Technical University of Crete

Department of Electronics

Digital Image and Signal Processing Laboratory

Panoramic Image Construction:

A break-down of the process

Dalamagkidis Efthymios

Graduation Thesis

Chania January 2004

Panoramic Image A Break-down of the process

Dalamagkidis Efthymios

Graduation Thesis

Abstract

The gradual development of video as mean of information transmission has created new needs for storing, transmitting and processing of its content. The lack of representation, manipulation and editing tools of video sequence does not allow the full exploitation of its content. This can be achieved by the construction and use of an image or a number of images that represent the static background of scene. These images are the panoramic views of the scene. They are also known as mosaics. In our work, we propose a method of constructing these images.

The basic steps of the construction of panoramic views are the frame alignment, which is based on the camera motion estimation, and the frame integration of the sequences. The estimation of motion parameters is performed by two different algorithms the Black Anandan Gradient Descent Algorithm and the Full Search block matching algorithm. Comparison of the efficiency of each algorithm prepared and their limitations are discussed. The integration can be achieved by frame overlapping or by using mean and median value filters. Experimental results are presented in order to indicate the efficiency of the above methods.

Supervisor: Prof. Michalis Zervakis
Prof. Euripides Petrakis
Prof. Nikos Sidiropoulos

Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision of Dr. Michalis Zervakis as also his unreserved support and belief in me.

I am also grateful to my parents for bringing me to this world.

Finally, I am forever indebted to E.T. for her understanding, endless patience and encouragement when it was most required.

Contents

List of Figures

List of Tables

1. Introduction

1.1. General

The term “panoramic image” denotes an image, which has a wide field of view. More specifically it is an image that represents a wide scene such as a landscape in only one photograph. The word panoramic derives form the Greek word panorama that is a composite word. The first composite is the word pan or in Greek “παν” which means everything and the second composite is the word “όραμα” which means vision, so from the definition of the word panoramic images are images that contain the whole field of view even though in practice this is not entirely true.

The idea of panoramic images is not a new one. One of the first panoramic cameras ever invented was the T.Sutton’s Panoramic Camera which used a spherical lens filled with water to accomplish the panorama created at 1858. Special cameras have evolved since then, which allow direct panoramic or omni directional capture.

Today there are two major trends in the field of panoramic vision. The first one arguing that panoramic images can be constructed by images taken by ordinary cameras by some software algorithm and the second one that claims that conventional cameras cannot meet the needs of Panoramic and in general Computer Vision. The last trend is established in the fact that conventional cameras were designed with the sole goal of capturing images to be properly shown in television or paper and are inefficient for the modern goals of Computer Vision. The hardware methods specially developed for panoramic imaging are of a fixed geometry meaning that the panoramic images captured have specific constant shape and dimensions. Currently there are hardware systems that either use conventional cameras enhanced with supporting mechanisms, such as mirrors and rotational devices, to achieve the panorama or they utilize special developed sensors. These sensors are also called omni-directional sensors as they provide multiple views of a scene always compared with the human vision.

Software methods or else mosaicing have the advantage of software implementation which can be more widely spread, have a great variety of applications, not only for constructing panoramic images but also for video handling, and are for sure cheaper and easier to customize than hardware method. They also have the advantage of dynamic geometry meaning that the final panoramic image can be represented in a more convenient way. The strive continues and only time will show the winner. In this thesis a brief summary is attempted on hardware methods but the main focus is placed on software techniques as the writer considers that other applications of mosaics are of great

importance to a lot of researchers. Such indicative applications are video indexing and retrieval and video compression.

1.2. Applications

Panoramic images have various applications and benefits apart from visualization and the aesthetics produced. It is proved that panoramic images can be of great use in surveillance applications as they provide a greater field of view and allow the observer to easily perceive changes, which is much more comfortable than the conventional viewing used today that presents in one monitor multiple views in turns. In the following sections some applications are presented to prove the need for continuous development of panoramic images.

1.2.1. Applications using Omni directional sensors

An indicative application showing the usefulness of omni directional sensors is localizing robots. The new trend in robot applications is to use more simplified robots than a highly sophisticated one as it reduces and simplifies the cost of manufacture; having a production line of robots is much cheaper than constructing only one original.

A communication scheme between the robots must be established but apart from that, the information coming from the sensors the robot is equipped with can be utilized to improve performance and robustness. The first task a robot is called to perform, is to localize itself within its environment and to the other robots operating in a common environment.

The utilization of omni-directional cameras broadens the field of view of the robot so that each robot is visible within the field of view of the other ones and the viewing angle can be easily computed because fixed geometry is used. This information can be shared between the robots via the communication scheme they utilize and an iterative algorithm can perform the localization.

The main problems of this application are:

- Obstacles may exist in the environment and so one robot may not be visible in another's field of view.
- Some of the robots may have identical angles.
- Some of the angles calculated may have significant errors due to the implementation of the system.

To overcome these problems the utilization of the whole information from the robots is suggested; it is highly unlikely that a robot cannot be seen by any other robot and, if that is the case, is excluded by the localization algorithm. Another suggestions is to built robust algorithms for the localization procedure which can cope with the extreme situations mentioned above.

The robot knowing its own position in the environment can simplify some tasks, like obstacle detection and avoidance and environment mapping.

1.2.2. Mosaic Applications

Mosaics, as software methods, tend to have various uses mostly irrelevant with the original purpose for which they were developed. A synopsis of the various utilizations of mosaic for video sequences is presented below. More detailed information can be found in [1, 2]

1.2.2.1. Storage and Reconstruction of video sequences

Mosaics are suitable for the storage of independent shots of scenes that the video sequence describes. In this application the video sequence is stored as panoramic images representing the background. A motion segmentation algorithm has preceded the mosaicing process and so individual moving objects and possible changes of the scene over the passage of time are also available. In the additional information of moving objects is overlaid on the static background obtained from the above process. In order to store all the information and to be able to reconstruct the original sequence later special data structures must be developed to hold the information the mosaic cannot handle. To fully reconstruct the original video the alignment parameters must be also stored and methods to will combine all the information of the current representation and to reconstruct the video sequence must be available.

1.2.2.2. Compression of video sequences

Compression of video sequences is a step higher than storage. With the intention of storage some compression schemes have been already developed, as a portion of the redundant information placed into the overlap between frames is not stored. Further compression can be achieved by compressing independently, using already existing techniques, the individual parts of the representation. So, for the composed backgrounds, which are actually large images, the JPEG compression may be used. For the alignment parameters and for the motion information of independent objects a data

compression algorithm like “zip” is possible to be utilized. This application is very important for the computer vision community therefore a flowchart of the procedure is shown in Figure 1

1.2.2.3. Efficient Transmission of video sequences

Transmitting a video sequence represented by a mosaic is much faster than transmitting the original video sequence, whereas considerably less information is transmitted. It is obvious that the transmission of a vast amount of independent frames is much slower than the transmission of a small number of mosaics. Although there is an overhead in the transmission, as additional information must be transmitted the amount of information to be transmitted is significantly reduced.

1.2.2.4. Synopsis Mosaics

These types of mosaics are a synopsis of the events taking place in the video sequence and it can be utilized in surveillance applications. Showing the independent moving objects above the static background makes the task of the observer significantly easier.

1.2.2.5. Manipulation and editing of video sequences

Video editing and manipulation are applications for which user input is required. Mosaics can be used here to provide the user with much more capabilities and ease the already existing ones. A common task in video editing is the insertion of an object in the scene. With the previous methods the user had to “copy paste” the same object plenty times in all the frames of the sequence but with the utilization of mosaics the object is be inserted only once in the scene. Mosaics also provide the user with a broadened field of view of the sequence thus the user develops a higher perception of the content of the video sequence. Finally, mosaics increase the efficiency and decrease the computational time of the video capturing and editing procedure.

1.2.2.6. Video indexing and retrieval

If the proposed storage method is utilized, the indexing and then retrieval of a specific portion of the video sequence becomes more efficient. A full representation of the video sequence, by using a scene detection algorithm, is realized by an automated procedure of creating a set of mosaics. In that case every mosaic represents a different scene of the video sequence and thus with a minimal user input, indexing can be achieved. Also, the information of moving objects is stored separately, so the retrieval of scenes containing moving objects or the retrieval of the objects themselves is much faster then the procedure of searching the video frame by frame.

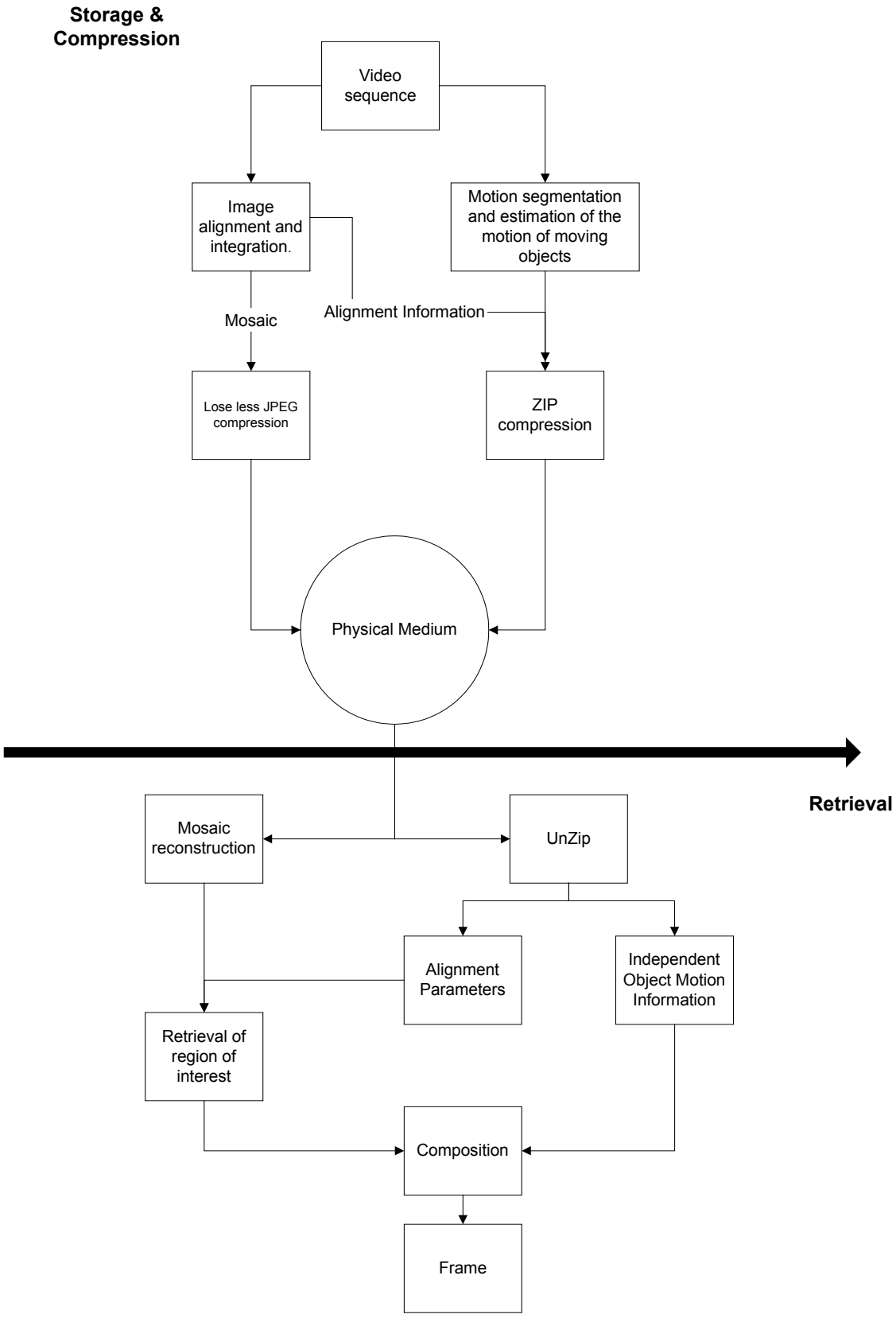


Figure 1: Mosaic storage, compression and retrieval

1.3. Thesis Structure

In Chapter 2 some fundamental concepts are presented related to the topic of computer vision. Firstly the homogenous coordinates, which are widely used to model the 3d world and the 2d plane of an image, are defined. Secondly the procedure of capturing a frame is described, as it is essential to understand how the objects in the 3d world are depicted in an image. The way motion of scene is described and its modeling is presented. Then the subject of mosaics is introduced as software methods used for the construction of panoramic images. An extensive report is presented regarding as well as all current mosaics categories and the processes of construction. The chapter concludes with a brief summary of hardware methods currently developed and used for panoramic imaging.

In Chapter 3 and 4 the procedure of constructing mosaics is discussed. First in chapter 3 the motion estimation problem is addressed, because it is essential for the mosaic process. The way the motion from two consecutive frames can be recovered is discussed and various ways of solving this problem as well. Then, in Chapter 4 the integration procedure of the overlapping regions between consecutive frames is considered in relation to what is the best solution according to the application. Also procedures to eliminate common errors occurring in mosaics such as the “ghosting effect” are reported.

In Chapter 5 the algorithm implemented for this thesis is described in detail and results are shown for the video sequences. Finally in Chapter 6 the ways to improve the current algorithm are discussed and the conclusions deriving from our work are summarized.

2. Fundamental Concepts

Some fundamental concepts of computer vision and the geometry that occupy computer vision must be explained in order to make this text more comprehensive. In summary we explain the procedure followed when a camera captures a snapshot of the 3d world and transforms it to a 2d surface the image as well as the ways we can model the camera motion from the frames of a video sequence. But first some geometric primitives must be discussed.

2.1. Homogenous Coordinates

Homogenous coordinates have been developed in projective geometry, so that they are also called projective coordinates, and have great use in computer vision as they "linearize" many problems. Homogenous coordinates of a point are derived from common Euclidean coordinates by adding a third coordinate with initial value one, so if a point in the Euclidean plane was represented by the pair (x,y) the corresponding triple of homogenous coordinates will be (x,y,1). In general a point in a n-dimensional Euclidean space can be represented by n+1 homogenous coordinates. The homogenous coordinates are equal up to a scale factor, meaning that the triple (x,y,1) is identical with the triple (λx,λy,λ) where λ is real and thus equal with the triple (U,V,S), this allows multiple representations of the same point. For notation reasons and to distinguish them from Euclidean coordinates the homogenous coordinates will be denoted with capital letters The homogenous coordinates of a point cannot be all three zero and thus the point (0,0,0) is not defined in projective geometry. The inverse procedure to go from homogenous coordinates to Euclidean is achieved by dividing the triple with the third coordinate S if S is non-zero, so for example the point (U,V,S) is the same with (U/S,V/S,1) if S is non-zero and the corresponding Euclidean coordinates will be (U/S,V/S). If S is zero the triple represents a point at infinity or an ideal point, which cannot be represented in Euclidean space. A point (U,V,S) in the Projective plane even though it contains three coordinates is a two dimensional space as the coordinates are unaffected by scalar multiplication. So the projective plane is the affine plane augmented by a single ideal line and a set of ideal points which are treated a common points and lines and there is no special treatment for them.

