

Chapter 1
Background

God created human beings with two eyes and one nose (obviously there are two holes in a nose). Few questions arose in mind: why are there exactly two eyes for humans to see? Is there any difference between person having both eyes and person having only one eye working? This was the question raised during the beginning of this project. So, an idea to study the reason behind the pairing of Vision System was introduced.

In our third year we were involved in a project entitled – "AGV – Autonomous Guide Vehicle" which was based on Robotics. The main purpose of the project was to build an automated guided vehicle that means a vehicle that could trace its path automatically without any manual control, but with a manual control provision as an additional attribute. The project was a success, partly, because it could trace a predefined black line above a plain white background, i.e., only in an ideal environment. For the visual purpose, a single camera was used to detect the line for the vehicle to follow. But the main problem of using a single camera was to avoid the vehicle from crash if there were any kind of obstacles ahead. The distance of the object could not be calculated from a single 2-Dimensional image captured by the camera. It could detect obstacles only through the use of a pair of sonar located in front such that it would receive the reflected sound from the obstacle and redirect its path, which is very weak in the real system environment.

There was a proposal of using two cameras like human beings to solve the problem, but How? So for the part of Final year project we tried to study the mechanism of human beings using two eyes to compute the forward distance.

Being an Engineering student, the innermost detail of Human Visual system (i.e. the biological aspects) was not considered; rather an analogical study of Human Visual System with Computer Vision was identified in the study.

If we train a man to calculate the product of two values and ask him to do so by providing a value (greater than two digits, for example 498 Multiplied by 743), he

will take some time and give a result after few seconds, but we cant say that the result is accurate. But nowadays, since there are computers to do these things, same thing can be done by a computer in less than a second with 100% accuracy. So computers are really good enough to follow instructions. If we train a man to identify objects like Tree, House, Grass, etc and ask him to do so by showing some scenery, he will identify the objects within it accurately in a second. But training a computer to identify object is first of all a nightmare. And even after training it, there is no such computer (till now), which can accurately identify the objects by processing in a short time, not at least a wide range of complex objects. This does not mean that computer cannot really identify an object, but there is also chance that the result is incorrect.

This shows that Vision System in context of computer is still far behind as compared to other field. In other field, computers have already overtaken human beings¹. So, we thought of giving some effort to contribute in the field of Computer Vision. There are different areas in the field of Computer Vision. But in analogy to human perception, the concept of two-eye vision; a field known as Stereo Vision was what we were intended to deal with.

To have an innovation in the field of Stereo Vision, the researcher must have wide as well as depth knowledge of it. So we planned to find out what have been done till now in this field. This was an easy task, but not a complete one. If not every day, then in every month there are peoples developing the new algorithm of stereo correspondence, occlusion detection and recovering dense depth map.

For the first six months, we tried to collect and be acquainted with some of these algorithms. Reading and understanding whatever other writes is really a fun, but writing is something very difficult for reader. So, we started to write with something

¹ We have to accept that Computer has overtaken most of the Human Beings in context of doing the works. And this does not mean to humiliate Human Being, we have not forgotten that Human Beings have created Computer.

from the beginning. At the end of first six month, we came up with a Paper entitled “Stereo Vision: An Introductory Approach”[1.]. After having knowledge of wide range of stereo algorithm, we came up in decision to implement and test one of the algorithm described. So we chose to implement and test the Cooperative Algorithm [4.]

Finally, at the end of one year of research project entitled “Stereo Vision” we came up with enough knowledge and materials which might be considered as the first step towards the field of Stereo Vision that includes first version of Cooperative Algorithm for Stereo Matching implemented in Java.

Chapter 2

Introduction

2. 1. Mammalian Vision

Nature has given animals the physical attributes necessary for survival. Mammals generally come equipped with two eyes in front of their body and the complex brain which process the images received from the eyes to acquire information about their surrounding. Generally these pair of eyes are located either on the side of their head or in front.

The first one that is lateral placement of the eyes is essential to the survival of hunted animals or herbivorous animals (e.g., horse, rabbit, cow) as it allow them to increase side or peripheral vision (*see fig. 2.1*). Side vision (increased by lateral placement) is a sensitive detector for motion or movement. Peripheral vision allows creatures to effectively scan for danger. The rabbit must be constantly aware of its natural enemies while it eats your garden greens. At the first sign of danger, peripheral



fig. 2.1: Lateral placement of eye

vision, the motion detector system, alerts the rabbit that there is danger. The immediate reflexive response is for the rabbit to run. In case of a horse, their eyes are placed on either side of their head, which makes it possible for them to see in almost every direction at once except directly in front of their nose and directly behind their tail.

Faster moving carnivorous hunters do not need as much peripheral vision as the hunted. It is more important for hunters to locate their prey and accurately determine

the distance from themselves to that prey. Therefore, animals that hunt (e.g. carnivorous or meat eating animals, e.g. lion, cat) as well as humans have frontal placement of the two eyes in order to determine the exact location of their prey. The hunters sacrifice the large peripheral motion detection system afforded by side placement of the eyes in favor of the incredibly accurate depth perception system created by frontal placement of the eyes. (*see fig. 2.2.*)



fig. 2.2: Frontal placement of eye

2. 2. Human Vision

If we take a look around the room that we are in, we can notice how the various images and colors that we see update constantly as we turn our head and re-direct our attention. Although the images appear to be seamless, each blending imperceptibly into the next, they are in reality being updated almost continuously by the vision apparatus of your eyes and brain. The seamless quality in the images that we see is possible because human vision updates images, including the details of motion and color, on a time scale so rapid that a "break in the action" is almost never perceived. The range of color, the perception of seamless motion, the contrast and the quality, along with the minute details, that most people can perceive make "real-life" images clearer and more detailed than any thing seen on a television or movie screen. The efficiency and completeness of our eyes and brain is unparalleled in comparison with any piece of apparatus or instrumentation ever invented. We know this amazing function of the eyes and brain as the sense of vision.

Vision in human is a complicated process that requires numerous components of the human eye and brain to work together. The initial step of this fascinating and powerful sense is carried out in the retina of the eye. Specifically, the photoreceptor neurons (called photoreceptors) in the retina collect the light and send signals to a network of neurons that then generate electrical impulses that goes to the brain. The brain then processes those impulses and gives information about what we are seeing.

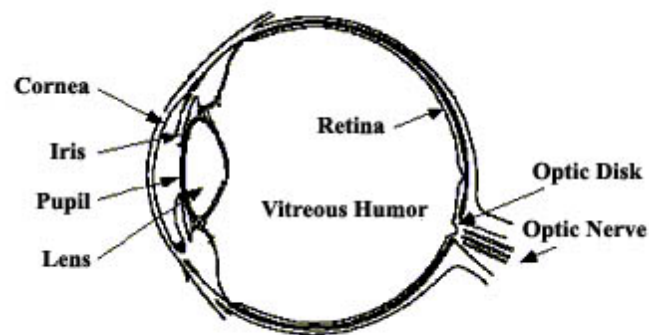


fig. 2.3.: Anatomy of human eye

Human beings as mentioned earlier generally come equipped with two eyes located side-by-side in the front of their heads. The close side-by-side positioning of each eye makes it possible to take a view of the same area from a slightly different angle. The two eye views have plenty in common, but each eye picks up visual information the other doesn't.

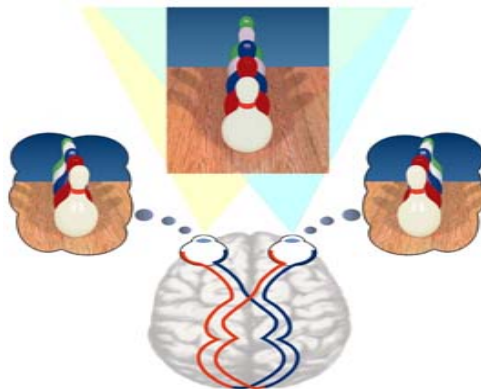


fig. 2.4.: Human eye using a pair of images to interpret correct vision. [22.]

