

3D OBJECT RECONSTRUCTION FROM MULTIPLE VIEWS:A COMPRESSIVE SENSING FRAMEWORK

D.S. Srikanth Reddy

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering

June 2012

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.



(Signature)

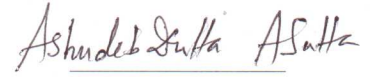
(D.S.Srikanth Reddy)

EE10M02

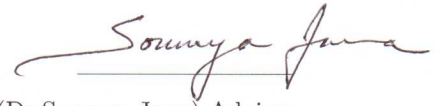
(Roll No.)

Approval Sheet

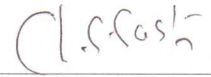
This Thesis entitled 3D OBJECT RECONSTRUCTION FROM MULTIPLE VIEWS: A COMPRESSIVE SENSING FRAMEWORK by D.S.Srikanth Reddy is approved for the degree of Master of Technology from IIT Hyderabad



(Dr.Asudeb Dutta) Examiner



(Dr.Soumya Jana) Adviser
Dept. of Electrical Engineering
IITH



(Dr.C.S.Sastry) Co-Adviser
Dept. of Mathematics
IITH

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor Dr. Soumya Jana for the continuous support during my M.Tech research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me a lot in pursuing my research and in writing this thesis.

I would also like to thank my co-adviser Dr.C.S.Sastry for his immense guidance and support throughout my project. I extend my sincere thanks to Dr.Amirtham Rajagopal for his valuable inputs during our group meetings which helpful for my research.

My special thanks to Mr. V. Kiran Kumar(Research Scholar) for his support and patience in making me understand the basic concepts and was very helpful during my presentations and in writing thesis.

I would like to thank all the members of the Immersive Multimedia and Telepresence lab, I had a great time working with them.

Above all I would like to thank my parents and all my family members for constantly supporting me in building my career.

Dedication

To My Parents

Abstract

Automatized life like representation of natural objects has been a cherished goal for humanity. Towards achieving this goal we propose a novel framework for the reconstruction of the 3D object. We laid the foundation for the representation of the signal by a developing a theory which deals with different sampling techniques(both Uniform and Non-uniform) for signals in the euclidean space, finite element method(Interpolative basis) for signals in the topological domain and finally the compressive sampling using which we can capture and represent the compressible signals by exploiting the sparsity.

Multiple view camera array is considered to capture the whole 360^0 view of the object, also for the reason that single camera cannot provide information about 3D content. 3D reconstruction from the multiple views captured necessitate estimation of the camera parameters. Existing camera calibration methods either require an external object or they may not provide unique camera parameters which introduces ambiguity in 3D reconstruction. Hence a novel auto calibration method has been proposed based upon Factorization algorithm and implemented using images captured from multiple views. Auto calibration requires finding corresponding points in all views captured(could be more than single camera). Whole 3D manifold is generated by iteratively applying aforementioned calibration method on a selected neighborhood around the corresponding points found in the previous iteration.

3D reconstructed data obtained will be generally very huge in size which puts a constraint on real time transmission. Compressing the data without losing the quality of reconstruction is challenging. In this regard we framed 3D compression problem in Compressive Sensing framework. Solution to this framework is possible if we can construct a basis for the manifold generated under which it has a sparse representation. We demonstrated this for a analogous problem of 2D image super resolution where a high resolution images is generated from a single low resolution image.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	vi
Nomenclature	viii
1 Introduction	1
1.1 3D Telepresence System	1
1.1.1 Acquisition of Multiple 2D views:	2
1.1.2 3D Reconstruction:	2
1.1.3 3D Compression:	3
1.1.4 3D Decompression and Rendering:	4
2 Signal Representation and Compressive Sensing	5
2.1 Uniform Sampling	5
2.2 Non-Uniform Sampling	6
2.2.1 MultiCoset Sampling	7
2.3 Interpolative Basis(FEM)	8
2.3.1 Finite Element Method Implication	8
2.3.2 Relating FEM to Image Processing	9
2.4 Compressive Sensing	9
2.4.1 Compressive Sensing Theory	9
3 3D Reconstruction	13
3.1 Problem statement	13
3.2 Camera Modelling	13
3.3 Solution	14
3.4 Point Correspondence	14
3.4.1 Interest Point Detection	14
3.4.2 Building Descriptor:	17
3.4.3 Matching	17
3.5 Correspondence Matching in Multiple Images	17
3.6 Auto Calibration of Multi-Camera Array	18
3.6.1 Factorization Algorithm	19

