picture quality. The system would be radically improved if a method could be found to successfully create models of the actors.

The techniques introduced in this thesis propose an interesting and productive approach to representing images. In its current condition, the scope of images sequences which can be modeled and encoded in this way is limited. Under the present paradigm for image coders this would render it to be of little use. However, there is no ideal 'universal' image coder and it is becoming clear that encoding techniques should be tuned and modified to suit the material which they are attempting to encode. The techniques presented here worked very well with 'I Love Lucy' but would be highly unsuitable for many other pieces of film. What has been created is an optimised 'Lucy encoder'.

Acknowledgements

I would like to thank a number of people for their help in creating this thesis:

Andy Lippman, my supervisor, for emotional outbursts at appropriate moments to indicate whether the work was headed in the right direction. Particularly I appreciated having a supervisor who *did* think it was going in the right direction.

Mike Bove for carefully reading the drafts and for constructive advice on many technical aspects of the project.

Henry Holtzman, for some of the segmentation software, but much more for being a font of all knowledge and comrade in arms.

John Watlington for always being at hand for some of the trickier questions.

Laura, for making great pasta, putting up with me and providing regular entertainment.

Mok, for being Mok.

And of course the garden crew, particularly Janet, Judith, Hakon, Foof, Abha and Barry; the only reason I really come in to work all hours.

In addition I would like to thank the Media Lab for being an enjoyable and exciting place to work. It has been a fun and an educational experience.

Bibliography

[1] Til Aach, Uwe Franke, and Rudolf Mester. Top-down image segmentation using object detection and contour relaxation. *ICASSP*, 1989.

[2] E. Adelson and J. Bergen. Spatiotemporal energy models for the perception of motion. *Optical Society of America*, 2, 1985.

[3] V. Michael Bove Jr. *Synthetic Movies Derived from Multi-Dimensional Image Sensors.* PhD thesis, Massachusetts Institute of Technology, 1989.

[4] Peter Burt and Edward Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics Vol 2 No 4*, 1983.

[5] William J. Butera. Multiscale coding of images. Master's thesis, Massachusetts Institute of Technology, 1988.

[6] Xiang Chen. Adaptive bit allocation for spatiotemporal subband coding using vector quantization. Master's thesis, Massachusetts Institute of Technology, 1990.

[7] C.L.Fennema and W.B.Thompson. Velocity estimation in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9, 1979.

[8] J. Cooley and J. Tukey. An algorithm for the machine computation of complex fourier series. *Mathematics of Computation*, 19, 1965.

[9] William Equitz. Fast algorithms for vector quantization picture coding. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2, 1987.

[10] Falk, Brill, and Stork. *Seeing the Light: Optics in Nature.* John Wiley and Sons, NY, 1986.

[11] F.Pereira, L.Contin, M.Quaglia, and P.Delicati. A CCITT compatible coding algorithm for digital recording of moving images. *Image Communication*, 1990.

[12] Bernt Girod. Motion-compensated prediction with fractional-pel accuracy. *Proceedings of the IEEE, Special Issue on Multidimensioanl Signal Processing*, 1990.

[13] Bernt Girod and David Kuo. Direct estimation of displacement histograms. *OSA meeting on Image Understanding and Machine Vision*, 1989.

[14] W. Glenn, Karen Glenn J. Marcinka, R Dhein, and I. Abrahams. Reduced bandwidth requirements for compatible transmission of high definition television. *38th Annual Broadcasting Engineering Conference, NAB*, 1984.

[15] Robert Gray. Vector quantization. *IEEE Acoustics, Speech and Signal Processing Magazine*, 1984.

[16] Motion Picture Experts Group. MPEG draft video standard. 1990.

[17] Robert Haralick and G. Kelly. Pattern recognition with measurement space and spatial clustering for multiple images. *Proc IEEE*, 57, 1969.

[18] Robert Haralick and Linda Shapiro. Survey of image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29, 1985.

[19] Eugene Hecht and Alfred Zajac. *Optics.* Addison-Wesley Publishing Company, Reading, MA, 1979.

[20] David Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1988.

[21] E. Hildreth. *The Measurement of Visual Motion.* MIT Press, 1984.