Each eye captures its own view and the two separate images are sent on to the brain for processing. When the two images arrive simultaneously at the back of the brain, they are united into one picture. The mind combines the two images by matching up the similarities and adding in the small differences. The small differences between the two images add up to a big difference in the final picture. The combined image is more than the sum of its parts. It is a three-dimensional stereo picture.

2. 3. Computer Vision

Computer Vision, which is widely known for the machine vision, refers to the ability of the machine along with computer to perceive the visual information about the 3D real world through the use of images captured by camera. It is also sometimes known as Image Analysis, Scene Analysis or Image Understanding. Though, a precise and accurate definition of such a multi faceted discipline is hard to make, we can focus on some import issues.

- *The problems of computer vision:* The target problem is computing properties of the 3D world from one or more pair of digital images. The properties that interest us are mainly geometric (for instance, shape and position of solid objects).
- *The tools of computer Vision:* As the name suggests, computer vision involves computers interpreting images. Therefore, the tools needed by a computer vision system include hardware for acquiring and storing digital images in a computer, processing the images, and communicating results to users through interface.

Chapter 3

Image Processing Fundamentals

Image processing is a general term for the wide range of techniques that exist for manipulating and modifying images in various ways. Photographers and physicists can perform certain image processing operations using chemicals or optical equipment; however, our concern is solely with digital image processing, i.e., that which is performed on digital images using computers.

A Digital image is represented by a numerical matrix E , with N number rows and M number of columns. $E(i, j)$ denotes the intensity level (image brightness) at pixel (i, j) (i^{th} row and j^{th} column), and encodes the intensity recorded by the photo sensors of the CCD camera contributing to that pixel. $E(i, j)$ is an integer in the range $[0-255]$ in case of grayscale image.

Digital imaging actually predates modern computer technology; however, true digital image processing (DIP) was not possible until the advent of large-scale digital computing hardware. Now, more than three decades later after its early motivation in the NASA's space program 1960's, DIP finds application in areas as diverse as medicine, military reconnaissance and desktop publishing. Some examples of DIP might be contrast enhancement, removal of motion blur, image warping, etc. These processing requires several basic algorithms to be implemented in digital image to get required result. Some of such algorithms necessary for stereo algorithm to be explained in Chapter 5 are discussed here.

3. 1. Convolution and Correlation

Convolution and Correlation are the fundamental neighbourhood operations of image processing. They are linear operations. The computation performed in convolution or correlation has two main applications. One is the filtering of images – e.g. to suppress noise or enhance edges to be explained later. In this case, it is normal to describe the calculations done at each pixel as convolution. The other application is in measuring the similarity of two images. This is useful in feature recognition and in registration,

where we wish to place one image relative to another at a position of maximum similarity. In these applications, we use the term correlation to describe the calculations.

3.1.1 Calculating a Convolution

In convolution, the calculation performed at a pixel is a weighted sum of gray levels from a neighbourhood surrounding a pixel. The neighbourhood includes the pixel under consideration, and it is customary for it to be disposed symmetrically about that pixel. If a neighbourhood is centered on a pixel, then it must have odd dimensions, e.g. 3×3 , 5×5 , etc. The neighbourhood need not be square, but this is usually the case- as there is rarely any reason to bias the calculations in the x or y direction. Gray levels taken from the neighbourhood are weighted by coefficients that come from a matrix or convolution kernel. During convolution, we take each kernel coefficient in turn and multiply it by a value from the neighbourhood of the image lying under the kernel. We apply the kernel to the image in such a way that the value at the top-left corner of the kernel is multiplied by the value at the bottom right corner of the neighbourhood. Let us denote a 3×3 kernel by h and the image by f , then the entire calculation is

$$\begin{aligned} g(x, y) = & h(-1, -1) * f(x+1, y+1) + \\ & h(0, -1) * f(x, y+1) + \\ & h(1, -1) * f(x-1, y+1) + \\ & h(-1, 0) * f(x+1, y) + \\ & h(0, 0) * f(x, y) + \\ & h(1, 0) * f(x-1, y) + \\ & h(-1, 1) * f(x+1, y-1) + \\ & h(0, 1) * f(x, y-1) + \\ & h(1, 1) * f(x-1, y-1) \end{aligned}$$

This summation can be expressed more concisely as

$$G(x, y) = \sum_{k=-1}^1 \sum_{j=-1}^1 h(j, k) \times f(x-j, y-k)$$

For Example:

-1	0	1
-2	0	2
-1	0	1

Fig: A 3*3 Convolution kernel

	47	51	69	
	73	35	54	
	87	66	58	

Fig: The image neighbourhood

For the kernel and neighbourhood shown above, the result of convolution is

$$g(x, y) = (-1 \times 58) + (1 \times 87) + (-2 \times 54) + (2 \times 73) + (-1 \times 69) + (1 \times 47) = 45$$

Hence, the generalized equation of Convolution is:

$$g(x, y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j, k) \times f(x - j, y - k)$$

where, n_2 and m_2 are halves of height n and width m of kernel respectively. Where n and m are both odd, and hence $n_2 = \text{floor}(n/2)$ and $m_2 = \text{floor}(m/2)$. [floor function of $n/2$ is the greatest integer value that is less than or equal to $n/2$.]

3.1.2 Calculating a correlation

A Correlation is computed in almost exactly the same way as a convolution. The computation can be expressed as

$$g(x, y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j, k) \times f(x + j, y + k)$$

where n_2 and m_2 are defined as before. This differs from the previous one only in that kernel indices j, k are added to, rather than subtracted from, pixel coordinates x and y . This has the effect of pairing each kernel coefficient with the image pixel that lies directly beneath it. Correlation is often used in applications where it is necessary to measure the similarity between images or parts of images. For instance, we might need to locate a particular feature in an image. This can be done if we create a small image which acts as a model or template for that feature. In such application the

kernel h is replaced by the template image. Note that the above equation implicitly gives higher values for correlation in brighter parts of an image, which can make it difficult to identify the point of maximum similarity. It is therefore customary to normalize $g(x, y)$. One way of doing this is to divide by the sum of gray levels in the image neighbourhood, i.e.

$$g'(x, y) = \frac{g(x, y)}{\sum_k \sum_j f(x + j, y + k)}$$

Correlation works well only if we know the size and orientation of the feature of interest, and can design an appropriate template. If the size and orientation of the feature can vary, we will need to generate a range of templates and correlate each with the image, at great computational cost.

3. 2. Noise in an Image

Attenuating, or ideally suppressing image noise is important because any computer vision system begins by processing intensity values. Thus removal of such noise is of importance in stereo vision as well.

In computer vision, *noise* may refer to any entity, in images, data or intermediate results that is not interesting for the purpose of the main computation. The effect of noise is, essentially, that image values are not those expected, as these are corrupted during the various stages of image acquisition. As a consequence, the pixel values of two images of the same scene taken by the same camera and in the same light conditions are never exactly the same. Such fluctuations will introduce errors in the results of calculations based on pixel values; it is therefore important to estimate the magnitude of the noise.

The amount of noise in an image can be estimated by means of σ_n , the standard deviation of the random signal $n(i, j)$. It is important to know how strong is the noise

with respect to the interesting signal. This is specified by the signal-to-noise ratio, or SNR:

$$SNR = \frac{\sigma_s}{\sigma_n}$$

where σ_s is the standard deviation of the signal (the pixel values $I(i, j)$). The SNR is often expressed in decibel:

$$SNR_{db} = 10 \log_{10} \frac{\sigma_s}{\sigma_n}$$

We assume that the main image noise is *additive* and *random*; that is spurious, random signal, $n(i, j)$, added to the true pixel values $I(i, j)$:

$$I'(i, j) = I(i, j) + n(i, j)$$

Noise is categorized according to the amount and way it is present in an image, here we will give introduction of Gaussian noise and Impulsive noise.