2.2. Camera Model and Video Capturing

First an explanation of the process of capturing the image must be clarified. The pinhole camera model will be presented below in order to unravel the procedure of capturing an image of the 3d world. Even though the pinhole model looks extremely simple it can adequately model the geometry of optics of may modern cameras like CCD. The pinhole model consists of a plane called the retinal or image plane, in Figure 2 called *R*, a point in 3d space, which is the camera or optical center, in Figure 2 called C, which denotes the point that light inserts the lens of the camera. The distance between the retinal plane and the optical center is called the focal length and the plane that is parallel to the image plane and passes through C focal plane denoted by *F*. Considering for the world 3d space the coordinate system (x,y,z) with center at C and Z axis the line which is perpendicular to the retinal plane and for the image the system (u,v) with center c, the standard coordinate system for the camera model is derived. Supposing a 3d point in the world with coordinates (x,y,z) then the corresponding point in the retinal plane will be the point m that intersects with the line CM and the retinal plane. The obvious problem is that any point lying on the half line CM is reflected to the same point m in the retinal plane, thus resulting in loss of information and more specifically the loss of depth information.

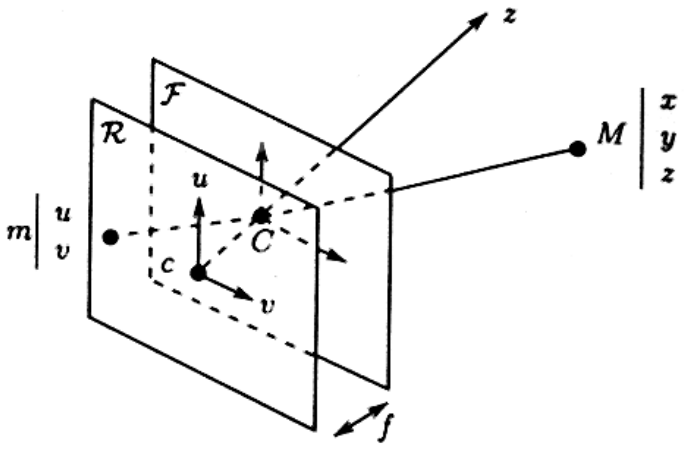


Figure 2: The pinhole model

It is obvious that the relationship of a 3d point of the world coordinate system and the corresponding 2d point in the retinal plane is denoted by the relationship:

$-\frac{f}{z}=\frac{u}{x}=\frac{v}{y}$ which if expressed in homogenous coordinates can be written in the form of :

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$ which is a linear relationship. Thus utilizing the homogenous coordinates the non-linear problem becomes a linear projection of 3d space to 2d space and as a consequence easier to compute.

The image plane chosen in the previous model is not always the image plane that coincides with the image captured by the camera, as it depends on the camera design. The most common difference is that the axes have different units, which is due to the electronics used in the capturing process. This introduces some parameters to the projective matrix P:

$$P = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

If the distance in the world system is expressed in units of length such as meters and in the image plane in pixel units then the k_u, k_v represent the horizontal and vertical resolution of the camera and u_0, v_0 denote the center of the image. These parameters are called the intrinsic parameters of the camera and give the capabilities of metric measurements with the camera such as angle.

Following the same pattern, as above the 3d world coordinate system may differ from the standard coordinate system for the camera because six more parameters are introduced, which map the world coordinate system to the standard coordinate system. As a result the projective matrix becomes:

$$P = \begin{bmatrix} -fk_u r_1 + u_0 r_3 & -fk_u t_x + u_0 t_z \\ -fk_v r_2 + v_0 r_3 & -fk_v t_y + v_0 t_z \\ r_3 & t_z \end{bmatrix}$$

where r_1, r_2, r_3 are the row vectors of the rotation matrix used to rotate the world coordinate system to the standard coordinate system and t_x, t_y, t_z are the corresponding translational parameters.

The transformation of coordinates of a 3d point in the 2d retinal plane is in fact a linear projective transformation that can be expressed as a matrix multiplication:

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = P * \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$ where P is the projective matrix. For most of the applications an ideal camera with a normalized focal length and no distortion by the lens is sufficient. With this admittance the transformation becomes:

$$u = \frac{X}{Z}, v = \frac{Y}{Z}$$

Further information on these systems as also about how to estimate the intrinsic and extrinsic parameters of the camera can be found in [3]

2.3. Motion models

In general the camera can make 3 types of translational and 3 types of rotational movements each one in each axis (X,Y,Z) of the 3d world. The translation along the X,Y axis is also called "Track and Boom" the rotation around the X,Y without significant change in depth Z are called "Pan" and "Tilt" respectively, "Roll" is called the rotation around the Z axis and "Zoom" the translation in the Z axis or the change of focal length. So the camera has 6 degrees of freedom in movement. And the equations describing the motion are:

$$\vec{V} = \begin{bmatrix} \frac{dX}{dt} \\ \frac{dY}{dt} \\ \frac{dZ}{dt} \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} \Omega_Y Z - \Omega_z Y \\ \Omega_z X - \Omega_X Z \\ \Omega_X Y - \Omega_Y X \end{bmatrix}$$

Where T_X, T_Y, T_Z is the translation motion and $\Omega_X, \Omega_Y, \Omega_Z$ the rotational motion. The full recovery of the 3d motion of the camera from video sequences is an ill posed problem as every attempt to estimate a richer representation from a poorer one. Several methods have been developed for the 2d motion estimation by making specific assumptions and implying constraints to the motion behavior.

From the equations above the mapping from 3d motion to 2d motion can be extracted:

$$\frac{dx}{dt} = \frac{1}{Z}\frac{dX}{dt} - \frac{X}{Z^2}\frac{dZ}{dt} \text{ and } \frac{dy}{dt} = \frac{1}{Z}\frac{dY}{dt} - \frac{Y}{Z^2}\frac{dZ}{dt}$$

Supposing that the surface the camera captures can be approximated by a plane with this equation:

$$n_X X + n_Y Y + n_Z Z = 1$$

then the 2d motion vectors are:

$$\frac{dx}{dt} = (n_Z T_X + \Omega_Y) + (n_X T_X - n_Z T_Z)x + (n_Y T_X - \Omega_Z)y + (-n_X T_Z + \Omega_Y)x^2 - (n_Y T_Z + \Omega_X)xy$$

$$\frac{dy}{dt} = (n_Z T_Y - \Omega_X) + (n_X T_Y + \Omega_Z)x + (n_Y T_Y - n_Z T_Z)y - (n_Y T_Z + \Omega_X)y^2 + (-n_X T_Z + \Omega_Y)xy$$

This equation is a complex non-liner notion model with eight degrees of freedom. Substituting the coefficients of the above equations with real parameters parametric motion models arise. The assumptions made for the behavior of the motion the model has to describe set the parameters; parameters, which express a different kind of motion than the one the model is able to handle, are set to zero; so the number of the parameters and thus the complexity of the model depends mainly on the motion the model is called to describe. In Table 1 some commonly used parametric models are shown.

Model	**Transformation**	**Parameters**
Translation	$x' = x + b$	$b \in \mathbb{R}^2$
Affine	$x' = Ax + b$	$A \in \mathbb{R}^{2x2}, b \in \mathbb{R}^2$
Bilinear	$x' = q_{x'xy}xy + q_{x'x}x + q_{x'y}y + q_{x'}$ $y' = q_{y'xy}xy + q_{y'x}x + q_{y'y}y + q_{y'}$	$q_* \in \mathbb{R}$
Projective	$x' = \frac{Ax + b}{c^T x + 1}$	$A \in \mathbb{R}^{2x2}, b, c \in \mathbb{R}^2$
Pseudo-perspective	$x' = q_{x'x}x + q_{x'y}y + q_{x'} + q_a x^2 + q_b xy$ $y' = q_{y'x}x + q_{y'y}y + q_{y'} + q_a xy + q_b y^2$	$q_* \in \mathbb{R}$
Biquadratic	$x' = q_{x'x^2}x^2 + q_{x'xy}xy + q_{x'y^2}y^2 + q_{x'x}x + q_{x'y}y + q_{x'}$ $y' = q_{y'x^2}x^2 + q_{y'xy}xy + q_{y'y^2}y^2 + q_{y'x}x + q_{y'y}y + q_{y'}$	$q_* \in \mathbb{R}$

Table 1: Parametric Motion Models

The Translation model is the simplest model and it is able to describe only translational motion which is the most common camera motion encountered. This motion is expressed as a shift of the image coordinates by a scale factor. Although this model is very simple and has no constraints in its use it is improper for more complex motion including rotation and zooming.

Affine model is the most commonly used model for 2d motion estimation and has six parameters each one describing a different kind of motion. The motions that the Affine model is able to describe are translational motion, camera rotation around the optical axis, zooming and small curvature in the horizontal or vertical direction, introduced as shear factor.

The most appropriate and efficient model for the description of all camera motions is the Projective. Projective model has eight parameters and as it is an extension of the Affine: It can describe all the motion that the Affine is able to, plus the rotation along X,Y axis. The only constraint the Projective model imposes is that all the objects of the scenes, including the background, must be in the same depth.

The most complicate model is the Biquadratic and is capable of describing all the possible camera motions. The problem of this model is the twelve parameters, which make it prone to errors. Its use is confined only if multiple and complex motions are present in a sequence, as it's the only one that can fully describe it.

The Bilinear model is an eight-parameter model and it is a simplified biquadratic as it can be extracted from it by zeroing the x^2 and y^2 factors. Bilinear is not capable of modeling rotational movements along the axis.

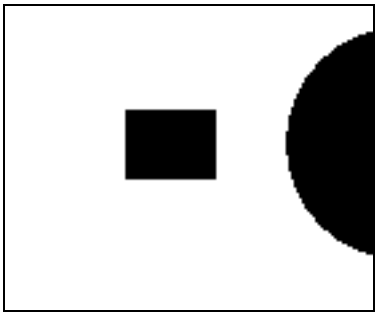
In order to estimate the parameters motion vectors of some specific number of pixels must be calculated. The minimum number of points need for its model is $\frac{\text{no of parameters}}{2}$. So, for example, in the Affine model three points will suffice theoretically for the calculation of the parameters. In practice, for a good estimation of the parameters, a larger number of points is needed, as the most 2d motion estimation algorithms have some wrong estimates occurring (either by noise added during acquisition or by independently moving objects in the video sequence).

2.4. Mosaics

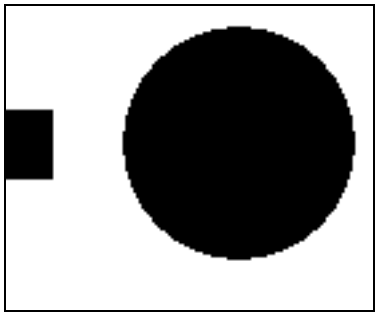
2.4.1. General

In the late 80s as a result of the spread of television, video became a part of ordinary people's life. More devices became slowly widely available providing the possibility to record and capture video sequences such as camcorders and ordinary video players and recorders. A major breakthrough came along with the digitization of video sequences, which opened new horizons concerning their exploitation. A video sequence is, in fact, a set of continuous images captured at a high sample rate, in usual video standards such as PAL is 25 frames (images) per second or in NTSC 30 frames per sec, and thus can hold a vast amount of information about time changing scenes. The amount of information a video sequence holds demands, though, a vast amount of storage space; for example a typical video in PAL format of 10 seconds will hold about 250 Mbytes. This cost of storage space and the complexity of handling this amount of information led to the development of new techniques for video compression and storage, and for indexing and retrieval too. The main concept is that video sequences hold much of redundant information, because usually there is a lot of overlap between two consecutive frames.

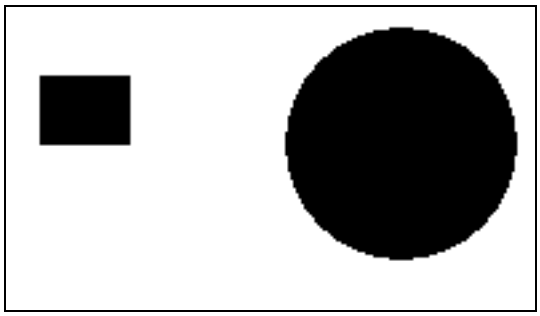
One technology developed for these needs is "mosaicing". The basic idea is to create an efficient representation of a continuous video sequence as one image holding only the non-redundant information. An example can be found in Figure 3. In this example, if the black circle and rectangle are considered to be background, all the motion is due to the "camera motion" moving right. The mosaic image is a synthesized image of the former images and there is a 30% reduction of the information.



Frame1



Frame2



Mosaic Image

Figure 3: Figure showing two consecutive frames and the image mosaic derived from them.

The example above is relatively simple and with real world frames the process of building and representing mosaics becomes even more complex, as multiple motions, moving objects and moving camera may be present. Depending on the application, and as a consequence of the information a mosaic "needs" to hold, there is a variety of types of mosaics and techniques of construction. Nevertheless in all cases the mosaicing process can be divided in two sub processes. The first is the image alignment or motion estimation and the other one is the image integration.

The first sub-process poses the problem of the calculation of the relative motion between two consecutive frames and is not a new issue in video processing research area. Motion estimation is a well-known ill-posed problem as it tries to recover 3d motion from 2d images, which means going from a poorer representation to a richer.

The second sub-process is called image integration or image blending. This is the process of stitching the images seamlessly together and it is interlocked with the image alignment process. If the motion estimates are poor, then the integration will be bad and the final mosaic will have severe

errors. This sub process chooses the pixels that will construct the final mosaic. Carefulness must be shown in the area that frames overlap; depending on the technique used, problems may occur, especially if motion exists from moving objects, apart from the camera, which tend to visualize a "ghosting" effect.

The simplest type of mosaic is the one that does not hold any kind of temporal information and there is only one motion in the video sequence the one of the camera. This kind of mosaic is called "static mosaic" and from this one it is possible to construct panoramic images. Such techniques will be examined more thoroughly in the following chapters.

In the following section there is an attempt to extensively report all currently available mosaics and to organize them in categories according to specific criteria for the convenience of the reader. These criteria are relate to the process of construction, the information that the mosaic must hold and the way of representation. They are summarized in Figure 4

Figure 4: Mosaic Categories

2.4.2. Categories

Mosaic construction is achieved by utilizing all the frames of a video sequence and results in one image. According to the needs of the application several types of mosaic occur depending on specific criteria. Such criteria are the ability of the mosaic not only to hold static information but also temporal information, the alignment method used to register the methods and also the manifold of representations. Manifold is a geometrical surface where the frames of the video sequence are mapped; the most common manifolds are cylindrical, spherical and plane. The following criteria will be examined thoroughly below. Keeping in mind that there is no best method or type of mosaicing and that each one has its own advantages and disadvantages, one must take into consideration the tradeoffs to choose the right one for the corresponding application.