3.6.2	Auto Calibration of Multiple View camera array	20
3.7	Generating Manifold	24
4	Compressive Sensing Framework of 3D Reconstruction	26
4.1	3D Compression: A Compressive Sensing Framework	26
4.1.1	Problem Statement:	26
5	Image Super Resolution	28
5.1	Super Resolution Problem in Compressive Sensing Domain	28
5.2	Problem Statement:	28
5.3	Solution:	29
6	Results and Conclusion	31
6.1	Point Correspondence	31
6.2	Auto Calibration	32
6.2.1	Kanade's Factorization	32
6.2.2	Proposed method(Notion of Visibility)	33
6.3	SuperResolution	34

Chapter 1

Introduction

In the field of communication, entertainment and medical image processing there is a demand for high precision representation of the signal. Various applications including environment capture, autonomous navigation and 3D Telepresence systems need life like representation of the objects. 3D Telepresence stands out to be an important application might necessitate the use of a multi-camera networks in different configurations to effectively capture all the features of the 3D object.

1.1 3D Telepresence System

The overall block diagram of the 3D Telepresence is given by Figure 1.1

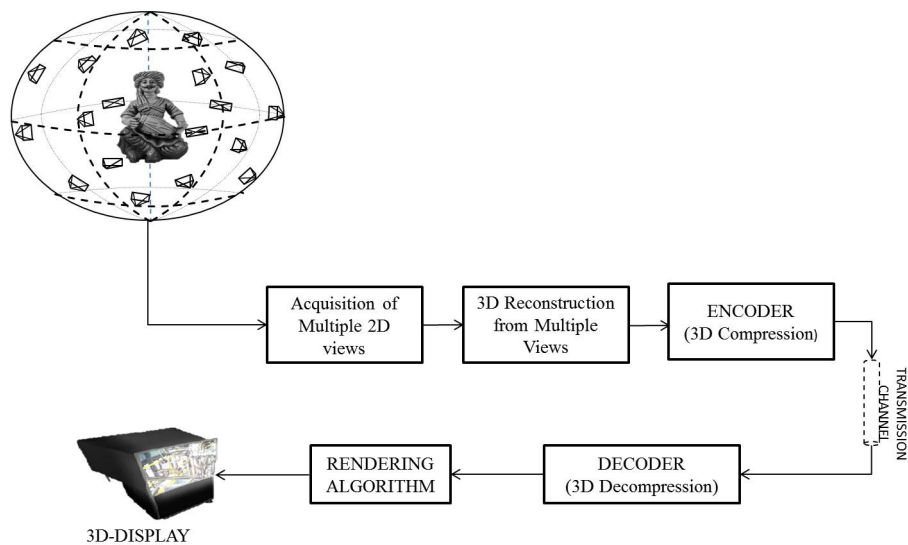


Figure 1.1: Block Diagram of the 3D Telepresence

A 3D Telepresence system should be able to reconstruct 3D object data captured from multiple images taken from different views and compress the data to transmit over communication channel. At the receiver end it should be able to decompress and represent the 3D data received maintaining color, luminous and texture consistency and render it to display. The block by block description of 3D Telepresence system is

1.1.1 Acquisition of Multiple 2D views:

A single camera cannot be able to represent the whole object, it can only capture single perspective(view). Hence it is intuitive that a 3D Telepresence system need more than one cameras to represent whole object. As shown in the figure 1.1 the input to system is an array of 2D images captured in different views covering whole 360° space around the object(person). It is obvious that how many cameras are required? and where to place the cameras? are two basic questions that strikes to our mind. But there is no proper research was done in this direction.