The *Gaussian noise* model is often a convenient approximation dictated by ignorance: i.e. if we do not know and cannot estimate the noise characteristics, we take it to be Gaussian. In the absence of information, one often assumes $n(i, j)$ to be modeled by a white, Gaussian, zero-mean stochastic process. For each location (i, j) , this amounts to thinking of $n(i, j)$ as a random variable, distributed according to a zero mean Gaussian distribution function of fixed standard deviation, which is added to $I(i, j)$ and whose values are completely independent of each other and of the image in both space and time.

Impulsive noise, also known as spot or peak noise, occurs usually in addition to the one normally introduced by acquisition. Impulsive noise alters random pixels, making their values very different from the true values and very often from those of neighboring pixels too. Impulsive noise appears in the image as a sprinkle of dark and light spots. It can be caused by transmission errors, faulty elements in the CCD array,

or external noise corrupting the analog-to-digital conversion. *Salt-and-pepper* noise is a model adopted frequently to simulate impulsive noise in synthetic images.

3.2.1 Noise Filtering

Given an image I corrupted by noise n , attenuate n as much as possible (ideally, eliminate it). Attenuating or, if possible, suppressing image noise is important as the result of most computations on pixel values might be distorted by noise. An important example is computing image derivatives, which is the basis of many algorithms: any noise in the signal can result in serious errors in the derivatives. A common technique for noise smoothing is *linear filtering*, which consists in convolving the image with a constant matrix, call *mask* or *kernel*.

A linear filter replaces the value $I(i, j)$ with a weighted sum of I values in a neighborhood of (i, j) ; the weights are the entries of the kernel. The effects of a linear filter on a signal can be better appreciated in the frequency domain. Through the *convolution theorem*, the Fourier transform of the convolution of I and A is simply the product of their Fourier transforms $F(I)$ and $F(A)$. Therefore, the result of convolving a signal with A is to attenuate (or suppress) the signal frequencies corresponding to low (or zero) values of $|F(A)|$, spectrum of the filter A . If all entries A are non-negative, the filters perform *average smoothing*.

The main problems of averaging filter are: blur, poor feature localization, secondary lobes in the frequency domain, and incomplete suppression of peak noise. Gaussian filter (which is a particular case of averaging, in which the kernel is a 2-D Gaussian) solve the third one, as the Fourier transform of a Gaussian has no secondary lobes. The first two problems are tackled efficiently by *nonlinear filtering*; that is filtering methods that cannot be modeled by convolution.

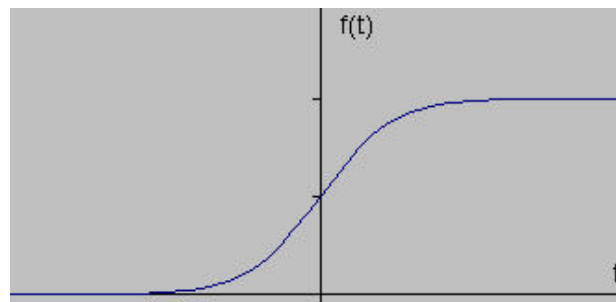
The *median filter* is a useful representative of this class. A median filter just replaces each pixel value $I(i, j)$ with the median of the values found in a local neighborhood of (i, j) . As with averaging, the larger the neighborhood, the smoother the result.

3. 3. *Edge Detection*

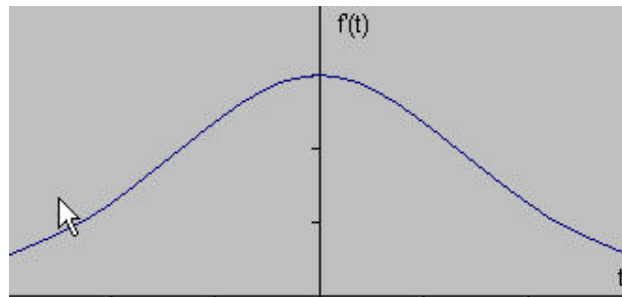
Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories, gradient and Laplacian.

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image whereas the Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

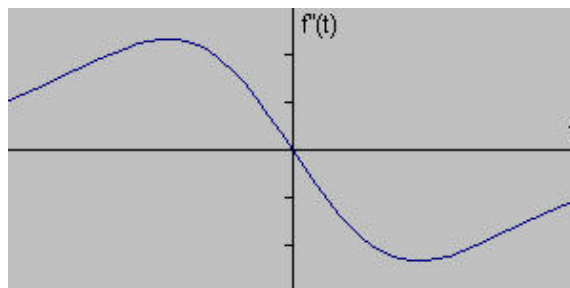
Suppose we have the following signal, with an edge shown by the jump in intensity below:



If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to t) we get the following:



Clearly, the derivative shows a maximum located at the center of the edge in the original signal. This method of locating an edge is characteristic of the “gradient filter” family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, you can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is shown below:



Based on this one-dimensional analysis, the theory can be carried over to two-dimensions as long as there is an accurate approximation to calculate the derivative of a two-dimensional image. The Sobel operator performs a 2-D spatial gradient

measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The Sobel edge detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The actual Sobel masks are shown below:

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

The magnitude of the gradient is then calculated using the formula:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

An approximate magnitude can be calculated using

$$|G| = |G_x| + |G_y|$$

The 5x5 Laplacian used is a convoluted mask to approximate the second derivative, unlike the Sobel method which approximates the gradient. And instead of 2 3x3 Sobel masks, one for the x and y direction, Laplace uses 1 5x5 mask for the 2nd derivative in both the x and y directions. However, because these masks are approximating a second derivative measurement on the image, they are very sensitive to noise. The Laplacian mask is shown below

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

In general the edge detection can be summarized in three steps as

- i. *Noise Smoothing*: Suppress as much of the image noise as possible, without destroying the true edges. In the absence of specific information, assume the noise white and Gaussian.
- ii. *Edge Enhancement*: Design a filter responding to edges; that is, the filter's output is large at edge pixels and low elsewhere, so that edges can be located as the local maxima in the filter's output.
- iii. *Edge Localization*: Decide which local maxima in the filter's output are edges and which are just caused by noise. This involves:
 - a. Thinning wide edges to 1-pixel (non-maximum suppression):
 - b. Establishing the minimum value to declare a local maxima of an edge (thresholding).

3.3.1 Canny Edge Detection

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. A list of criteria to improve current methods of edge detection was included in his theory. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single

edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T_2 .

Chapter 4
Stereo Vision

4. 1. Introduction

The word *Stereo* came from the Greek word *stereos* which means firm or solid. With Stereo Vision we see an object as solid in three spatial dimensions: width, height and depth or x, y and z. It is the added perception of the depth dimension that makes Stereo Vision so rich and special.

Stereo Vision or stereoscopic vision probably evolved as a means of survival. With Stereo Vision, we can see where objects are in relation to our own bodies with much greater precision, especially when those objects are moving toward or away from us in the depth dimension. We can see a little bit around solid objects without moving our heads and we can even perceive and measure "empty" space with our eyes and brains. Some occupations that depend heavily on Stereo Vision are Cricket player, Waiter, Driver, Architect, Surgeon, Dentist and many more. We can identify how the use of stereo vision to perceive the depth are unknowingly being done everyday by us, here are just a few examples of general actions that depend on Stereo Vision:

- Throwing, catching or hitting a ball.
- Driving and parking a car.
- Planning and building a three-dimensional object.
- Threading a needle and sewing.
- Reaching out to shake someone's hand.
- Pouring into a container.
- Stepping off a step.

The different perspectives of our two eyes lead to slight relative displacements of objects (disparities) in the two monocular views of scene. The human visual system is able to

- Use these disparities for depth-estimation.

- Merge both monocular views into a fused cyclopean view of the scene.