2.4.2.1. Temporal Content Criterion

This criterion examines the mosaics from the perspective of holding temporal information. As it is known, video sequences are holding information of continuous changing scenes. The relative motion between two consecutive frames can occur either from a camera movement (egomotion) or by moving objects in the scene. A characteristic example is the one of a sequence of a football match. The camera moves in order to cover all the terrain and also catches the movement of the players and possibly the ball and crowd. So, on one hand in this example, the term temporal information or temporal content represents the movement of the players and the ball; on the other hand the static information or background is the terrain and possibly the crowd because their movements are relatively small compared to the movements of the players.

The two most common mosaics are static mosaics, which do not hold any temporal information in the mosaic; on the contrary the information is possible to be held separately. And the dynamic mosaics, which are in fact video sequences which have mosaics as frames, instead of images. Both these types of mosaics as proposed by Irani in [2], are the extremes of another representation the one of the temporal pyramid which is capable of holding and disregarding temporal information depending on the form of the representation. For simplicity and traditional (yes there is a tradition even in Computer Vision) reasons and because these categories are the most popular they will be examined separately.

2.4.2.1.1. Static Mosaics

Static mosaics are a representation of mosaics that cannot hold any temporal information. This representation holds information which represents the background of the video sequence and thus a panoramic image. On one hand their inability to hold temporal information makes them very popular for creating mosaics where only egomotion exists and there are no moving objects as they are pretty simple. On the other hand, if moving objects are present and the background possesses the major part each frame (admission of dominant motion assumption), the mosaic can still be constructed. But other problems will arise as the redundant information is also integrated in the mosaic producing erroneous results. Such problems are the possible misalignment of some frames because of the presence of other motions, as well as the most classical error in composing mosaics the ghosting phenomenon. If the background is not the dominant motion between the two frames, the method will probably fail. The ghosting phenomenon appears when in the final mosaic the moving objects leave a trace on the corresponding locations and this is very obvious in Figure 5. Observe that in the final mosaic a ghost like figure of the player is visible. This type of error can be compensated with special "deghosting" methods discussed later or by choosing a better integration technique. Another way to correct this error is to perform motion segmentation before the construction of the mosaic and construct the mosaic with information concerning only for the background as proposed in [4]. Then the temporal information can be held separately and so the result is a complete representation of the video sequence without any loss of information. Unfortunately motion segmentation algorithms are not so stable and trustworthy and they can lead to problems of the representation. Also in the mosaic constructed by the background information gaps may occur because there is no information for these areas as they were occluded by the moving objects.

The information about moving objects -or else residual- can also be estimated approximately by comparing each frame and the static mosaic. Holding his information, either separately or in another layer, can make the static mosaic a very efficient representation of a video sequence. Unfortunately, most researches have sufficed to building the static mosaic and disregarded the residual information so there is no stable or efficient way to track the residuals.

Static mosaic is ideal for video indexing and storage, as it can very easily retrieve the original frame at any time, supposing that the residual information exists in some form. Also it's ideal for video compression, as it disregards the needless overlapping information between frames. For

efficient video compression a scene detection algorithm must be applied because mosaics have meaning for video sequences, which represent scenes with backgrounds not changing in time.

Static mosaics, being the most common go also under other names such as "mosaic" and "salient still".

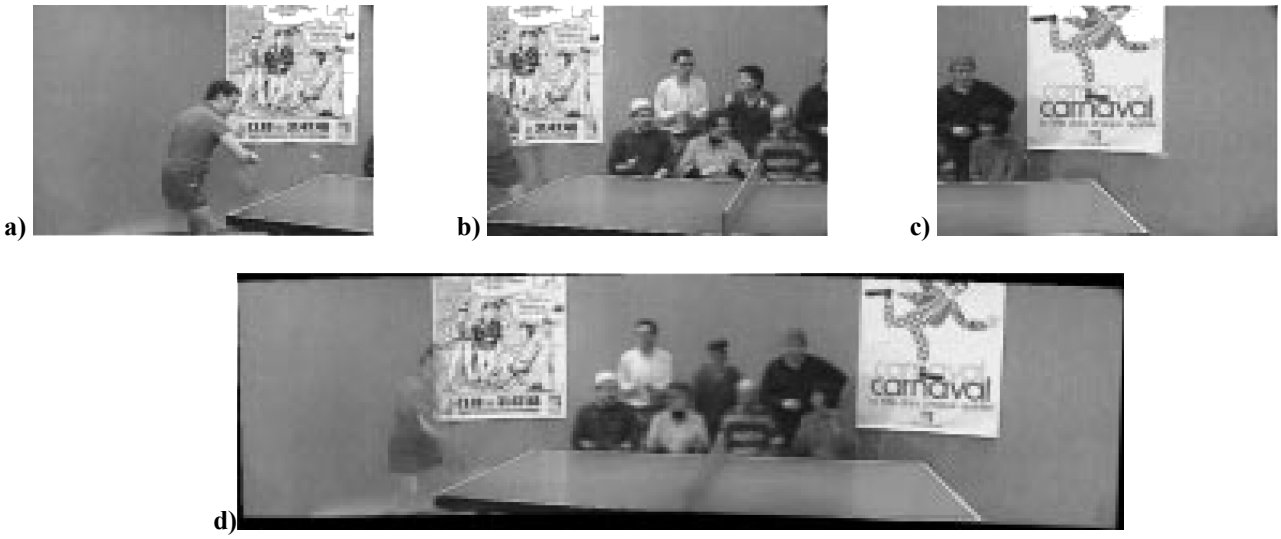


Figure 5: Static Mosaic image of table-tennis game sequence.

a),b),c) Three indicants frames from the sequence d) The static mosaic Images are extracted by [2]

2.4.2.1.2. Dynamic Mosaics

Dynamic Mosaics came out of the need of mosaics to represent except for static information temporal information, too. In fact the representation consists of a sequence of mosaics, each "mosaic frame" being composed by the previous mosaic and the most recent frame. So the temporal information of the video sequence is stored in the corresponding "mosaic frame". An example of a dynamic mosaic is shown in Figure 6. which make obvious the process of construction. For the dynamic mosaic construction, either the frame to mosaic alignment is used or the mosaic to frame alignment is used, which will be explained below. It must be clarified that in dynamic mosaics moving objects are clearly shown and there is no ghosting phenomenon. Furthermore dynamic mosaics may suffer from wrong representations of scenes because the most recent information exists only in the corresponding region of the last frame; therefore possible simultaneous existing movement in other parts of the scene is lost and this can lead to confusing representations. However this problem can be solved only with hardware methods, as the input video sequence does not hold the extra information.

The dynamic mosaics have significant less residual compared to the static mosaics and that is expected to happen as they are designed to hold temporal information. The possible residuals exist due to the changes in the scene over time, in areas that the camera does not turn to and on objects first obscured by other objects and later revealed.

In conclusion, dynamic mosaics are better than static mosaics as far as the representation of scenes including high temporal information is concerned, as static mosaics cannot represent this kind of information. On the other hand the high storage capacity required and the lack of direct access to the original frames make it unsuitable for applications like video indexing and retrieval.

More information for dynamic mosaics can be found in [5, 6] but these methods are semi-automatic and human intervention is required.

Figure 6: Dynamic Mosaic of table-tennis game sequence.

Left column: Three frames from the original sequence.
Right column: The corresponding dynamic mosaic images.
Images are extracted by [2]

2.4.2.1.3. Temporal Pyramid

The static and dynamic approaches are extremes of another theoretical mosaic, called "the temporal pyramid". Temporal pyramid is usually a Laplacian pyramid with factor two, although it can be scaled up to any factor. Supposing the pyramid's factor as F and the number of frames as N: then the pyramid will have $[\text{int}(\log_F N+1,5)]$ levels. The finest level of the pyramid contains in fact the original frames. Each coarser level consists of a mosaic composed by F images or mosaics of the finest level. A figure of the pyramid is shown in Figure 7. The final or coarsest level will be a mosaic very similar to the static mosaic. The intermediate levels will be a sequence of mosaics made of a specific subset of frames of the original video sequences, which are very similar to dynamic mosaics.

Such a representation provides easy video indexing and retrieval, as the original frames are stored as independent information and it also provides with a panoramic view of the scene. On the contrary the high computational requirement along with the storage capacity requirements make the integration of such a system prohibiting.

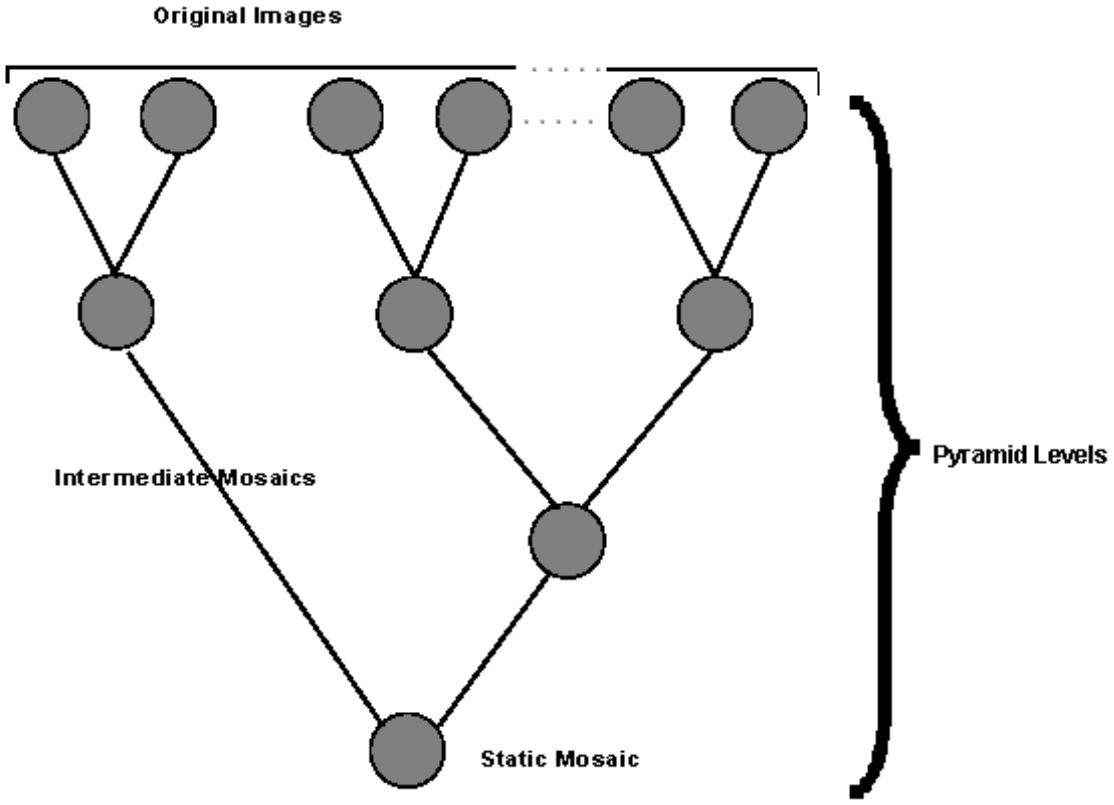


Figure 7: Graphical Representation of the Temporal Pyramid with factor 2

2.4.2.2. Alignment method

Another category of mosaics derives from the alignment method chosen; meaning what is aligned to what and so a reference object is selected. All the methods below can be also used in the process of making a mosaic with a fixed coordinate system. The procedure consists in mapping the reference object to the fixed coordinate system so as to have the transformation matrix of the coordinates. Then it is possible to map all the following objects to the coordinate system of the reference object using the transformation matrix obtained by the procedure above.

2.4.2.2.1. Frame to Frame

The first approach used to construct a mosaic as far as the alignment of the frames of a video sequence is concerned was the frame-to-frame alignment method. In this method, firstly, the dominant motion between two consecutive frames of the video sequence is computed. The motion parameters deriving from the motion estimation process are then used to align the two frames. The whole process is repeated for all the frames of the video sequence.

The main drawback of this method is that small errors occurring in each alignment stage are cumulative and so errors may be visible in the final mosaic and they will be more obvious between the frames in the beginning and in the end of the sequence.

2.4.2.2.2. Frame to Mosaic

To compensate the cumulative error in the frame-to-frame alignment process, it is possible to register each image frame of the video sequence to the current mosaic image, utilizing a block matching algorithm or a motion model which searches for motion over a specific region. The current mosaic is composed by all the frames previous to the frame which is to be registered. To handle the problem of large displacements between the current mosaic and the frames to be registered the initial parameters of the search are set to be the results of the previous registration.

2.4.2.2.3. Mosaic to Frame

In some applications holding temporal content like real time video-transmission it is important to keep the frames in their original coordinates. For these cases the current mosaic is aligned with the most recent frame. The Mosaic-to-Frame method is usually used for dynamic mosaics.

2.4.2.3. Motion Model Criterion

Motion models are used in the process of image alignment as a tool to estimate the motion between two frames. This criterion will divide mosaic in categories based on scene complexity.

2.4.2.3.1. 2D Mosaic

This is the most common case in constructing image mosaics. Of course the motion projected to the video sequence must undergo some specific constraints. In general, only one dominant motion either the motion of the camera (egomotion) or the motion of the objects must be present and the entire scene can be approximated by a single parametric surface, usually it is a plane. This means that the entire scene must have the same depth. Motion models commonly used in this case are the Affine model, if there is no rotation, and the Projective or the Pseudo-perspective to cover the possibly existing rotations. In practice some of the constraints imposed above can be violated under certain circumstances for example when there are small changes in the depth of the sequence relative to the global depth of the sequence. This parallax can be neglected, as it will be represented in only a small number of pixels in each frame.

In order to have a good estimation of the parameters the framework proposed by Bergen [7] as well as a minimization technique can be used. Usually the minimization criterion is given by the sum of squared differences measure applied over specific regions of interest. The minimization criterion is expressed by:

$$E(\{u\}) = \sum_{x} (I(x,t) - I(x - u(x), t-1))^2$$

where x=(x,y) denotes the image coordinates, I the image intensity and u(x)={u(x,y),v(x,y)} the corresponding motion vector at that point. The sum is computed over all the pixels of the region and {u} is the entire motion field of the region.

The function can be minimized then by an optimisation technique like the Gauss-Newton method.

2.4.2.3.2. Parallax Based

The 2d mosaic is sufficient only when there is small parallax related to the surface. In order to overcome this problem the alignment process is extended into three dimensions. The computation of the 3d depth parallax information can be done in two ways.

One method uses a sequential registration approach to solve the problem; first the dominant plane is registered using a 2d model and then the residual effects are computed. After the plane is aligned this way, a quasi-parametric motion model is used to calculate the motion. The main drawback of this method is the assumption of a visible planar surface, which occupies the dominant area of the scene.