[22] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17, 1981.

[23] J. Jain and A. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, COM-29, 1981.

[24] Victor Klassen and Richard Bartels. Using b-splines for re-sizing images. Technical report, University of Waterloo, Ontario, 1986.

[25] Robert Kraft and Jeffrey Green. Distance perception as a function of photographic area of view. *Perception and Psychophysics*, 45, 1989.

[26] Edward Krause. *Motion Estimation for Frame-Rate Conversion.* PhD thesis, Massachusetts Institute of Technology, 1987.

[27] Doris Lessing. *Play with a tiger.* Vintage Books, NY, 1971.

[28] J. Limb and J. Murphy. Estimating the velocity of moving images in television signals. *Computer Graphics and Image Processing*, 4, 1975.

[29] Andrew Lippman. Semantic bandwidth compression. *PCS*, 1981.

[30] David Marr and Thomas Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B-204, 1979.

[31] David Marr and Shimon Ullman. Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London*, B-211, 1981.

[32] D. Martinez. *Model-Based Motion Interpolation and its Application to Restoration and Interpolation of Motion Pictures.* PhD thesis, Massachusetts Institute of Technology, 1986.

[33] Patrick McLean. Videosketch. Class paper, Digital Video. MIT Media Laboratory, 1990.

[34] T. Mizuno. Knowledge-based coding of facial images based on the analysis of feature points, motions and isodensity lines. *SPIE1199, Visual Communications and Image Processing IV*, 1989.

[35] Chee Kong Mok. Implicitly coded knowledge: Content-based representations of image sequences. Master's thesis, Massachusetts Institute of Technology, 1991.

[36] K. Nakayama and J. Loomis. Optical velocity patterns, velocity sensitive neurons, and space perception: a hypothesis. *Perception*, 3, 1974.

[37] Nasser Nasrabadi and Robert King. Image coding using vector quantization: A review. *IEEE Transactions on Communications*, 36, 1988.

[38] Arun Netravali and Barry Haskell. *Digital Pictures, Representation and Compression.* Plenum Press, New York, 1988.

[39] R. Ohlander, K. Price, and D. Raj Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8, 1978.

[40] Sinan Othman and S. Wilson. Image sequence coding at 64kbps using vector quantisation and block matching. *ICASSP*, 1989.

[41] Don Pearson. Towards a theory of model-based image coding. *IEE Colloquium on low bit rate video*, 1990.

[42] J.L Potter. Velocity as a cue to segmentation using motion information. *IEEE Trans., Systems, Man, Cybernetics SMC-5*, 1975.

[43] W. Reichardt and T. Poggio. Figure-ground discrimination by relative movement in the visual system of the fly. *Biological Cybernetics*, 35, 1980.

[44] Pasquale Romano Jr. Vector quantization for spatiotemporal sub-band coding. Master's thesis, Massachusetts Institute of Technology, 1989.

[45] Ronald Schafer and Lawrence Rabiner. A digital signal processing approach to interpolation. *Proceedings of the IEEE*, 61, 1973.

[46] William Schreiber. Psychophysics and the improvement of television image quality. *Society of Motion Picture and Television Engineers*, 1984.

[47] N. Seed and A. Houghton. Background updating for real-time image processing at tv rates. *Image Processing, Analysis, Measurement, and Quality*, 1988.

[48] A. Spoerri and S. Ullman. The early detection of motion boundaries. *IEEE First International Conference on Computer Vision*, 1987.

[49] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 1986.

[50] W. Thompson, K. Mutch, and V. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7, 1985.

[51] T.V.Robertson. Extraction and classification of objects in multispectral images. *Machine processing of remotely sensed data IEEE 73*, 1973.

[52] Shimon Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, Mass., 1979.

[53] H. Watanabi, H. Kuroda, and H. Hashimoto. A 64kbit/s video coding equipment. *IEEE Globecom*, 1987.

[54] John Watlington. Synthetic movies. Master's thesis, Massachusetts Institute of Technology, 1989.

[55] Steven Yelick. Anamorphic image processing. Bachelor's Thesis. Massachussets Institute of Technology, 1980.