Thus, the term Stereo Vision refers to the ability to infer information on the 3D Structures and the distances of a scene from at least two images (Left and right), taken from different viewpoints. Stereo Vision utilizes the slightly different views of a scene, projected on to the right and left images, to recover depth information, which is known as Binocular Stereo Fusion. From a generic point of view, in both humans and machines, the problem reduces to a matching of the two views, in order to find the displacement (disparity) of corresponding patterns of the projected images. Hence, a Stereo system must solve two essential problems – correspondence and reconstruction which is explained later.

4. 2. Application

Stereo Vision offers a wide range of application. Some of them are already known to be used in practice. A few of them are listed below with an identification of how stereo vision can be implemented to those fields.

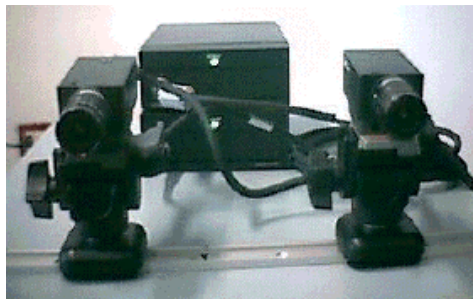


fig. 4.1: Stereo Vision System used for performance Analysis. [14]

4.2.1 Autonomous Robots & Vehicles

3D object detection, location and depth perception calculated from the stereoscopic view of objects by the pair of cameras located on the robotic body can be used for the mobilization of the robot or robotic vehicle. The calculated results from the

stereoscopic views can be used to detect and avoid the obstacles and take a safe path of action. One of such project making use of Stereo Vision is PRASSI project [14]. The figure shown in *fig 4.1* is the system that was used for the test in analyzing the performance of it.

4.2.2 Industrial Inspection and Quality Control

Industrial inspection and quality control done manually does not always guarantee correctness. In this condition, where quality control is of crucial matter, stereo vision system can be thought of as a good choice of machine. Stereoscopic vision can be used to detect the cracks, damage, unusuality in goods, raw materials or products, during manufacturing process.

4.2.3 Road monitoring and traffic light system

Road monitoring and traffic light management can be really dynamic rather than the existing static approach. The static approach here refers to the one in country like Nepal and other several countries, where traffic light changes its state in fixed interval of time, thereby making the vehicle stand and wait even though there are no vehicles on the other side. The frequency and duration of traffic light alteration can be dynamically determined by the amount of load on either side of street using Stereo Vision.

4.2.4 Military Application

With an intension to target with precision heavy loss in enemy & few losses in offending side is key to military actions. Use of stereoscopic views to precisely calculated depth, detected & located enemy targets, autopilot or remote controlled vehicles, air-ships can have wide use of Stereo Vision. In some job of military, which includes greater risk of life such as detonation of bomb is still done manually. This

can be improved by making use of remotely operated or autonomous vehicle which is able to precisely pick particular point of 3D world.

4.2.5 Bio-Medical Engineering

The Stereo Fusion of stereoscopic views can help explain the psychophysical and neurophysiological data. In the field like hair transplant, the operation procedure requires great care and accuracy. There is a widely used product “Mantis Stereo viewing system” produced by vision engineering [23] is used for this purpose. Another product of same group is “Lynx Eyepieceless Stereo Microscope” which is providing highly sophisticated product to make angioplasty heart surgery a safer and less complicated procedure.

Chapter 5
Stereo Algorithm

The primary objective of Stereo Algorithm is of course to estimate depth from a pair of 2-Dimensional images. The need of a pair of 2D image is to figure out the displacement of feature between the images, when camera is displaced (usually horizontally). A technique to calculate such disparity is known as Stereo Correspondence. Several approaches have been used by different people to find out corresponding match and its displacement. With the use of this displacement value, which is also called as Disparity, we can recover depth of an object using algorithm explained in section 5.4.

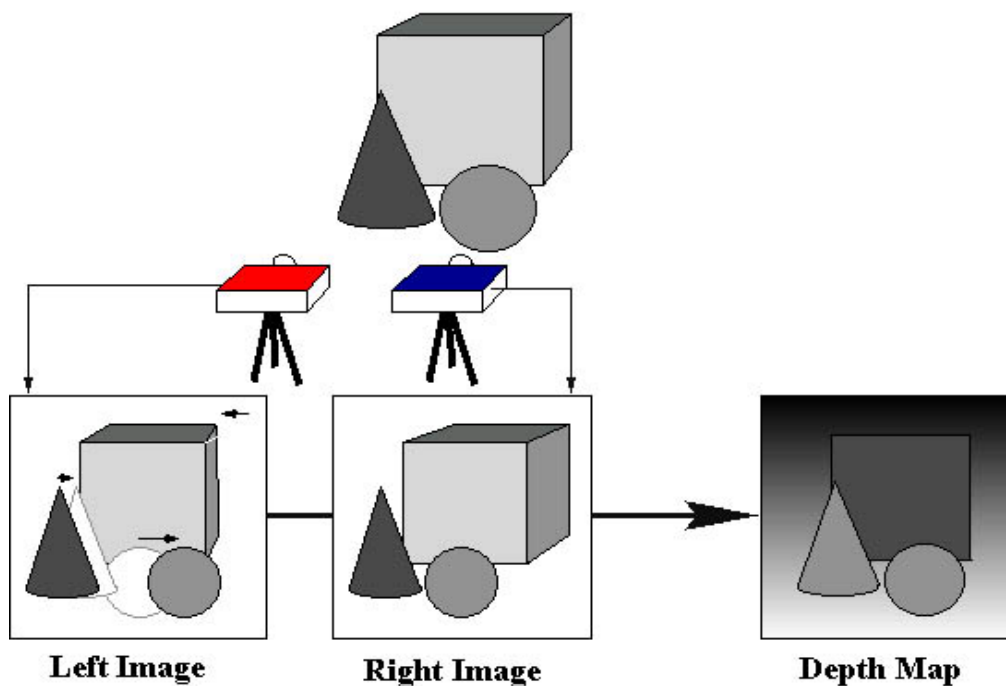


fig 5.1.: A Stereo System capturing the image from two horizontally aligned camera that gives the depth map as an output .

5.1. Stereo Correspondence

According to Marr and Poggio [8], three steps (S) are required to solve the problem of measuring Stereo Disparity.

[S1.] A particular location on a surface in the scene must be selected from one image;

[S2.] That same location must be identified in the other image; and

[S3.] The disparity in the two corresponding image points must be measured.

If one could identify a location beyond doubt in the two images, for example by illuminating it with a spot of light, steps S1 and S2 could be avoided and the problem would be easy. In practice one cannot do this (*see fig 5.2.*), and the difficult part of the computation is solving the correspondence problem. In order to formulate the correspondence computation precisely, they tried to compare its basis in the physical world and identified two constraints(C), they are

[C1.] A given point on a physical surface has a unique position in space at any one time; &

[C2.] Matter is cohesive, it is separated into objects, and the surfaces of objects are generally smooth compared with their distance from the viewer.

These constraints apply to locations on a physical surface. These physical constraints C1 and C2 can now be translated into two rules(R) for how the left and right descriptions are combined:

[R1.] Uniqueness. Each item from each image may be assigned at most one disparity value. This condition relies on the assumption that an item corresponds to something that has a physical position.

[R2.] Continuity. Disparity varies smoothly almost everywhere. This condition is a consequence of the cohesiveness of matter, and it states that only a small

fraction of the area of an image is composed of boundaries that are discontinuous in depth.

In practice, R1 cannot be applied simply to gray level points in an image, because, a gray level point is in only implicit correspondence with a physical location. It is in fact impossible to ensure that a gray level point in one image corresponds to exactly the same physical position as a gray-level point in the other. A sharp change in intensity, however, usually corresponds to a surface marking, and therefore defines a single physical position precisely. The positions of such changes may be detected by finding peaks in the first derivative of intensity, or zero-crossings in the second derivative.