The other method estimates simultaneously the two components and uses a virtual reference plane to compute. In this case the motion vector in each pixel of the scene is represented by the sum of the translational motion, for which an initial estimation can be given by applying a 2d alignment technique and the residual motion which is initialized to zero. The motion is incrementally refined by the processes described above.

More information can be found in [2, 7].

2.4.2.4. Manifold Criterion

In order to construct a mosaic a common manifold is required to project the individual frames of the video sequence. Manifold is a closed geometrical surface and it is polymorphic. Such common manifolds are the cylindrical and the spherical manifold. In the following sections they will be examined in greater detail. Other manifolds have been proposed but exceed the scope of this thesis indicative references are: [8, 9, 10, 11]

2.4.2.4.1. Cylindrical Mosaics

Cylindrical Mosaics are the first mosaics ever created but are of limited capabilities. The cylindrical manifold is in fact a cylinder, which allows a 360 degrees field of view in a limited height. In order to construct cylindrical mosaics constraints on the motion of the video sequence must be imposed. First, the camera motion must have only horizontal translational components and this can be assured by mounting the camera on a tripod and rotating it around its optical axis. This procedure confines that type of mosaic since special preparation of the video sequence must be done and thus cannot be used in every video sequence. In practice compensation must be made for vertical components of motion but that is an easy process to realize as the tripod assures them to be small and easily calculated. As any 2d alignment method, no depth parallax should exist in the video sequence.

The equations that transform a 3d point with world coordinates (X,Y,Z) to cylindrical coordinates are:

$$\theta = \tan^{-1}(\frac{X}{Z})$$

$$u = \frac{Y}{\sqrt{X^2 + Z^2}}$$

The transformation of the world coordinates to image coordinates depend on the camera intrinsic and extrinsic parameters and it is a linear projective transformation

Another problem of cylindrical coordinates is that become undefined as they approach the north and south pole thus resulting in registration errors. Also, the depth of the scene must be known which can only be approximated.

A system for building cylindrical mosaics has been constructed by Apple under the name of VideoVR [12].

.

Figure 8: Cylindrical Mosaic Captured by the Video VR system [12]

2.4.2.4.2. Spherical Mosaics

Spherical mosaics are very similar to cylindrical mosaics and they also suffer from the same limited capabilities. They are built in a similar way to the cylindrical mosaics, having a proper moving of the camera in order to capture a spherical field of view. To build a Spherical mosaic the world coordinates p=(X,Y,Z) are mapped to 2D cylindrical coordinates (θ,φ) using the following formulas:

$$\theta = \tan^{-1}(\frac{X}{Z})$$

$$\phi = \tan^{-1}(\frac{Y}{\sqrt{X^2 + Z^2}})$$

2.4.2.4.3. Plane Mosaics

Plane Mosaics are the most commonly used. A plane is set as a reference plane usually the plane of the first image; then all images are warped to that plane and then they are registered to each other. In order to achieve a successful warping a good motion estimation between the frames must be computed. Plane mosaics are simple, straightforward and can represent almost the whole information concealed in the video sequence thus their preference of use. Their disadvantage is that because there is no constraint in the movement the final mosaic is of "peculiar" dimensions. A plane mosaic is shown in Figure 9.

Figure 9: Plane mosaic

Image Extracted by [13]

2.5. Hardware Methods

On the last years a lot of hardware systems have been developed to directly capture and construct panoramic images. These systems can be divided in those that use a special kind of cameras and lenses, such as a fish-eye lens, to directly capture the panoramic images and those that use conventional cameras and some other kind of mechanism to broaden the field of view. These systems have been successfully used in the industry for inspection, surveillance and robot navigation. An extensive report of all these systems can be found in [1]. A brief summary of the currently available systems will be presented below and their advantages and disadvantages will be discussed.

2.5.1. Fish Eye Lens

The term "fish eye" lens was first introduced by Robert W. Wood in his book "Physical Optics" in 1911. Wood had the inspiration of the term by observing the refraction of rays entering the surface of a pond and extending to the description of a water-filled pinhole camera capable of simulating a "fish eye" view of the world. In general the term "fish eye lens" is used to describe

cameras that can capture images of 180-degree field of view. There are two major types of "fish eye lens", the Full frame and the Circular Image. The former captures a hemispherical image while the latter captures an image of 180-degree field of view with a narrow height. The main drawback of this lens is the extreme distortion occurring in the edges of the captured image, which is a consequence of the wide field of view. Fish eye lens is a relatively old technology and has been used by many manufactures of cameras around the world.

2.5.2. Systems using special cameras

These systems usually utilize some kind of mirror to broaden their field of view by composing the final image from the reflections on the mirror. The mirror is placed in front of a camera and a special apparatus supports it. Both the mirror and the apparatus are very important in designing the system and they both have their own advantages and disadvantages making it hard to decide the best one for each application. These sensors are usually called omni-directional vision sensors and the images acquired omni-directional images (it is in fact a sub-category of panoramic images). The omni-directional sensors have the advantage of being able to directly capture in real-time an image with wide field of view but on the other hand are pretty expensive to manufacture and the images taken present a relatively low resolution.

Except for the mirror which will be examined separately below, the apparatus that holds the mirror is of great importance. This apparatus must be made this way so to eliminate any internal reflections and it must have a smooth surface for acquiring non-distorted images. The most common apparatus used are a clear cylinder or sphere made of plastic or glass.

There are four type of mirrors used for the construction of these sensors:

1. Spherical
2. Conic
3. Hyperboloid
4. Parabolic

The advantages and disadvantages of each type of mirror will be presented and then conclusions are going to be made about the efficiency of each mirror.

2.5.2.1. Spherical

This kind of mirror is suitable for observing objects, which are in the same height with the sensor.

Advantages:

- Low manufacturing cost
- Small astigmatism
- No need of long focal length for acquiring a focused image

Disadvantages:

- Does not have one center of projection
- Images cannot be transformed into normal perspective images
- Viewing angle to large
- Image distortion in the peripheral of the image

2.5.2.2. Conic

Suitable for acquiring images in which vertical visual field is limited

Advantages:

- Easy to manufacture
- Can combine several mirrors
- Can observe horizontally with the use of telecentric lens

Disadvantages:

- High cost if telecentric lens used
- Large astigmatism
- Need of long focal length for acquiring a focused image
- Images cannot be transformed into normal perspective images
- Does not have one center of projection

2.5.2.3. Hyperboloid

Suitable for monitoring applications

Advantages:

- Has one center of projection
- Images can be transformed into normal perspective images
- If the curvature is small astigmatism is small

Disadvantages:

- Hard to manufacture and design
- High cost

2.5.2.4. Parabolic

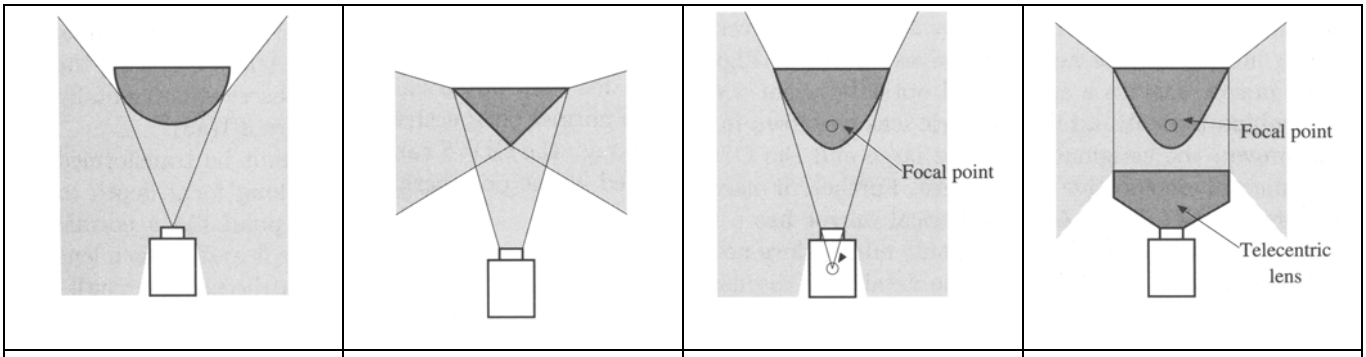
Ideal system to acquire panoramic images

Advantages:

- Has one center of projection
- Images can be transformed into normal perspective images
- If the curvature is small astigmatism is small
- Use of telecentric lens

Disadvantages:

- Hard to manufacture
- High cost
- The telecentric lens increases the size of the system



Spherical Mirror	Conic Mirror	Hyperboloid Mirror	Parabolic Mirror

Figure 10: Field of view of each mirror

Image extracted from [1]

	Mfg. Cost	Astigmatism	Focal Depth	Vert. Viewing angle	Single center of projection	Lens
S	Low	Small	Short	-90 … 10	No	Normal
C	Low	Large	Long	-45 … 45	No	Normal
HS	High	Small	Short	-90 … 10	Yes	Normal
HL	High	Large	Long	-90 … 45	Yes	Normal
PS	High	Small	Short	-90 … 10	Yes	Telecentric
PL	High	Large	Short	-90 … 45	Yes	Telecentric

S: Spherical Mirror
C: Conic Mirror
HS: Hyperboloid with small curvature
HL: Hyperboloid with large curvature
PS: Parabolic with small curvature
PL: Parabolic with large curvature

Table 2: Summary Table of Mirror characteristics

Table extracted from [1]

2.5.3. Systems using enhanced conventional cameras

The construction of special panoramic sensors is in most cases difficult and expensive so approaches to use conventional cameras and some kind of other external mechanism in order to enhance the conventional camera and broaden its field of view have been attempted. Such systems are using either an external mirror or mirrors in some distance from the camera or a rotating device which allows them to capture the panorama.

2.5.3.1. Systems using external mirrors

External mirrors are placed in a distance from the camera so the camera can capture the reflections of its surroundings. And then specialized software/hardware inverse the mirroring effect and stitch the reflections together composing a panoramic image.

Such a system is reported in [1]. The current system was developed for cylindrical pipe inspection and uses a double lobed mirror enabling it to have a 360-degree field of view, as shown in Figure 11. In order for the system to also obtain vertical information about the pipe, it is elevated accordingly. The images produced from each lobe usually have overlapped areas, giving the advantage to use epipolar geometry and thus the ability to recover depth. Of course, the system must be properly calibrated, which in this particular case is rather easy, considering it can be done in the lab with a specific calibrating pattern. This system is pretty efficient for pipe inspection but the images produced are of low-resolution, because of the static and the relatively small size of the mirror in comparison with its surroundings, so the images produced are blurry.

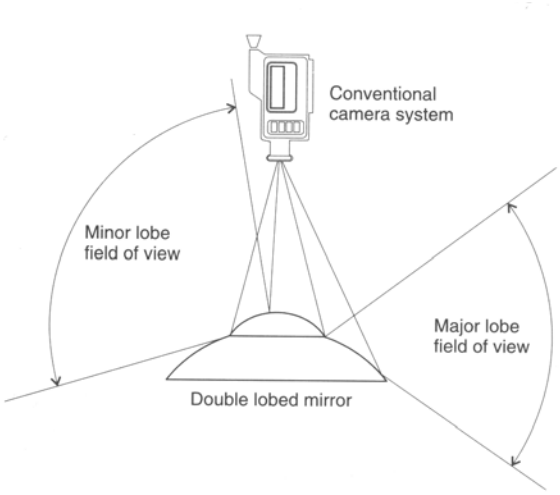


Figure 11:System using conventional camera and double lobed mirror
Image extracted from [1]

Another similar system was proposed in [14] which uses four CCD cameras aimed upward at four triangular mirrors. Special software reverses the mirror images and blends the individual pictures seamlessly into a single image. This system is able to display seven and a half panoramic images per second.

2.5.3.2. Systems using Rotational Cameras

These systems use some kind of mechanism to rotate the camera around its optical axis and thus have 360 degrees of view of the surroundings creating a cylindrical mosaic. This is in fact the evolution of the Video VR system presented in [12] which constructs cylindrical mosaics by rotating a camera mounted on a tripod around the optical axis of the camera. Multiple cameras can be placed in stacked configuration to enhance the field of view into the vertical axis. Such a system using a stepper motor to rotate the cameras is presented below:

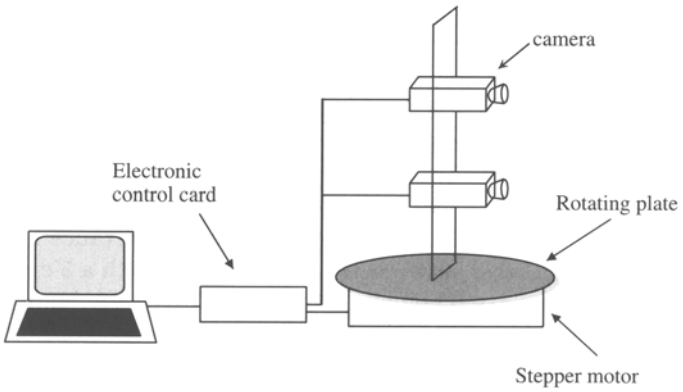


Figure 12: System using rotating cameras

Images extracted from [1]

In this particular configuration the image alignment is an easy task. As the motor is controlled by the computer the amount of rotation, or a very good approximation of it, is known. By possessing this knowledge before hand it is very easy to calculate the relative motion between two consecutive frames, as there is a one-to-one correspondence between the cylindrical coordinates, used to create the mosaic, with the world coordinates. More emphasis on this technique will be shown in the Software Mosaics section. The same principle is valid and for the case in which multiple cameras are used. Because the cameras are stacked, the height difference between them is known; allowing the system to compensate for the vertical axis, too. On the one hand the main drawbacks of this system are the cost as it usually needs more than one camera, the volume it occupies and that it is cumbersome, too. On the other hand is a straightforward approach, which gives very good results, as there are a few registration errors.

2.5.4. Summary

Regarding the problem of mirror choice, it is obvious there is no best solution. Spherical and Conic mirrors are relatively easier and cheaper to manufacture, but they suffer from multiple centers of projections, although they are successfully used in robot navigation. These sensors allow the robot to detect moving objects around it and also enable it to localize itself. The hyperboloid and parabolic mirrors are hard to manufacture and thus high-cost; but the images taken can be easily transformed into perspective images. On the one hand the system, which utilizes the parabolic mirror with a telecentric lens, is considered to be the most efficient in acquisition resulting in non blurred images but on the other hand the use of the telecentric lens increases the volume of the camera and the cost of the total system. The hyperboloid mirror experiences difficulties in designing, as it is under the constraint that one of the focal points must coincide with the camera center. Table 2 summarizes all the major features of each mirror in order to make the comparison easier, and in Figure 10 the field of view of each camera is shown so that the reader can have a more clarified opinion about each mirror.