1.1.2 3D Reconstruction:

Multiple 2D images, captured by the network, play a central role in providing the depth related information that is difficult to perceive from individual 2D images. 3D reconstruction from single views is not a straight forward problem as we loose one dimension(depth) in the process of capturing image from a camera. This is considered as a ill-posed problem. The basic pinhole camera model can be seen in Figure 1.2.

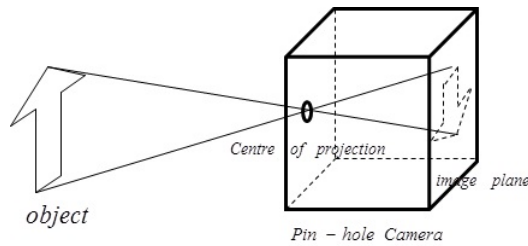


Figure 1.2: pinholecamera.

To lay down the mathematical framework for 3D object reconstruction, the working of a pin-hole camera needs to be completely understand and the transformation it affects on the 3D object when converting it into a 2D image. The set-up illustrated in Figure 1.3. shows how a 3D world coordinate is captured on to a 2D image plane.

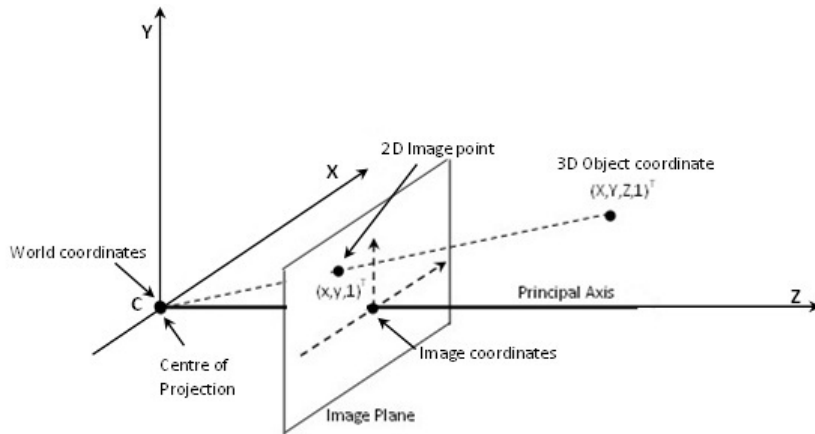


Figure 1.3: Camera and image plane placement in the world coordinate system.

The drop from three-dimensional space to a two-dimensional image is a projection in which one

dimension is lost. This is modeled using a process of central projection in which a ray from a point in space is drawn from the 3D object through the *center of projection*. This will intersect the *image plane* at a corresponding *image point*. To completely characterize the transformation between the image and the real world co-ordinates, the complexity of the situation can be increased incrementally starting from a base case. Consider a 3D projective space represented by P^3 called the *world co-ordinates* where the points are written in terms of homogeneous co-ordinates $(X, Y, Z, 1)^T$. The camera is placed with the origin, $(0, 0, 0, 1)^T$, as the center of projection \mathbf{C} and its principal axis aligned with the Z-axis. The *image plane*, represented by P^2 where the points are written in terms of homogeneous co-ordinates $(x, y, 1)^T$, is placed at a distance Z equal to the focal length of the camera and parallel to the XY-plane. The 2D image co-ordinates are defined with the origin at the point of intersection of the Z-axis and the image plane. The matrix that transforms the object point to the image point is denoted by P and is called the camera projection matrix and can be expressed as:

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1.1)$$

Most existing data acquisition systems for 3D reconstruction use stereo cameras. The idea behind stereo vision is to mimic human biology by trying to recreate the behavior of human eyes. The eyes behave like two pin-hole cameras displaced by a certain distance. Each eye generates a slightly different perspective of the 3D scene and the brain then extracts position and depth information from these two 2D images. Stereo cameras with parallel axes model the human eye and are the ideal choice for 3D data acquisition applications. But stereo camera gives only depth information from only one perspective i.e., complete representation of the object is not possible. Multiple camera array is a possible solution to obtain complete information. Multiple 2D images, captured by the network, play a central role in providing the depth related information that is difficult to perceive from individual 2D images. 3D shape reconstruction using visual hulls generated based on Silhouettes of images captured from multiple views is an interesting method which is suitable for multiple camera systems.