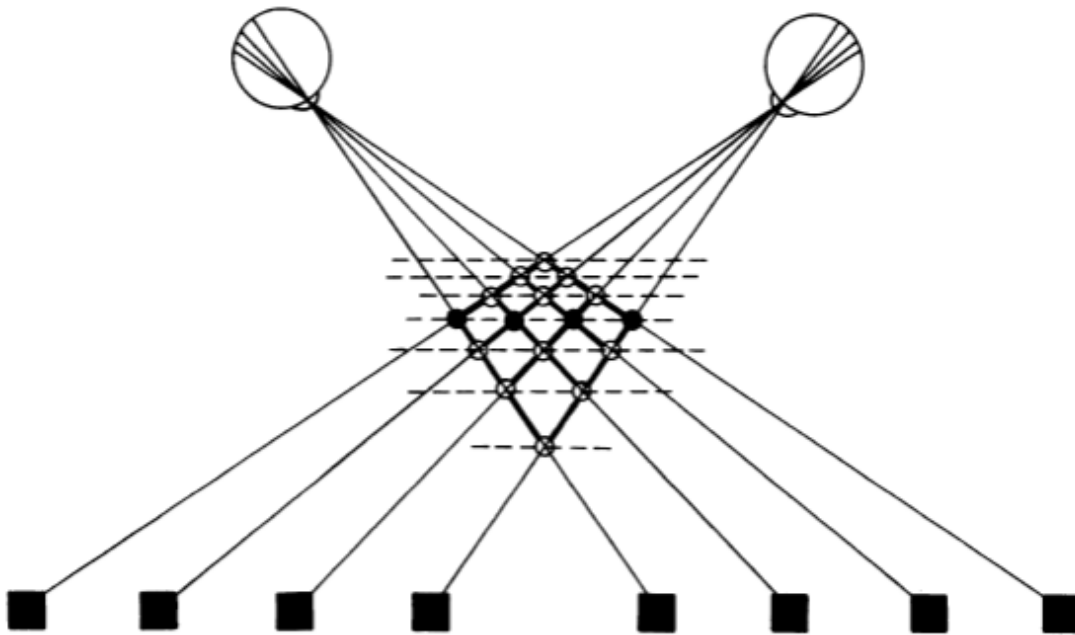


fig 5.2.: Ambiguity in the correspondence between the two retinal projections. In this figure, each of the four points in one eye's view could match any of the four projections in the other eye's view. Of the 16 possible matching only four are correct (filled circles), while the remaining 12 are "false targets" (open circles). It is assumed here that the targets (filled squares) corresponds to "matchable" descriptive elements obtained from the left and right images. Without further constraints based on global considerations, such ambiguities cannot be resolved. [9]

5.2. Constraints

The inherent ambiguity of the correspondence problem can also in practical cases be reduced using some other constraints that Marr and Poggio didn't include. These constraints have been used by some other stereo correspondence algorithms proposed. It is not necessary that all of these constraints have to be used; different approaches use different constraints. These constraints may broadly be classified as two types:

5.2.1. Constraints due to geometry & photometry of image capturing process.

- **Epipolar constraint:** This says that the corresponding point can only lie on the epipolar line in the second image. This reduces the potential 2D search space into 1D.
- **Photometric compatibility constraint:** This states that intensities of a point in the first and second images are likely to differ only a little. They are unlikely to be exactly the same due to the mutual angle between the light source, surface normal, and viewer differing, but the difference will typically be small and the views will not differ much. Practically, this constraint is very natural to image-capturing conditions. The advantage is that intensities in the left image can be transformed into intensities in the right image using very simple transformations.
- **Geometric similarity constraints:** These build on the observation that geometric characteristics of the features found in the first and second images do not differ much (e.g., length or orientation of the line segment, region, or contour).

5.2.2. Constraints exploiting some common properties of objects

- **Figural disparity constraint:** This says that corresponding points should lie on an edge element in both right and left images, as well as fulfilling the continuity constraint.
- **Feature compatibility constraint:** This places a restriction on possible matches on the physical origin of matched points. Points can match only if they have the same physical origin – for example, object surface discontinuity, border of a shadow cast by some objects, occluding boundary or specular boundary. Notice that edges in an image caused by specular or self-occlusion cannot be used to solve the correspondence problem, as they move with changing viewpoint. On the other hand, self-occlusion caused by abrupt discontinuity of the surface can be identified (*see fig 5.3*).

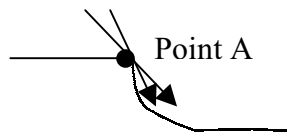


fig 5.3.: Self-occlusion due to abrupt surface discontinuity can be detected. [15].

- **Disparity Limit Constraint:** This originates from psycho-physical experiments in which it is demonstrated that the human vision system can only fuse stereo images if the disparity is smaller than some limit. This constraints the lengths of the search in artificial methods that seek correspondence.
- **Mutual correspondence constraint:** This helps to rule out points that do not have a corresponding counterpart due to occlusion, highlight or noise. Assume the search started from the left image point P_L and a corresponding P_R was found. If the task is reversed, and a search starting from the point P_R fails to find P_L , then the match is not reliable and should be ruled out.

5.3. Approach

Till now different researchers have already found out the different approach of Calculating Stereo Disparity. According to our research, these approaches have been identified according to two broad divisions, viz. Area Based and Feature Based.

5.3.1. Area Based Approach

Area-based stereo attempts to determine the correspondence for every pixel, which results in a dense depth map. Correlation is the basic method to find corresponding pixels.

Given any two views of the same scene it can be seen that, at some image scale, a degree of similarity exists between the two views, and in general, the coarser the scale the more similar the views become. There are two primary reasons for this; the first is that a fixed disparity of position of fixed point in two views of the same scene (determined by camera and world geometry) has a proportionately smaller effect at large image scales, the second is that at coarser scales there is a reduction in the quantity of visible features such that only more dominant features exist. These effects form the basis for cross-correlation area based stereo algorithms by the explanation which now follows. If a view is specially quantised into ever smaller sub regions, eventually any given sub region will begin to look more similar to its corresponding sub region in the other view. Thus, by quantising a view into a number of sub regions or blocks, or by changing the scale of the view in question, it is possible to apply an area based similarity metric to find the most likely correspondence between the same regions from two different views.

The use of an appropriate similarity metric is fundamental to area based stereo methods as a means of enforcing local figural consistency and a surface smoothness assumption. The concept of a cross-correlation function and a search space is also introduced with the use of area-based methods. Theoretically similarity metrics

derived from probability density functions offer the best solution although, in practice, approximations such as Euclidean distance and dot product metrics have been used because of the simplicity of implementation.

In their most simplistic form area based approaches involve subdividing the whole view into sub regions and applying a photometric similarity measure to all regions. The aim of this type of algorithm is usually to return a dense depth map, whereby a depth estimate is made at every pixel within the scene. This approach is generally only applicable to the class of stereo problems where the following criteria are satisfied:

- The lighting source must ideally be a point source at infinity;
- The surfaces in the scene should ideally be lambertian;
- The amount of figural dissimilarity or distortion between the views is small.

In many situations where stereo has applications idealized environments and light sources cannot be assumed. Simplistic gray-level correlation techniques suffer from a lack of robustness to lightning artifacts, which exist in many situations.

Area-based stereo attempts to determine the correspondence for every pixel, which results in a dense depth map. Correlation is the basic method to find corresponding pixels. Several real time systems have been developed using correlation-based stereo. However, correlation assumes that the depth is equal for all pixels of a correlation window. This assumption is violated at depth discontinuities. The result is that object borders are blurred and small details or objects are removed, depending on the size of the correlation window. Small correlation windows reduce the problem, but increase the influence of noise and lead to decrease of correct matches.

Simple correlation exhibits a systematic error, i.e. blurring of object borders. However, the assumed location of a computed depth discontinuity is still near (i.e. within the size of the correlation window) to the location of the real depth

discontinuity. Furthermore, correlation has proven to be fast enough for a real time implementation and has a regular structure with fixed execution time, which is independent of the scene contents.