The fact that the utilization of mirrors tends to produce low-resolution and possibly blurred images caused by the inability of the mirrors to provide high-resolution reflections and the weakness of the CCD sensors of the camera must be also taken in consideration. For the creation of high-resolution panoramic imaging systems, like the ones using rotational cameras, some software methods must be possibly examined.

Another drawback is that the local environment captured by each system may have several different intensity values, so a camera possessing a wide dynamic range is required. Current CCD sensors cannot fully support so large dynamic ranges, so information is lost in the process. The rotational systems do not have this problem because they capture part of the view each time but they are too expensive and too large, minimizing their ease of use and popularity among hardware systems.

3. Motion Estimation

Motion estimation problem tantalizes researchers for many years now. The relative motion existing between two consecutive frames derives from the 3d motion of the camera and other objects moving in the capturing scene projected in a 2d space, which is the image plane and thus resulting in 2d motion. The estimation of 2d motion, which is known under the name of "optical flow" or "apparent motion", is an ill posed problem as we try to recover the motion from a poorer representation, and so usually some constraints are imposed providing the possibilities of estimating it. The two most common constraints imposed for the estimation of optical flow are the:

- Data conservation constraint
- Spatial coherence constraint

The data conservation constraint utilizes the fact that surfaces generally persist over time in a video sequence although they may exist in different locations in each frame. This assumption is often formulated as a first or second order constraint on image gradient. Another way to exploit the data conservation constraint is correlation techniques, which attempt to estimate the displacement of a region that minimizes the difference between pixels over a specific region. The most common technique is the "Block Matching Algorithm". However the data conservation constraint is not enough as it is easily violated in motion boundaries and when noise or reflections and shadows are present, so the spatial coherence constraint is imposed as well. The spatial coherence constraint derives from the assumption that the same motion is typically present in a region of the image and thus the neighboring pixels of the pixel for which the estimation is performed, must have a similar motion. It is obvious that this constraint is violated at object boundaries and much work has been done in the process of trace when the violation occurs and on how to reformulate it.

The other big problem is that the motion estimation problem is computationally expensive especially when large displacements occur between consecutive frames. A hierarchical framework for more efficient computation of the motion, even when large displacements are present, has been developed in [7] which utilizes a Laplacian Pyramid and computes each time the difference between the motion already calculated.

3.1. Hierarchical Motion Estimation

A general framework for hierarchical motion estimation has been proposed by Bergen in [7] which handles the motion estimation problem under the perspective of image registration. One can say that image registration or alignment and motion estimation are practically the same thing. The framework proposed includes four steps and can be put in practice in already existing motion estimation algorithms. The steps are:

1. Pyramid Construction
2. Motion Estimation
3. Image Warping
4. Coarse to fine Refinement

This framework proposes the construction of a Pyramid (usually Laplacian), in each level of which resized images of the original frames will be used; in the case of the Laplacian pyramid in each level the image is reduced by a factor of two in each dimension. So in the coarsest level there are the initial frames and in the finest level the lower-resolution frames. The process starts with the motion estimation between the images and the finest level, as shown in Figure 13. The motion parameters estimated are propagated to the next level and the corresponding image is warped with this information. Then the motion is computed between the warped image and the corresponding second frame, as far as the resolution is concerned. This process estimate the incremental motion, which is added to the propagated motion parameters. The process is repeated until the coarsest level is reached where the final motion parameters are extracted.

This approach increases the computational efficiency of the process, since large displacements are calculated in the low-resolution frames, which are small in dimensions, making the algorithm more robust. Also this framework implementation does not depend on the motion estimation algorithm which will be used in the process, making its application very easy.

The main drawback of this framework is the warping process because an interpolating function must be chosen to perform the warp thus loss of information exists. Also the higher the complexity of the interpolation function the higher the computational cost.

However this framework is very useful especially in cases that large motions are present as it can cope with them more easily. Its easy implementation and flexibility make it very popular in the development of motion estimation algorithms.

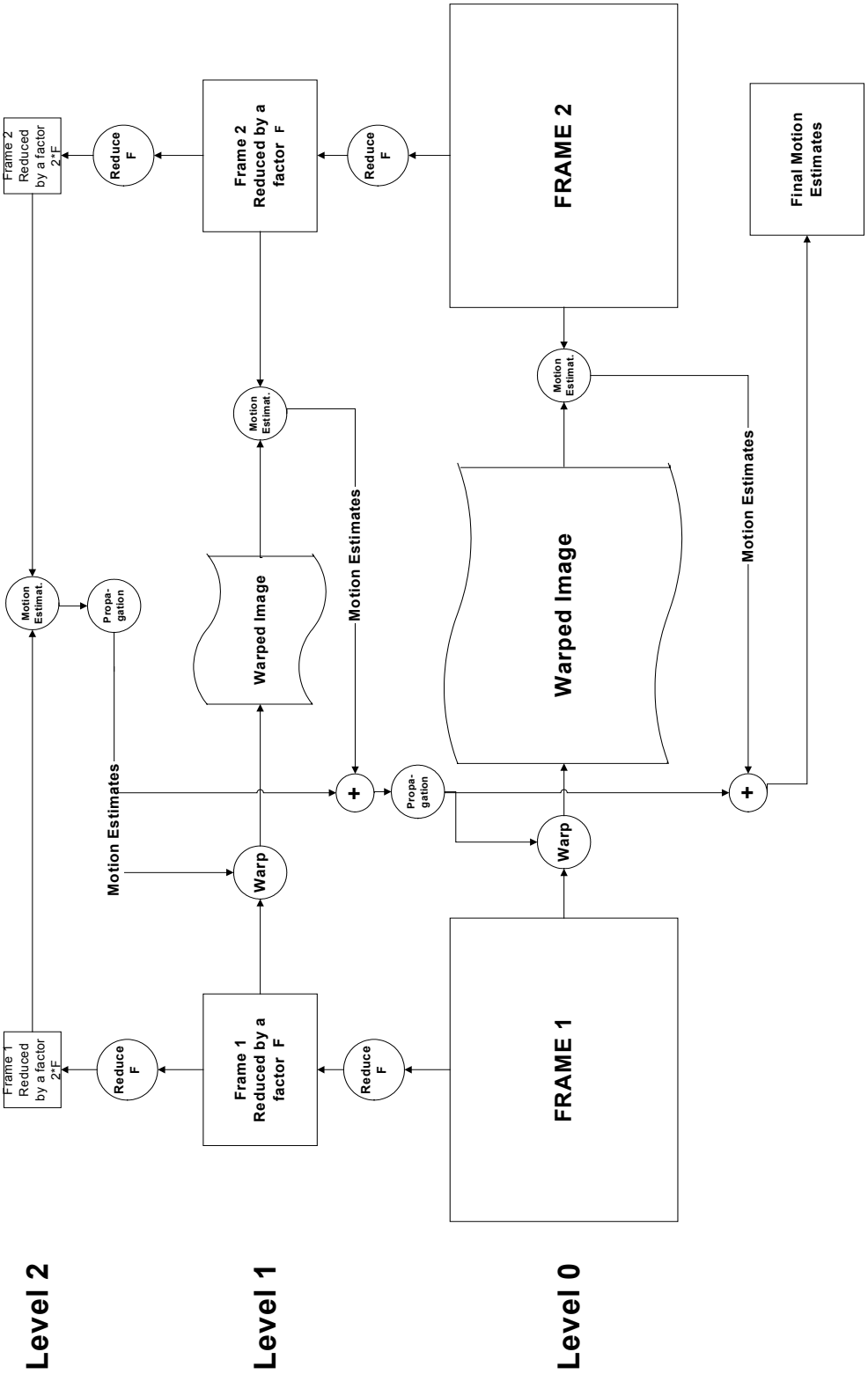


Figure 13 Hierarchical Motion Estimation Framework Flowchart with level=2

3.2. Gradient Methods

The data conservation constraint is based on the observation of the intensity of a pixel remaining constant over time. This, in general, is not always true as illumination conditions may change over time or frames may be captured under different exposure settings. So some pre-filtering is required such as a "Gaussian" filter, and a histogram equalization technique.

If $I(x, y, t)$ is the image intensity at a point with coordinates (x, y) at time t then the constraint can be expressed under the intensity constancy equation:

$I(x, y, t) = I(x + dx, y + dy, t + dt) = I(x + udt, y + vdt, t + dt)$ Where u, v are the horizontal and vertical motion vectors of the point and dt is relative small.

Gradient-based approaches proceed by taking the Taylor series expansion of the right hand term resulting in:

$I(x, y, t) = I(x, y, t) + I_x udt + I_y vdt + I_t dt + c$ where I_x, I_y, I_t are the first partial derivatives of I with respect to the x, y, t and c containing the high order terms. By simplifying the equation and by eliminating the dt by division, the standard optical flow constraint equation derives:

$I_x u + I_y v + I_t = 0$ yielding to an under-constrained problem as there is one equation for two unknowns. The standard optical flow constraint equation constrains the motion vector to lie in the direction of the image gradient, which is called "normal flow"; in order to recover a unique motion vector additional constraints are required.

The constraint usually used is the spatial coherence constraint or else smoothness constraint. This constraint implies that the motion in the neighboring pixels must be similar with the pixel examined and it can be implemented in various ways.

The first implementation was a regression approach by Horn in [15] which was an attempt to minimize the sum of squares of the Laplacians of the x,y components of the flow.

The Laplacians are expressed by: $\nabla^2 u = \frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}, \nabla^2 = \frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}$

and can be approximated by: $\nabla^2 u \approx k(\bar{u}_{i,j,t} - u_{i,j,t}), \nabla^2 v \approx k(\bar{v}_{i,j,t} - v_{i,j,t})$

where the $\bar{u}, \bar{v}$ are the local averages and are defined as:

$$\bar{u}_{i,j,t} = \frac{1}{6}(u_{i-1,j,t} + u_{i,j+1,t} + u_{i+1,j,t} + u_{i,j-1,t}) + \frac{1}{12}(u_{i-1,j-1,t} + u_{i-1,j+1,t} + u_{i+1,j+1,t} + u_{i+1,j-1,t})$$

$$\bar{v}_{i,j,t} = \frac{1}{6}(v_{i-1,j,t} + v_{i,j+1,t} + v_{i+1,j,t} + v_{i,j-1,t}) + \frac{1}{12}(v_{i-1,j-1,t} + v_{i-1,j+1,t} + v_{i+1,j+1,t} + v_{i+1,j-1,t})$$

and thus can be calculated by convolving the mask $\begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$ with the motion vectors matrices respectively.

So, the problem finally becomes a minimization problem of the error functional:

$\varepsilon^2 = (I_x u + I_y v + I_t)^2 + \lambda^2[(\bar{u} - u)^2 + (\bar{v} - v)^2]$ where λ is a constant and denotes the relative importance factor between the two constraints. By using calculus of variations, an iterative scheme for calculating the optical flow derives:

$$u^{n+1} = \bar{u}^n - I_x \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\lambda^2 + I_x^2 + I_y^2}$$

$$v^{n+1} = \bar{v}^n - I_y \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\lambda^2 + I_x^2 + I_y^2}$$

where (u^n, v^n) are the optical flow estimation in iteration n and $(\bar{u}^n, \bar{v}^n)$ are local averages at iteration n.

Instead of using the above first order smoothness constraint, a second order constraint can be used in the exact same way. For briefness reasons, only the constraint for the horizontal motion will be examined as the process is identical for the horizontal and vertical components of the motion. So the second order constraint for the horizontal component is:

$\varepsilon_s(u) = u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2$ where the s in $\varepsilon_s(u)$ denotes that the constraint is a smoothness constraint and u that is only for the horizontal component of motion. The subscripts indicate the second partial derivatives of the flow. Using a four-neighborhood region the derivatives can be formulated as:

$$u_{xx} = u_{i,j-1} + u_{i,j+1} - 2u_{i,j}$$
$$u_{yy} = u_{i-1,j} + u_{i+1,j} - 2u_{i,j}$$
$$u_{xy} = u_{i+1,j} + u_{i,j+1} - u_{i,j} - u_{i+1,j+1}$$

in which the smoothness constraint is minimized when the second derivatives are zero.

Another approach has been proposed by Nagel in [16] which does not use a smoothness constraint but can compute the optical flow directly in the "gray corners of an image" by calculating the second order derivatives and it is expressed as:

$$\begin{bmatrix} I_{xx}(\mathbf{x},t) & I_{yx}(\mathbf{x},t) \\ I_{xy}(\mathbf{x},t) & I_{yy}(\mathbf{x},t) \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} I_{tx}(\mathbf{x},t) \\ I_{ty}(\mathbf{x},t) \end{pmatrix} = 0$$

3.3. Block Matching

The block-matching algorithm exploits the constraints mentioned too but in a different way from the gradient based methods and it is pretty straightforward. Firstly, a reference frame must be chosen between the two frames, whose motion needs to be calculated. Usually the first frame occurring in time is selected, as forward motion estimation is more common. Then the reference image is divided into non-overlapping blocks of M by N pixels, typically M and N are equal and their common values are eight or sixteen. For every block in the reference image a search process is performed in the other image to find the best region that matches the block. In order to match the block a matching criterion is used based on the intensities of the block and the region currently examined. The procedure of searching the whole image is computationally expensive and usually needless, therefore a limit to the maximum motion is imposed, which is denoted by *d*. In the utilization of the block-matching algorithm certain assumptions are obtained. Firstly all the pixels in a block are assumed to have the same motion vector and thus the whole block undergoes global translational motion.

Block matching is a very common algorithm used in video compression and in the MPEG standards and has many variations. The basic categories of block matching algorithms can derive from two criterions: the measure used for matching the block and the search pattern used.

3.3.1. Matching Measures

The most common measures are:

- The Mean Square Error (MSE) denoted by:

where (u,v) are the coordinates of the top left corner of the block in the second image, M,N the block width and height respectively, S is the block and (x,y) are the coordinates with reference to the block. Finally $t,t-1$ denotes the previous and current frame.

- The Mean Absolute Difference (MAD) denoted by:

$$MAD(u,v)=\frac{1}{MN}\sum_{y=0}^{M-1}\sum_{x=0}^{N-1}\left|S(x,y;t)-S(x+u,y+v;t-1)\right|$$

where (u,v) are the coordinates of the top left corner of the block in the second image, M,N the block width and height respectively, S is the block and (x,y) are the coordinates with reference to the block. Finally $t,t-1$ denotes the previous and current frame.

- The Normalize Cross-correlation Function (NCF) denoted by:

$$NCF(u,v)=\frac{\sum_{y}\sum_{x}S(x,y;t)S(x+u,y+v;t-1)}{\sqrt{\sum_{y}\sum_{x}S^2(x,y;t)}\sqrt{\sum_{y}\sum_{x}S^2(x+u,y+v;t-1)}}$$

where (u,v) are the coordinates of the top left corner of the block in the second image, M,N the block width and height respectively, S is the block and (x,y) are the coordinates with reference to the block. Finally $t,t-1$ denotes the previous and current frame.