The problem with using a single depth estimate to describe the depth over a finite sub region of the image is that this quantisation introduces location errors. Area based approaches which have attempted to address this problem are often referred to as window shaping techniques. Whilst the field of application for these techniques is still for estimating dense depth maps, the location accuracy, which can be achieved, is now not limited by block quantisation effects. However, the reliance on luminance consistency is, if anything, more important to the success of this class of relaxation algorithms.

In general, existing area based stereo techniques provide data, which is locally inaccurate due to the lack of figural deformation invariance. Additionally, their lack of photometric invariance restricts their use to problems where gray-level consistency exists between views, however, if these constraints are satisfied they do deliver a more dense depth map than feature based approaches.

5.3.2. Feature Based Approach

Algorithms which perform stereo matching with high-level parameterizations called image features.

Many areas of computer vision, such as stereo, object recognition and object tracking explicit a feature-based approach. The definition of a feature is arbitrary and the only real generalization, which can be made, is that a feature must be in some sense a useful (reliably reconstructable) parameterization of the image. In general useful features must have the following properties: uniqueness, repeatability and physical meaning. In the context of stereo the aim of these properties is to provide unambiguous matches with a degree of noise immunity. Since stereo vision involves

extracting three-dimensional data (3D data) from the scene, the features, which are useful in the stereo sense, are features, which describe the underlying 3D structure of the scene. In the main, and particularly in manmade environments, the underlying 3D structure is described by the edges and by edge intersections (one definition of a corner). For this reason much of the feature based stereo work has firstly involved the extraction of high-level edge primitives (edgels) and in some cases corners. In some cases the extracted edgels, which have been obtained using something similar to Canny, are linked into high-level data objects called edge-strings, stereo matching would then proceed at the edge-string level.

In many respects feature based algorithms have been established as the most robust way to implement stereo vision algorithms for the class of problems. The advantages offered by using features are that feature based representations contain desirable statistical properties and provide algorithmic flexibility to the programmer. The flexibility being that algorithmic constraints can be applied explicitly to the data structures rather than implicitly as with area based correlation techniques. In particular the use of edge-string based representations has led to algorithms, which are as locally accurate as the precision to which the edges can be extracted.

Summarizing, in comparison to area based stereo algorithms, which attempt to provide dense depth data, edge-string based stereo algorithms provide sparser depth data which is locally more accurate and globally more reliable for the following reasons:

- Edge-string based algorithms do not use a region based planarity model to describe the world, they instead model the world as consisting a linked edges, a more useful model for “difficult” stereo problems where the world model must incorporate some figural deformation invariance.

- Edge-string based algorithms exploit the properties of edge based data, which can be extracted reliably from the scene in a way that is immune to noise and invariant to luminance variations between views caused by non-ideal lighting.

Area based approaches do have some advantages when considering hardware platforms however, as the regular data structure simplifies considerably the control and data flow within the algorithm. This has recently led to an attempt to combine the constraints embodied in edge-based techniques with an area-based formulation.

5.4. Depth Recovery

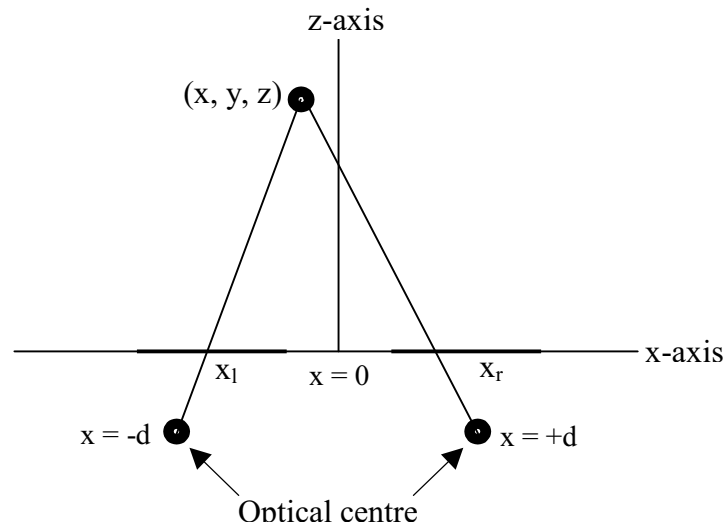


fig 5.4. The Stereo images are formed at $(x+d)$ and $(x-d)$ locations along x-axis.[2]

Refer to the fig 5.4. and assume that the correspondence is established between the stereo image pair. Let the disparity between two cameras are $2d$. Disparity is the distance of separation between the left and the right camera. In the above figure, two separate images of the point $p(x, y, z)$ are formed at $(x + d)$ and $(x - d)$ locations along x-axis.

If f is the focal length of the pinhole camera system and x_l and x_r are the left and right camera images respectively, then

$$x_l = \frac{(x-d)f}{f-Z} \text{ and } x_r = \frac{(x+d)f}{f-Z}$$

or,

$$x_r - x_l = \frac{2df}{f-Z}$$

Which gives,

$$Z = f - \frac{2df}{x_r - x_l}$$

Therefore, for a given binocular camera model with known focal length and disparity of camera pair, the depth of the object could be recovered once the correspondence between the left and right eye images are solved.

5.5. Issues And Challenges

Some of the important issues that can be raised during the study and implementation of Stereo Vision are discussed below.

- Digital Image Processing and use of complex mathematical formulae in the algorithms requires great deal of computation, thus consuming more time. Use of hardware with high capacity is recommended for better performance.
- Identification of suitable algorithms to implement is a challenging job, where issues of performance, correctness, and restrictions have to be considered.
- If two (or more) images are available, then the three-dimensional point can be obtained as the mapping of the corresponding points of the two images. A number of things are needed for this
 - Corresponding image points
 - Relative pose of the camera for the different views
 - Relation between the image points and the corresponding line of sight

Note, however that the calculation of disparity is not an easy task to attain.

- The correct and fast estimation of disparities is a difficult problem. Besides disparities, various additional image variations occur between the left and right view of a scene. Differences might be caused by occlusions of objects, specular reflections, which move independently of the surfaces of objects, sensor noise, and various other causes.
- The relation between an image point and its line of sight is given by the camera model (e.g. pinhole camera) and the calibration parameters. These parameters are often called the intrinsic camera parameters while the position and orientation of the camera are in general called extrinsic parameters. How can all these elements can be retrieved from the images. The key for this are the relations between multiple views which tell us that corresponding sets of points must contain some structure and that this structure is related to the poses and the calibration of the camera. [11]
- Another type of problems is caused when the imaging process does not satisfy the camera model that is used. Some times radial distortion is present in the image. This means that the assumption of a pinhole camera is not satisfied. It is however possible to extend the model to take the distortion into account. However, sometimes image is much harder to use when important part of the scene is not in focus. And the problem becomes more prominent when blooming is present (i.e. overflow of CCD-pixel to the whole column). Most of these problems can however be avoided under normal imaging circumstance.
- Implementation of stereo system in the real time system must consider different affecting factors like motion of the scene, varying light intensity, appearance changes of models, complex natural objects, presence of noise (unwanted feature that hinders the real one) etc.
- Note that different viewpoints are not the only depth cues that are available in images. Shading, Shadows, Symmetry, Texture and focus also give some hints about depth or local geometry, but considering those cues add an extra overhead.

Chapter 6
Project Description

The result of this project after one year of research was a system built on java platform that implemented one of the area based stereo algorithm that make use of correlation based approach to find out the disparity map by taking an input of the set of stereo images. For the implementation part, a cooperative algorithm proposed by Zitnic and Kanade [4.] was used.