In theory the NCF yields the best results especially for pattern recognition although in video compression and other applications the MAD is more commonly used as it yields good results and is computationally less expensive. The MSE was used in early stages of the development of block matching algorithms and became obsolete by the MAD.

The block that has the minimum measure will be chosen as the most appropriate corresponding block in the latter image.

3.3.2. Search Algorithm

Block matching is performed as a means of matching the intensities of a block from the reference image with a block of the latter image. Finding the best match can yield to motion information extraction. The most straightforward algorithm for searching the latter image is the full search and it also has the best results. The reference block is compared to all the possible blocks of the latter image in a distance d from the coordinates of the reference frame. In Figure 14 the process is clarified. The gray rectangular is the corresponding position of the reference block in the latter image. The black dots represent the position of the left top corner of each block which is compared with the reference block. In Figure 14 a maximum displacement $d = 2$ is used.

To search for larger displacements the upper left corner of the block is displaced in a similar manner. For a n-pixel displacement 2*n+1 comparisons are required and so (2*n+1)*M*N pixel operations where M,N are the dimension of the block. The use of full search is generally avoided as it is computationally very expensive but it gives the best results and it is straightforward. Other search algorithms have been developed in order to reduce the computational cost of the algorithm and to perform block-matching more efficiently, too.

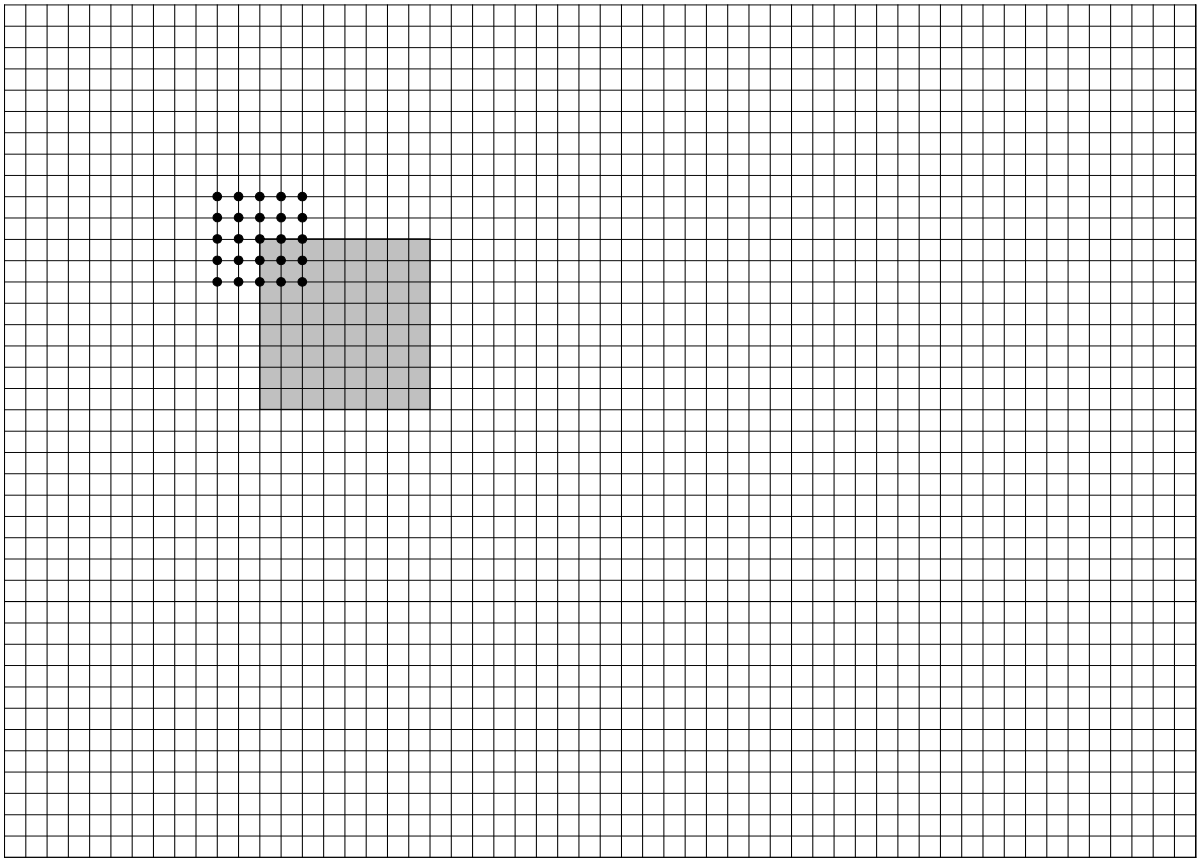


Figure 14: Block Matching Full Search with maximum displacement of two

The search algorithms that have been developed may be more efficient than the full search but they are sub optimal. This means that the best result is not always found as not all the possible displacements are examined. A summary of search algorithms used in block matching can be found in [17]. Below some indicative search algorithms other than the Full-Search will be discussed.

3.3.2.1. Three Step Search

This algorithm was introduced in 1981 and became popular because of its efficiency and near optimal performance. The algorithm named after the procedure can be divided in three steps.

- On the first step the central block and eight blocks, at a distance of the maximum displacement from the central block, are selected for comparison and the corresponding measures are computed.
- On the second step the center is moved to the block which had the minimum measure of the nine previously selected blocks, and the maximum displacement is halved.
- The third step repeats the first two steps until the maximum displacement becomes smaller than one.

A specific convergence of the algorithm is shown in Figure 15. The different colors in the dots represent different steps.

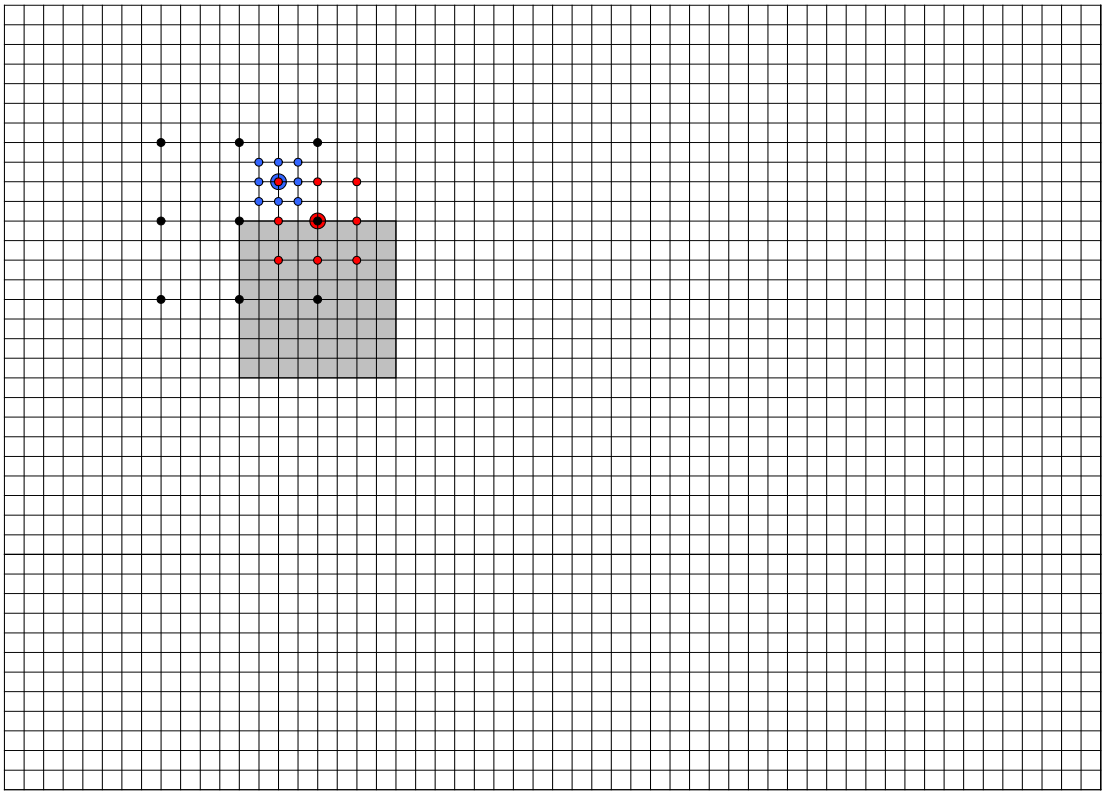


Figure 15: Block Matching Three Step Search with initial maximum displacement of four Black 1st Step, Red 2nd step, and Blue 3rd step.

3.3.2.2. Two dimensional Logarithmic Search

This algorithm is of the same "age" as the Three step algorithm and they are similar. Although this algorithm has a little more difficult implementation, as it requires more steps, it can yield more accurate results especially when the maximum displacement is large. The steps of the algorithm are:

- On the first step five candidate blocks are selected for the comparison: The central block and the four blocks at a distance that equals the maximum displacement from the center and form a cross shape. (More details in Figure 16)
- On the second step, if the best match according to the measure is in the center then the maximum distance is halved. Alternatively if one of the other blocks is the best match, the center is transferred to that point and the first step is repeated.
- Finally, when the maximum distance becomes one the measures of all the nine blocks around that particular point are calculated and the best match is picked as the best result.

In the following figure a specific convergence of the algorithm is shown. Again different colors represent different steps.

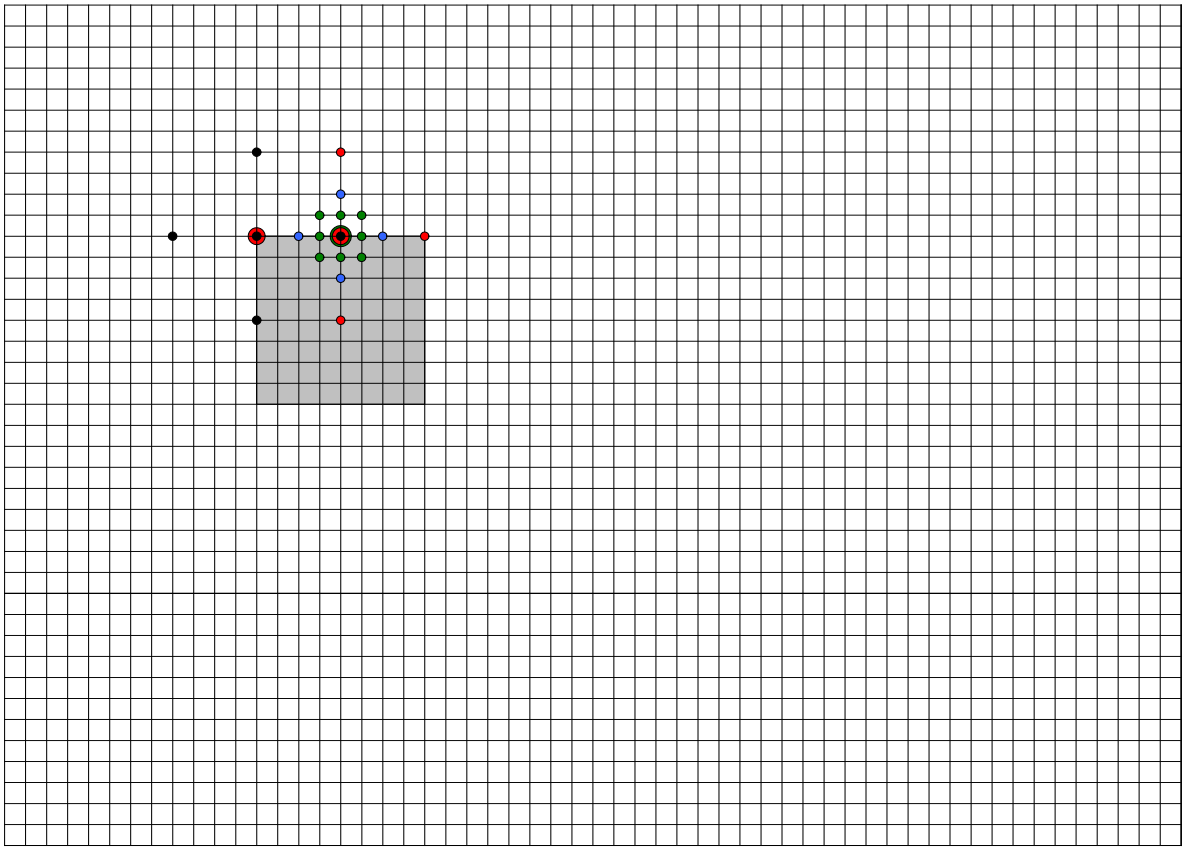


Figure 16: Block Matching Two Dimensional Logarithmic Search with initial maximum displacement of four Black 1st Step, Red 2nd step, Blue 3rd step, and Green Final step.

3.3.2.3. Summary

The block matching technique is an efficient technique to recover 2d translational motion from video sequences. It exploits the constraint that the motion is the same over a region (the block in this case) and that the intensity of the block will be constant over time. Block matching even in its more computational efficient forms is still expensive to use. For example the 80% of the time taken for video compression is the motion estimation process. In order to speed up the process the hierarchical motion estimation framework may be applied and this yields to good results in lower times.

4. Integration

Integration is the procedure of synthesizing the mosaic image from the video sequence and in order to proceed with this operation the frames in the video sequence must be aligned. The main problem which arises is that many frames of the video sequence may overlap over a specific region and thus the problem is the choice of the algorithm which combines the whole information and has the best results. A common challenge that must be faced is the one of the presence of moving objects in the scene. If moving objects are present some information they contain may be visible in the final mosaic, thus resulting in a "ghosting" effect. In the following section some common integration techniques will be discussed and ways to correct common errors occurring in panoramic view construction.

4.1. Common Integration techniques

The most common techniques used for image integration are usually some kind of filters used among the overlapping pixels of the frames. This occurs because more than one frame may have information for one point (x,y) in the mosaic and so a choice to blend in all this information must be made. Bellow a series of techniques for image integration are presented as gathered in [2].

1. A regular temporal average of the intensity values among the overlapping regions of frames of the video sequence may be used. This means that the intensity value of each point (x,y) in the mosaic is the average intensity value of all the pixels in the frames representing the same point of the aligned images.

2. A temporal median filtering applied in the same way as above. Both a temporal average and a temporal median applied to a registered scene sequence will produce a panoramic image of the dominant background scene, where moving objects either disappear or leave ghost-like traces. Temporal averages usually result in blurrier mosaic images than those obtained by temporal medians.

3. A weighted temporal median or a weighted temporal average where the weights decrease with the distance of a pixel from its frame center. This scheme aims at ignoring alignment inaccuracies near image boundaries due to the use of low order 2D parametric transformations

4. A weighted temporal average where the weights correspond to the smoothness areas of the motion computed in the motion estimation process. This scheme prefers the dominant background data over foreground data in the mosaic construction, and therefore gives less ghost-like traces of foreground objects, and a more complete image of the dominant background scene.

5. A weighted temporal average where the weights correspond to the motion discontinuities computed in the motion estimation process. This scheme prefers the non-dominant foreground data over background data in the mosaic construction. The mosaic image constructed by applying such an integration method would contain a panoramic image not only of the scene, but also of the event that took place in that scene sequence. This type of mosaic is called an “event mosaic" or a “synopsis mosaic", as it provides a snapshot view of the entire synopsis in the sequence. This kind of mosaic can be very useful for rapid browsing and surveillance

6. Integration in which the most recent information, i.e., the one which is found in the most recent frame, is used for updating the mosaic. This is particularly useful in the dynamic mosaic construction. Of course, if desired, the update can be more gradual, e.g., a decaying temporal average which give more weight to more recent information, and tends to forget information more distant in time.

7. Integration in which only the new information is added to the mosaic. This is the exact opposite from the integration with the most recent information. In each mosaic only the new information extracted from the new frame will be added.

4.2. Super Resolution

Super resolution is a special integration technique, which was first developed under the topic of image restoration. The basic idea is that by taking a number of images from a scene with small displacements, higher sample rates can be achieved than the ones of a single image as sub-pixel information can be extracted. The projection of a high resolution 3d scene into a 2d planar surface, the image, does not only lose information from the projection but also from the digitization process because the final image is a discrete representation with accuracy of a pixel. The super resolution algorithms estimate the sub-pixel motion and integrate the information from all the images to construct a higher resolution image.

The first step that must be done is the sub-pixel image registration between the frames as is also done in the mosaicing process and then the combination of the information. It is very easy to apply

super resolution algorithms to the overlapping regions of the video sequence in order to produce a higher resolution mosaic than the one obtained by the schemes discussed above.

The super resolution algorithms are strongly interlocked with the image registration process, as poor estimation of the motion between the consecutive frames will definitely lead to poor results. An indicative algorithm is the one proposed by [18]. The basic idea of the algorithm is based on *Iterative Backward Projecting – IBP* which derives from computer aided tomography. The algorithm has as initial values the averaging of the pixels of the frames in the corresponding locations and iteratively simulates the image process to obtain a set of low-resolution images, related to the super resolution image; it calculates the error between the simulated images and the original frames and then re-projects the error back to the high-resolution image.

Applications of mosaics using super resolution techniques can be found in [19, 20]

4.3. Error Elimination

The most common errors occurring in the construction of mosaics is because of the different exposure settings of each frame. A lot of modern cameras automatically change the exposure settings during capture to achieve better quality of the images captured. But this introduces another problem in mosaic construction, because if high exposure difference between frames exists it will be visible in the "seam" of the frames and it can also confuse some motion estimation algorithms working under the intensity constraint. A histogram equalization procedure is a good method to avoid this type of errors but is not always efficient. A better algorithm is proposed in [21] which is a block-based exposure adjustment technique. In this process first the image is divided into blocks of a constant size then within each block the quadratic transfer functions are computed using the least squares error over the block area with intensity criterion. The algorithm is iterative and uses the adjusted composite as the reference for the next step. Other effects, such as vignetting are also compensating in this method

Another common error occurring in the process of mosaic construction is the ghosting phenomenon described above, which is due to the existence of moving objects in the video sequence. Several methods have been proposed to eliminate this kind of error among which the most common is the use of a temporal median between the overlap pixels in the integration process. This method is pretty straightforward and gives good results when more than half the frames contain non-moving pixels. So is inappropriate for mosaics created from a small video sequence, as there is not much

overlap between the frames. Another method has been proposed in [13] for compensating ghosting effects utilizing the estimation of the optical flow followed by a multi way morph but it is too restricting and computationally expensive.

The method proposed in [21] seems to yield good results. First, the regions where the ghosting phenomenon occurs are defined by applying a threshold in the difference between the overlapping pixels. Then a labeling algorithm is performed to identify and label contiguous regions followed by a grouping of the overlapping regions. In the end, only one region of an image is selected to participate in the final mosaic. The choice of the region is performed by making a graph from each group of the overlapping regions and by solving for the minimum weight distance. The weights are set with regards to the size and other features of the region; for example large regions have high weights in order to avoid discontinuities.

5. Implementation

For the implementation of the mosaicing procedure; special software has been developed by the writer in the C++ programming language. The C++ programming language was chosen because the purpose of the software was not only to implement a mosaic procedure but also to implement an easy to use and extensible library for manipulating images and videos. The total source code exceeds the 120 Kilobytes and is widely available via internet from the homepage of the Digital Image and Signal Processing and LaboratorY (DISPLAY) of the Technical University of Crete: http://www.display-systems.tuc.gr.

The implementation was divided into two sub processes the motion estimation algorithm and the implementation of the mosaic based on the motion vectors extracted. A basic block diagram of the algorithm is shown in Figure 17. For the first step the initial global motion parameters are set to zero and the previous mosaic is set to be the first frame of the video sequence. The motion parameters are then estimated between the frames of the mosaic and the dominant motion between them is calculated. The dominant motion is found by computing the angles of the motion vectors and then the average of the motion vectors that have the most common angle. The new global motion parameters are added to the previous global motion parameters so the motion between the next frame and the previous mosaic becomes known. The global motion parameters are updated and the integration of the new mosaic is performed between the new frame and the previous mosaic. For the construction of the dynamic mosaics, the mosaic resulting from each step is stored and then the full set of the stored mosaics is presented as a video sequence.

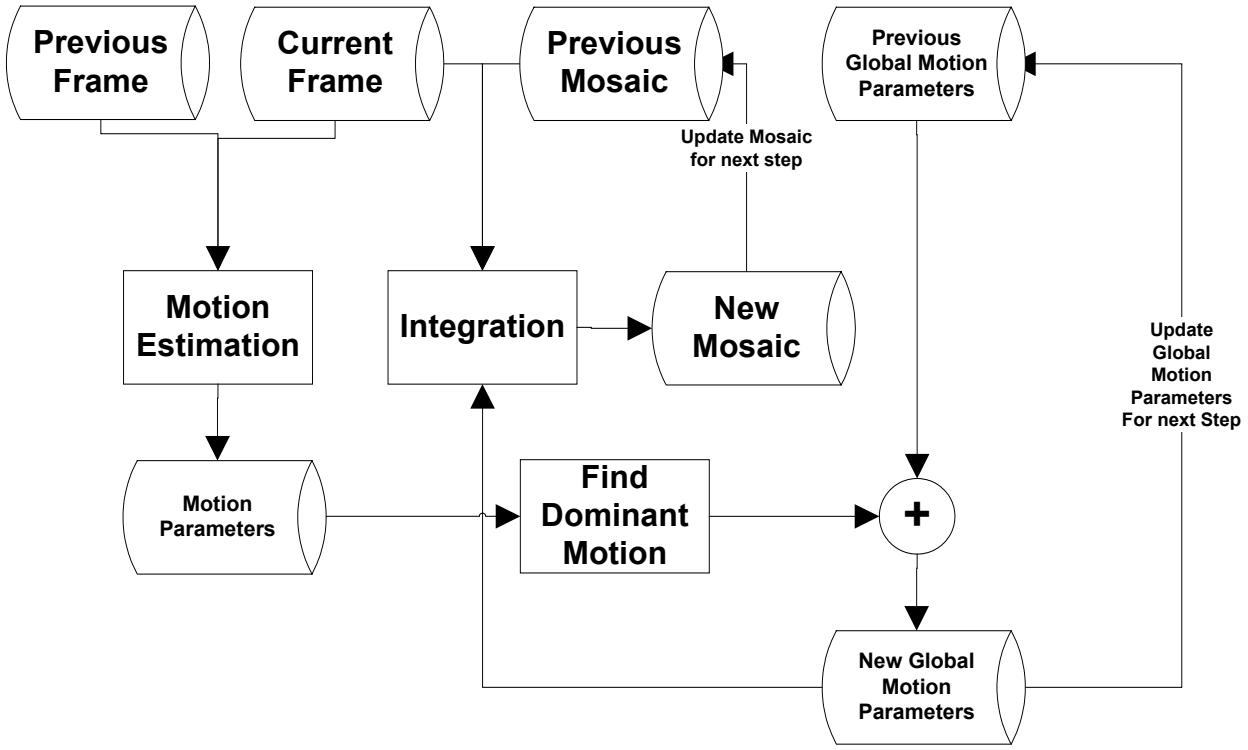


Figure 17: Static Mosaic algorithm block diagram

5.1. Motion Estimation

For the motion estimation sub process two algorithms have been developed. A full search block-matching and a robust gradient algorithm proposed by Black in [22]. The full search algorithm was preferred over the other block-matching algorithm as computational efficiency was not the issue of this thesis.

5.1.1. Full Search Block Matching

A block diagram of the algorithm is shown in Figure 18. The algorithm considers the dimensions of the block and the maximum search distance as input parameters and calculates the mean absolute error for each corresponding block. The block with the minimum absolute error denotes the displacement between the two frames.

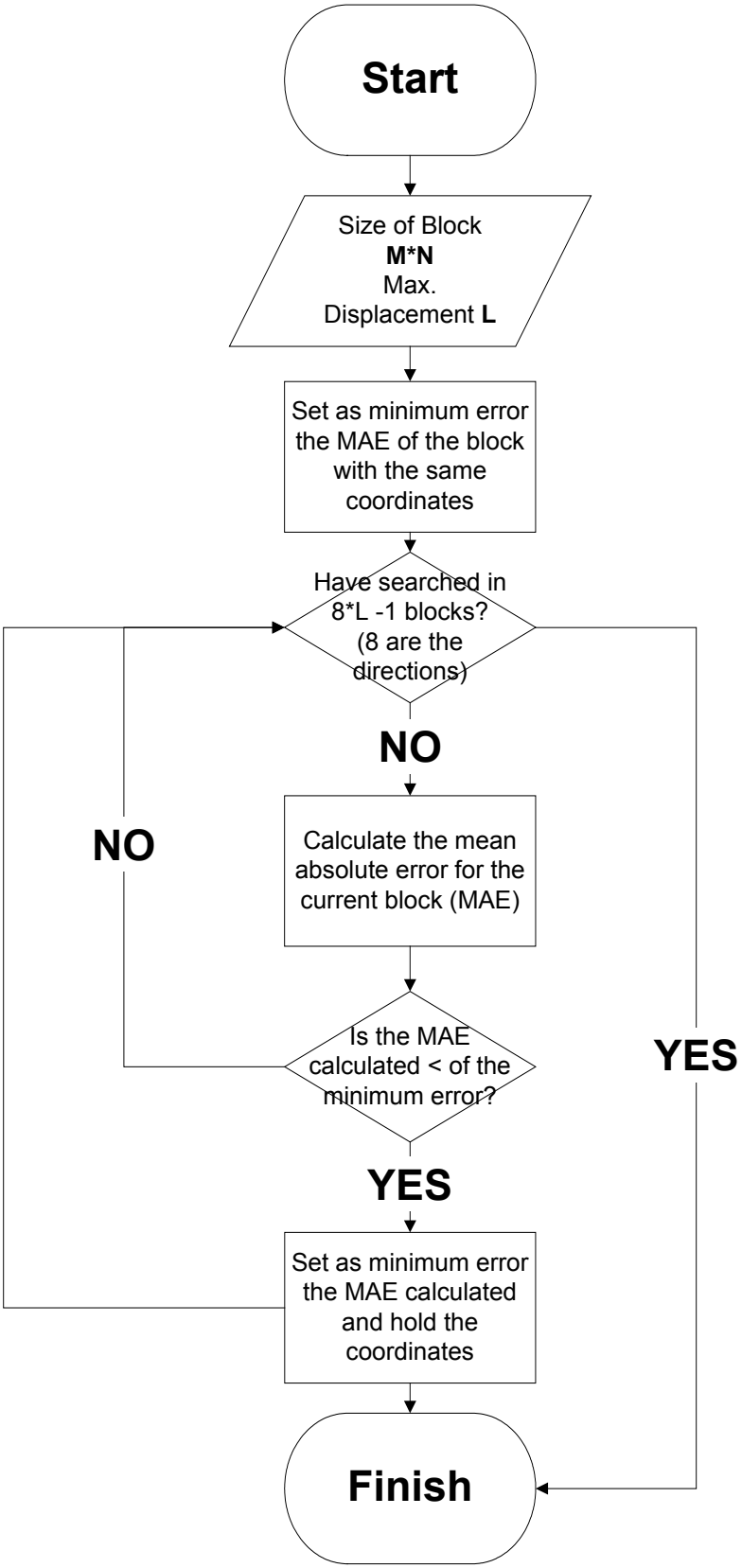


Figure 18: Full Search Block Matching Flowchart

5.1.2. Black and Anandan Algorithm

The Black and Anandan Algorithm is a first order gradient based algorithm utilizing the Bergen [J.R. Bergen, 7] framework to cope with large displacements and a robust estimator for the minimization of the constraints that are set.

It is shown that for the estimation of the motion vector the minimization of the error functional that follows must be performed:

$\varepsilon = p(I_x u + I_y v + I_t, \sigma_1) + \lambda[p(\bar{u} - u, \sigma_2) + p(\bar{v} - v, \sigma_2)]$ where p is the error measure which for this case is the Lorentzian Maximum Likelihood estimator. The Lorentzian estimator is denoted by:

$$p(x,\sigma) = \log\left[1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right]$$

$$\psi(x,\sigma) = \frac{2x}{2\sigma^2 + x^2}$$

$$\omega(x,\sigma) = \frac{2}{2\sigma^2 + x^2}$$

where p is the Lorentzian function, ψ is the cost function and ω is the influence function.

The minimization of the error is performed by utilizing simultaneous over relaxation method. In this case the minimization of the error is performed regarding the u,v parameters. The iterative updates equations for minimizing ε are:

$$u^{(n+1)} = u^{(n)} - \omega \frac{1}{T(u)} \frac{d\varepsilon}{du}$$

$$v^{(n+1)} = v^{(n)} - \omega \frac{1}{T(v)} \frac{d\varepsilon}{dv}$$

where n is the nth iteration, ω is an over relaxation parameter that is used to correct the estimates $\frac{d\varepsilon}{du}, \frac{d\varepsilon}{dv}$ are the first partial derivatives of ε denoted by:

$$\frac{d\varepsilon}{du} = \sum[I_x \psi(I_x u + I_y v + I_t, \sigma_1) + \sum \lambda \psi(\bar{u} - u, \sigma_2)]$$

$$\frac{d\varepsilon}{dv} = \sum[I_y \psi(I_x u + I_y v + I_t, \sigma_1) + \sum \lambda \psi(\bar{v} - v, \sigma_2)]$$

the terms $T(u), T(v)$ are the upper bound of the second order partial derivatives of ε which are maximized when both the constraints are zero:

$$T(u) = \frac{I_x^2}{\sigma_1^2} + \frac{\lambda 4}{\sigma_2^2}$$

$$T(v) = \frac{I_y^2}{\sigma_1^2} + \frac{\lambda 4}{\sigma_2^2}$$

For $\omega < 2$ the algorithm is proved to converge.