6.1. Theory of a Cooperative Algorithm

Zitnic and Kanade[4.] have presented a stereo algorithm for obtaining disparity maps with occlusion explicitly detected. To produce smooth and detailed disparity maps, two assumptions that were originally proposed by Marr and Poggio has been adopted: uniqueness and continuity. These assumptions are enforced within a three-dimensional array of match values in disparity space. Each match value corresponds to a pixel in an image and a disparity relative to another image. An iterative algorithm updates the match values by diffusing support among neighboring values and inhibiting others along similar lines of sight. By applying the uniqueness assumption, occluded regions can be explicitly identified.

The Cooperative Algorithm is summarized as:

- Prepare a 3D array, (r, c, d) : (r, c) for each pixel in the reference image and d for the range of disparity.
- Set initial match values L_0 using a function of image intensities, such as normalized correlation or squared differences.
- Iteratively update match values L_n using equation given below, until the match values converge.

Let $\Psi(r, c, d)$ denote the set of elements, which overlap element (r, c, d) when projected to an image. That is, each element in $\Psi(r, c, d)$ projects to pixel (r, c) in the left image or to the pixel $(r, c+d)$ in the right image. With the uniqueness assumption, $\Psi(r, c, d)$ represents the inhibition area to a match at (r, c, d) .

$$L_{n+1}(r, c, d) = L_0(r, c, d) * \left(\frac{S_n(r, c, d)}{\sum_{(r'', c'', d'') \in \Psi(r, c, d)} S_n(r'', c'', d'')} \right)^\alpha$$

The exponent α controls the amount of inhibition per iteration. To guarantee a single element within $\Psi(r, c, d)$ will converge to 1, α must be greater than 1. Here $L_0(r, c, d)$ is computed as

$$L_0(r, c, d) = \delta(I_{left}(r, c), I_{right}(r, c + d))$$

Where, δ is an image similarity function such as squared differences or normalized correlation.

And, $S_n(r, c, d)$ is the amount of local support for (r, c, d) , i.e. the sum of all match values within a 3D local support area.

- For each pixel (r, c) , find the element (r, c, d) with the maximum match value. If the maximum match value is higher than a threshold, output the disparity d , otherwise classify it as occluded.

6.2. Implementation Details

All the implementations of this algorithm are packaged under the package *edu.ku.stereo*. Methods for calculating the L_0 values and L_n values are the member of Stereo class and supporting methods to manipulate the pixel data are stored in image3d Class.

The class image3d make use of another private class pixData. The necessity of this private class become prominent when image3d class wants to have concurrent reference to more than one pixel value at a time; that is during summation of pixel values around the window of certain radius. Thus adding an instance of private class pixData within the image3d makes it simple, but before that all the modification in pixel value of class image3d has to subsequently modify the pixel data of an instance

of `pixData`. Till the `image3d` class, the pixel values are assumed to be stored and extracted by indexing as a multidimensional array, but when it comes to the `pixData` class, pixel values are stored in single dimensional manner.

Calculating L_0 & L_n values is just a matter of running the loop across rows, columns and depth level. Use of these loops is directly mapped from each summation (Σ) notation found in the equation described in section 6.1. The disparity values of each pixel always lie within the depth interval provided and it is better idea to scale it within a specified range (here we have taken 8 bit, i.e. from 0-255) and display. The confidence value is the vote of each pixel with its corresponding pixel taking into consideration to its neighboring pixels and these values are generally very very less than 1 after computation. Hence, these values are also to be scaled up to lie within the grayscale level and a confidence value less than threshold value (here 35 is assumed to be threshold, but that may vary according to the properties of image) are assumed to be occluded. So, the occlusion map only needs to be displayed in a binary format, i.e. the points that are occluded are expressed by a black dot and those non-occluded points as white.

Since, java is the platform for implementation of this algorithm, most of the things necessary for GUI development was already there. So, designing of GUI was to inherit those classes and addition of extra functionality of our particular application to the classes where necessary. `stereoDesktop` is the topmost class that is the parent of our application, which is instantiated directly from our main method. Since there is only one instance `stereoDesktop` at a time, there is static instance of the class itself. So the factory method of `stereoDesktop` does not create new object when requested, but returns the reference to an old object. Thus from any other class, simply importing the `stereoDesktop` class grants the use of already built desktop. The access of this instance is useful in cases while displaying the message box, dialog box or progress monitor that always asks for the parent class over which to display. Fig 6.1. shows an instance of desktop frame being displayed while it is processing.

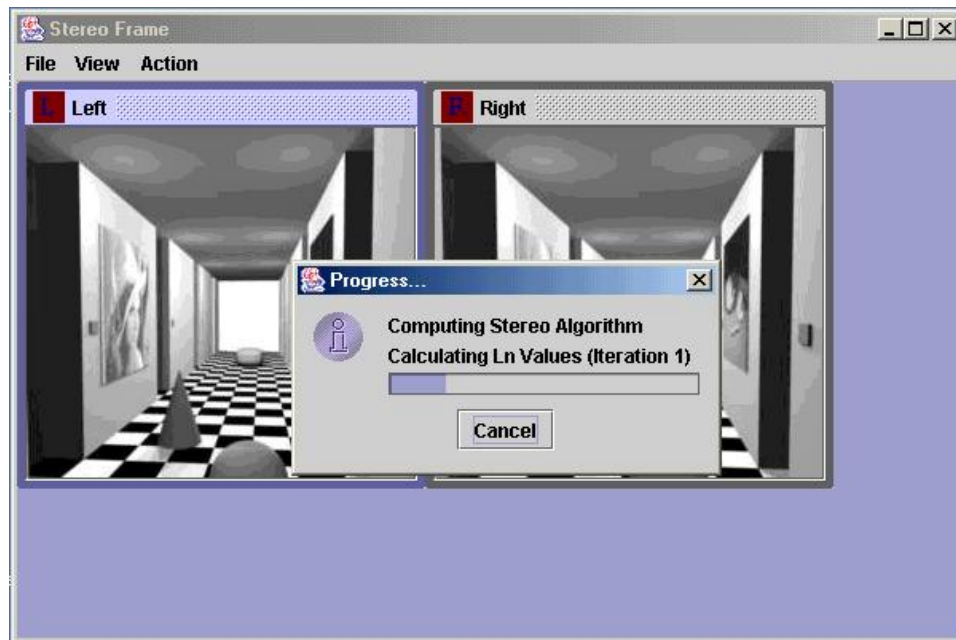


fig 6.1.: Stereo Desktop Frame

All the internal frames of the stereo desktop have to show images in one way or other. Thus an abstract class `imagePanel` contains few implementations like `getImage()` and `setImage()` that returns or sets the image in that frame. But in lower level, the sub-class of image panel which are `leftPanel`, `resultPanel`, & `rightPanel` have their own functionality such as enabling or disabling of frame properties, actions to perform on mouse event, the need of menu system and its contents. Thus the class `leftPanel` is made such that it works on mouse event such as marking of corresponding point in right image of the selected point and `resultPanel` gives its own menu for the user to save the output image.

If there is only few data that is required for the operation of any application, the data are sent as a part of argument. But if there are more than two or three arguments, rather than giving a list of arguments such as name of left image, name of right image, number of iteration to make, radius of window, etc. we decided to put all the arguments in a binary file and give a single argument; the name of that binary file. Thus after reading the given file, all the required arguments can be extracted out. This

extraction is made possible by the use of a parser, which will assume that all the data are arranged in a specified order and complete. So, after parsing the data it returns the data whenever required by other classes.

When it comes to implementation part, it is always unknown that when to stop. And this is easy for us; “Stop the work when the end of semester approaches with the minimum goal accomplished”. And with this much of implementation we are almost near to the end of our semester and have fulfilled what was decided earlier. As most of the time for this project was devoted in learning, i.e. from the beginning of image processing to the strategies of stereo vision. We could have done more than what was implemented and the ratio of acquired knowledge to output is quite high. We believe that we can use this knowledge at higher level of our studies.

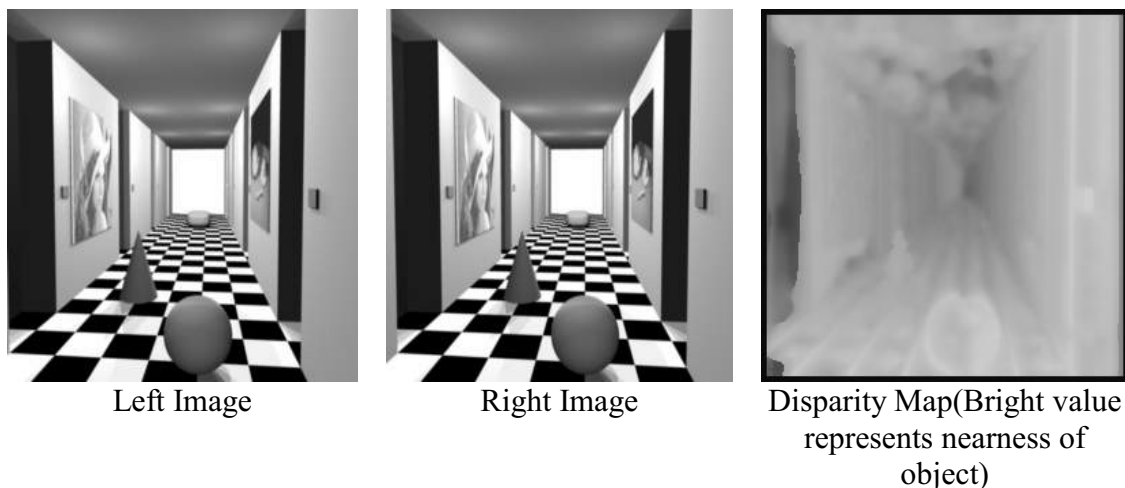


fig 6.2.: Stereo pair image of corridor and its depth map

Chapter 7

Conclusion And Further Recommendation

Researchers have been investigating methods to acquire 3D information from objects and scenes for many years. In the past the main applications were visual inspection and robot guidance. Nowadays however the emphasis is shifting. There is more and more demand for 3D models in computer graphics, virtual reality and communication. This results in a change in emphasis for the requirements. The visual quality becomes one of the main points of attention. Therefore not only the position of a small number of points have to be measured with high accuracy, but the geometry and appearance of all points of the surface have to be measured. This requires an extensive use of Stereo Vision.

Since, the project basically aimed for the member to be equipped with the knowledge of Computer Vision and particularly with the field of Stereo Vision. Hence, whatever targeted at the beginning of the project was acquired. In addition, the project members have worked in such a way that the differential in approaches of stereo vision proposed by different members have been summarized here in the report and implementation of one of such type is successfully done.

Further, this project may be used for the application in different areas listed in section 4.2 with a few modifications. Since, the implementation is done in object oriented manner, each class may be reused for the larger system that make use of Stereo Vision.

Appendices

Appendix A. Glossary

2-Dimensional a system considering only two factors, for example a picture having horizontal (x-coordinate) and vertical(y-coordinate) coordinate system.

CCD Camera Charged Coupled Device is a collection of tiny light-sensitive diodes, which convert photons (light) into electrons (electrical charge). The brighter the light that hits a single diode, the greater the electrical charge that will accumulate at that site.

Computer Vision a branch of Computer Science, which is capable of perceiving the information through the visual information; also known as image analysis, scene analysis, image understanding.

Cooperative Algorithm A cooperative approach using the disparity space to utilize the two assumptions: uniqueness and continuity of a disparity map.

Edge edge points, or simply edges, are pixels at or around which the image values undergo a sharp variation.

Epipole *see Epipolar Geometry.*

Epipolar Line *see Epipolar Geometry.*

Epipolar Geometry: Given a stereo pair of cameras, any point in 3-D space, P , defines a plane, π_P , going through P and the centers of projection of the two cameras. The plane π_P is called *epipolar plane*, and the lines where π_P intersects the image planes *conjugated epipolar lines*. The image in one camera of the projection center of the other is called epipole.

Ideal Environment a situation which does not contain extra factor that might affect in the operation of a system.

Intensity the quantity of light that can be discussed in the physics sense of energy.

Pinhole Camera The camera's aperture reduced to a point, called a *pinhole*, such that only one ray from any given point can enter the camera, and creates a one-to-one correspondence between visible points, rays, and image points.

Robot machine automatically completing a mechanical process.

Robotics art, science and study of robot design and operation.

Self Occlusion Points with no counterpart in the other image.

Sonar system for the detection of objects by reflected sound.

Stereo Matching Given an element in the left image, we search for the corresponding element in the right image.

Stereo Correspondence Consists in determining which item in left image corresponds to the item in right image.

Vision The act of perceiving and interpreting visual information.

Appendix B. **References & Bibliography**

- [1.] Binsan Khadka et al. (2003), “Stereo Vision: An Introductory Approach”, First National Students Conference on IT, Nepal College of Information Technology.
- [2.] B. Chanda, D. Dutta Majumder, “Digital Image Processing and Analysis”, PHI.
- [3.] B.G. Batchelor and P.F. Whelan (1984), “Machine vision systems: Proverbs, principles, prejudices and priorities”, Proceedings of the SPIE – The International Society for optical Engineering, vol. 2374 – Machine Vision Applications, Architectures, and Systems Integration III, Boston (USA), pp 374-383.
- [4.] C. Lawrence Zitnic, Takeo Kanade (1999) , “ A Cooperative Algorithm for Stereo Matching and Occlusion Detection”, CMU-RI-TR-99-35.
- [5.] Cay S. Horstmann, Gary Cornell, “Core Java Volume I-Fundamentals”, Pearson Education Asia.
- [6.] Cay S. Horstmann, Gary Cornell, “Core Java Volume II-Advanced Features”, Pearson Education Asia.
- [7.] D. Geiger et al. (1995), “Occlusions and Binocular Stereo,” International Journal of Computer Vision (IJCV), Vol. 14, pp. 211-226.
- [8.] D. Marr, T. Poggio(1976), “Cooperative Computation of Stereo Disparity”. Science, New Series, Volume 194, Issue 4262, pp 283-287.
- [9.] D. Marr, T. Poggio(1977), “A Theory of Human Stereo Vision”, Massachusetts Institute of Technology AI Lab. A. I. Memo No. 451.
- [10.] Donald Hearn, M. Pauline Baker, “Computer Graphics – C Version”, Pearson Education Asia.
- [11.] Forsyth and Ponce, “ Computer Vision -A Modern Approach”, Pearson Education. Chapter 1.
- [12.] Herbert Schildt, “The Complete Reference Java™ 2”, Tata McGraw Hill Edition.

- [13.] James D. Foley et al., “Computer Graphics Principles and Practice”, Addison Wesley.
- [14.] M. Bertozzi et al., “Stereo Vision System Performance Analysis”, Università di Parma 181-A I-43100, Parma, Italy.
- [15.] Milan Sonka et al., “Image Processing, Analysis, and Machine Vision”, PWS Publishing.
- [16.] Nick Efford, “Digital Image Processing- a practical introduction using Java™”, Pearson Education Asia.
- [17.] P. N. Belhumeur & D. A. Mumford, “A bayesian treatment of the stereo correspondence problem using half-occluded regions,” In Proc. IEEE Conf. On Computer Vision and Pattern Recognition, 1992.
- [18.] William Hoff & Narendra Ahuja, “Depth From Stereo”, University of Illinois, USA.
- [19.] William Hoff & Narendra Ahuja, “Extracting Surfaces from Stereo Images: An Integrated Approach”, Coordinated Science Laboratory, University of Illinois, USA.
- [20.] William Hoff & Narendra Ahuja (1989), “ Surfaces From Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection”, IEEE Transaction on pattern analysis and machine intelligence, Vol 11, No. 2, pp 121 –128
- [21.] <http://axon.physik.uni-bremen.de/research/stereo/principle.html>
- [22.] <http://www.vision3d.com/stereo.html>
- [23.] <http://www.visioneng.com/>