A flowchart of the motion estimation algorithm is shown in Figure 19.

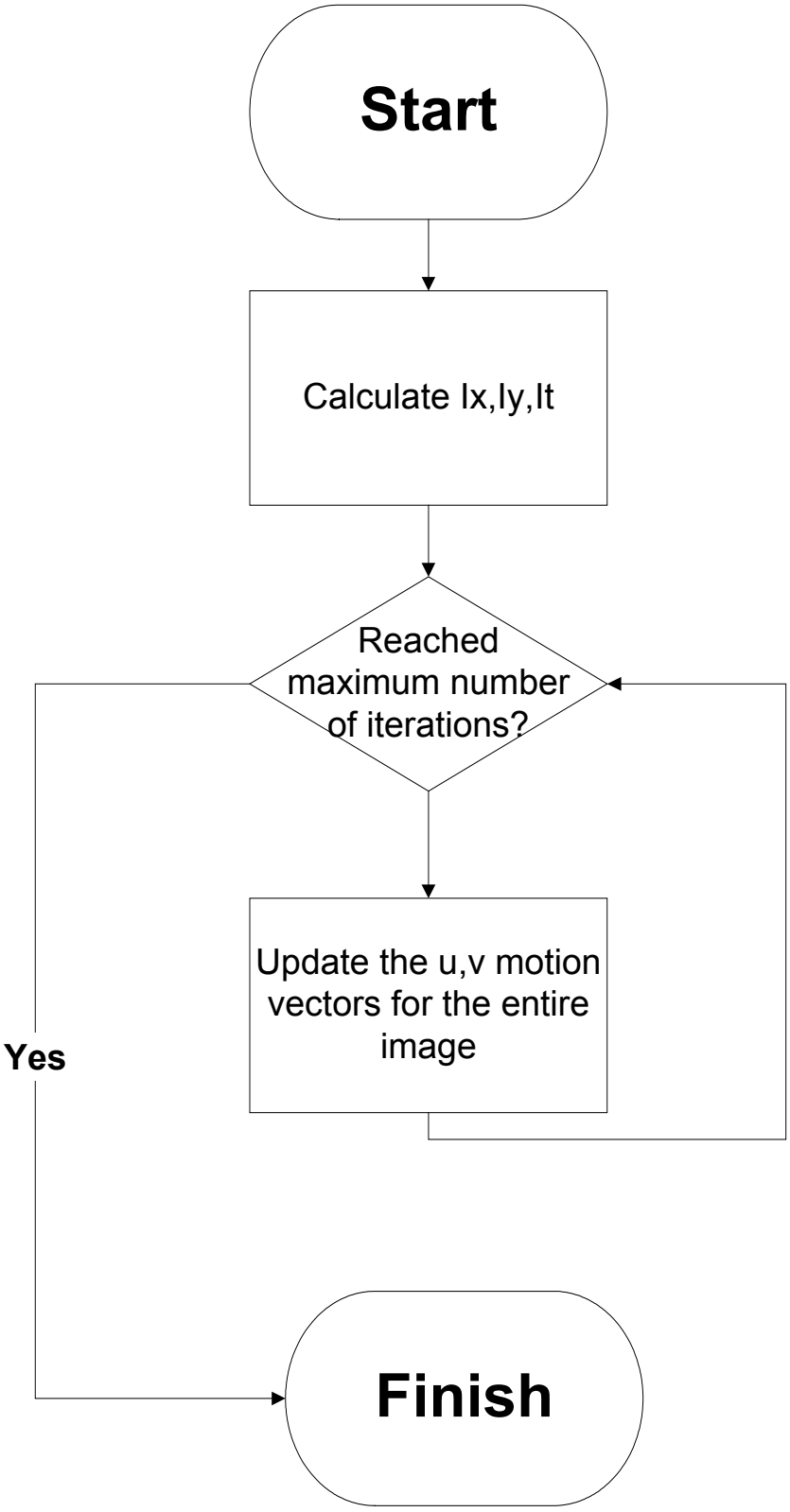


Figure 19: Black Anandan algorithm flowchart

5.2. Integration

For the integration scheme three algorithms have been implemented:

- A temporal average
- Most Recent information
- New Information only

Because the motion between the two consecutive frames is of sub-pixel accuracy; an interpolating function must be used to compute the intensity in the reference frame coordinates. The interpolating function that was implemented was a Bicubic B-spline interpolation function which is denoted by:

$$F(i', j') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} F(i+m, j+n) R(m-dx) R(dy-n)$$

$$where\ R(x) = \frac{1}{6}\left[P(x+2)^3 - 4P(x+1)^3 + 6P(x)^3 - 4P(x-1)^3 \right]$$

$$and\ \mathrm{P(x)} = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases}$$

i', j' are the integer coordinates in the reference frame i, j are the integer part of the coordinates (x, y) to be calculated and dx, dy are the difference between them: $dx = x - i, dy = y - j$. By applying the bicubic interpolation the integration schemes are easy to implement.

5.3. Results

Below are shown the results of the algorithm developed for the current thesis, for two different sequences. The first is a sequence used by Zomet at [Zomet and Peleg, 20] which consists is 5 second video captured at 15 frames per second and consists of 75 frames of dimensions of 160x120. The second sequence is representing the Technical University of Crete (T.U.C) dormitories and was captured using a webcam at 15 frames per second also. It consists of 64 frames of dimensions of 320x240.

5.3.1. Static Mosaics

5.3.1.1. Shelf Sequence

The results for this sequence form a single image of dimensions 543x142 for the Black Anandan and 568x140 for the Block Matching Algorithm.

Figure 20: Shelf sequence using Black Anandan and averaging

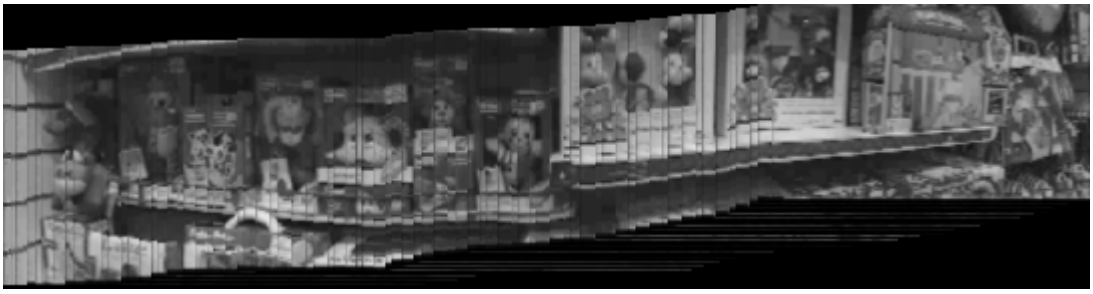


Figure 21: Shelf sequence using Black Anandan and most recent information

Figure 22: Shelf sequence using Black Anandan and only new information

Figure 23 : Shelf sequence using Block Matching and averaging

Figure 24: Shelf sequence using Block Matching and most recent information

Figure 25: Shelf sequence using Block Matching and only new information

5.3.1.2.

T.U.C Dormitories Sequence

The results for this sequence form a single image of dimensions 499x263 for the Black Anandan and 552x255 for the Block Matching Algorithm.

Figure 26: T.U.C. dormitories sequence using Black Anandan and averaging

Figure 27: T.U.C. dormitories sequence using Black Anandan and most recent information

Figure 28: T.U.C. dormitories sequence using Black Anandan and only new information

Figure 29: T.U.C. dormitories sequence using Block Matching and averaging

Figure 30: T.U.C. dormitories sequence using Block Matching and most recent information

Figure 31: T.U.C. dormitories sequence using Block Matching and only new information

Having the final mosaic, it is very easy to focus directly on a specific region of the image. For example, in Figure 32 the focus is set on the region of the dormitories which is a rectangular area of 200x100 pixels, with the coordinates of the upper left corner of the rectangular (180,100) with respect to the static mosaic.

Figure 32: Focused Region of Static Mosaic

5.3.2. Dynamic Mosaics

The results for the dynamic mosaics were obtained by using the most recent integration algorithm and are video sequences. Because a video sequence cannot be represented in paper three frames from the original sequence and the corresponding mosaics will be shown.

5.3.2.1. Shelf Sequence

Figure 33: Dynamic Mosaic of Shelf Sequence using Black Anandan Algorithm

Figure 34: Dynamic Mosaic of Shelf sequence using Block matching algorithm

5.3.2.2. T.U.C Dormitories Sequence

Figure 35: Dynamic Mosaic of T.U.C dormitories Sequence using Black Anandan Algorithm

Figure 36: Dynamic Mosaic of T.U.C dormitories sequence using Block matching algorithm

5.4. Summary

All the videos, in which the algorithm was applied, were captured using a handheld camera but no tripod or other special mounting device was used. Due to the nature of the video sequences the final mosaic is a plane mosaic, which was build using the frame-to-frame alignment. The black regions occurring in the mosaics appear because of lack of information in those particular regions and they are a common phenomenon in plane mosaics. That lack of information is unavoidable because of the arbitrary and non-smooth movement of the camera.

As it is obvious from the results the algorithm developed is not able to perform a fully "seamless" panoramic image although it is sufficient, as the purpose of this thesis was the proof of fact. This derives from the assumption that only translational motion is present in the video sequence and thus alignment errors occur which are because to the presence of other motions in the video sequence. This problem can be overcome by utilizing a more complex motion model from the ones discussed above. Having calculated the motion vectors for each pixel the problem becomes a minimization problem of finding the best set of parameters which best fit in the motion vectors computed. This is in fact an improvement of the dominant motion calculation algorithm.

The block matching algorithm is one of the most classic algorithms used for motion estimation and it is widely used by the MPEG compression standards. It yields pretty good results if only translational motion is present in the video sequence but cannot cope very well if rotations and change of the focal length (zooming) are present. Another drawback of the block matching algorithm is that it has pixel accuracy meaning that the resulting motion vectors will be integers which is a result of the search pattern that is utilized. It is reminded that the block each time is moved by one pixel in the corresponding location. This is obvious in the results as the mosaics obtained by using the block matching algorithm have more visible "seams" than the one obtained by the Black and Anandan Algorithm. (eg. Figure 26 and Figure 29)

The Black and Anandan algorithm is more robust since besides the motion estimation it can also detect the violations of the smoothness constraint, by applying a threshold in the robust formulation, to detect motion discontinuities. The necessity of the Bergen framework though, in order for the algorithm to be able to cope with large motion makes it even more computationally expensive than the Full search Block-Matching. This due to the fact that the motion estimation algorithm is performed as many times as the levels of the pyramid used. In our examples the algorithm was executed with a four level pyramid.

Regarding the integration schemes, the averaging process seems to yield the best results as it integrates the overlapping information and less seams are visible, although this comes with the cost of a blurring effect in the whole mosaic. The “most recent information” algorithm and its opposite “only new information” give more clear results as no averaging between the pixels is performed but also make the misregistration effects more visible. “The most recent information” is also used for the construction of the dynamic mosaics.

6. Conclusions and Future Work

The subject of mosaics was thoroughly examined. All the currently available methods have been extensively presented and studied. Two of the basic categories the static and the dynamic mosaics were implemented using a plane manifold and the frame-to-frame alignment. The results obtained by the algorithm were good as shown above and so proof that a mosaic can be obtained from a video sequence without any prior knowledge of the camera motion was given. In parallel an extensive study on currently available motion estimation methods was performed and some of them were implemented for comparison reasons. The main problem that appears in the construction process was the visible "seams" in the areas of the mosaic that the original frames overlap. The visible "seams" appear because of the wrong estimates of the motion in some pixels of the image. A better algorithm to compute the dominant motion can be developed in order to cope with this problem as in the purpose of this thesis only translational motion was assumed to exist in the video sequence. Using higher in order parametric motion model to fit the motion vectors, obtained by the algorithms developed, can yield to significantly better results but there was no implementation due to lack of time. Also a primitive motion segmentation can be performed in the video sequence before the mosaicing process, to limit the erroneous data which can be easily achieved by the Black and Anandan algorithm by applying a threshold in the robust formulation.

7. References

[1] *Panoramic Vision: sensors,theory, and Applications*: Springer, 2001.

[2] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient representations of video sequences and their applications," *Signal processing: Image Communication*, vol. 8, pp. 327-351, 1996.

[3] O. Faugeras, *Three-Dimensional Computer Vision, A geometric viewpoint*: MIT Press, 1993.

[4] C. S. Remi Megret, Walter Kropatsch, "Background Mosaic from Egomotion," presented at IEEE International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, 2000.

[5] H. Nicolas, "Optimal criterion for dynamic mosaicking," presented at Interntational Conference in Image Processing (ICIP'99), Kobe, Japan, 1999.

[6] H. Nicolas, "New methods for Dynamic Mosaicking," *IEEE Transactions on Image Processing*, vol. 10, pp. 1239-1251, 2000.

[7] P. A. J.R. Bergen, K.J. Hanna, R. Hing, "Hierarchical Model-Based Motion Estimation," presented at Second European Conference on Computer Vision (ECCV 92), 1992.

[8] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on Adaptive Manifolds," *IEEE Transanction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1144--1154, 2000.

[9] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, "Mosaicing New Views: The Crossed-Slits Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 741-754, 2003.

[10] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," presented at IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'97), San Juan, 1997.

[11] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha, "Universal Mosaicing using Pipe Projection," presented at IEEE 6th International Conference on Image Processing (ICCV), Bombay, 1998.

[12] M. C.-C. H. Ding-Yun Chen , Ming Ouhyoung, "VideoVR: A Real-Time System for Automatically Constructing Panoramic Images from Video Clips," presented at Modelling and Motion Capture Techniques for Virtual Environments: International Workshop, CAPTECH'98. Proceedings, Geneva, Switzerland, 1998.

[13] H. Shum and R. Szeliski, "Panoramic Image Mosaics," Microsoft Research, Technical Report 1997.

[14] V. Nalwa, "A true omnidirectional viewer," Bell Laboratories, Technical Report 1996.

[15] B. G. S. Berthold K.P. Horn, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.

[16] H.-H. Nagel, "On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results.," *Artificial Intelligence*, vol. 33, pp. 299-324, 1987.

[17] M. A. Deepak Turaga, "Search Algorithms for Block-Matching in Motion Estimation," Report for Mid-Term Project 1998.

[18] P. S. Irani M., "Improving resolution by image registration," *Graphical models and image processing*, vol. 53, pp. 231-239, 1991.

[19] D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98), Santa Barbara, 1998.

[20] A. Zomet and S. Peleg, "Applying super-resolution to panoramic mosaics," presented at IEEE Workshop on Applications of Comuter Vision (WACV), Princeton, 1998.

[21] M. Uyttendaele, A. Eden, and R. Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001), Hawaii, 2001.
[22] M. J. Black, "Robust Incremental Optical Flow (Phd Thesis)," Yale, 1992.