1.1.3 3D Compression:

Generally for the best representation of the signal the samples should be continuous, but with continuous signals we face problem while encoding i.e we require infinite number of bits to encode which is not practical. So we digitize the signal so that we can encode and transmit. But when we digitize the signal we loose some information, therefore we cannot have best possible representation of the signal at the decoder or receiver. So, we should look for some better way of representing the signal so that no or less information is lost.

When we captured an image with the help of a camera we will be getting the data as set of intensity values representing a particular frame. 3D reconstructed data from multiple 2D views will generally be very large but there are limitations on bandwidth allocated to transmit, also the processing delay should be very low for real time transmission. We can never have a camera or a sensor that will capture

points of interest i.e., the number of points that a camera can capture is nothing but fixed. So, there is high likelihood that we can have some redundant information in the views captured (Overlapping cameras is a trivial example). So how to get rid of this redundancy. One can pose a very interesting problem here i.e., Can we put a constraint on the number of views to be taken to have a minimal representation of the data?. what type of sampling one should prefer to balance both quality and bandwidth requirements.

In the transmitter (Encoder) 3D reconstruction of the object (person or scene) is done from 2D images captured in multiple views. The reconstructed 3D image is compressed and transmitted over the communication channel.

1.1.4 3D Decompression and Rendering:

The decompression algorithm is run at the receiver based upon the compression algorithm used in the transmitter. 3D rendering is one of the very important aspects of the 3D-Telepresence system. The 3D display should be able to provide viewer dependent view (motion parallax) and also it should be able to maintain color consistency, luminous consistency.

The most common solutions are stereoscopic displays with tracking. Stereoscopic displays (Ezra, 1995) can emit only two distinguishable light beams from each pixel, this is the reason for the compromises: the viewer dependent view (that is, the 3D scene is correct only from a single point of view), thus the necessity of tracking (Woodgate, 1998) to create motion parallax, but still, this will provide a correct view only for the driver (who leads the session and wears the object that is tracked). Perspective for all other participants who are not looking at the same direction will be incorrect. Tracking systems also introduce a small amount of latency, which can be reduced, but still disturbing. All these limitations are responsible for the seasickness and headache after using these systems for longer sessions. There are many more 3D display technologies but failed to give an elegant solution.

Chapter 2

Signal Representation and Compressive Sensing

In this chapter we will discuss the various methods of representing the signal. Here we are trying to remove the redundancy so that we can simultaneously achieve both compression and having a sparse representation. We will start with the most basic form of representing the signals i.e; uniform sampling and then introduce Non-uniform sampling[Multi coset sampling] further we will go through the most commonly used representation used in the image processing domain which is the wavelet basis and then about the most recent development which is the contourlets and then discuss about the theory of Finite Element Method which uses interpolative basis for representation of the signal and finally the compressive sensing which requires a basis under which the signal has a sparse representation.

2.1 Uniform Sampling

Continuous signals are represented in computers by their samples. The samples are taken according to the famous Uniform sampling theorem. Uniform sampling is the shanon-nyquist sampling theorem which states that when a signal(considering 1D-signal) is bandlimited in the frequency domain say B Hz it has to be sampled at a rate $\geq 2B$ Hz which is the Nyquist rate. The samples that we obtain by using the Nyquist rate are placed at uniform intervals of $\frac{1}{2B}$ Hz. In essence, it shows that a bandlimited analog signal that has been sampled can be perfectly reconstructed from an infinite sequence of samples if the sampling rate exceeds $2B$ samples per second. An image can be thought of as Piecewise constant function, or Uniform sampling of some underlying function.

But when the signal is not bandlimited but is a multiband signal with finite frequency spectral support then uniform sampling of such a signal at nyquist rate leads to more number of samples, so we consider the non uniform sampling[Multi coset Sampling] which is described in the following section.

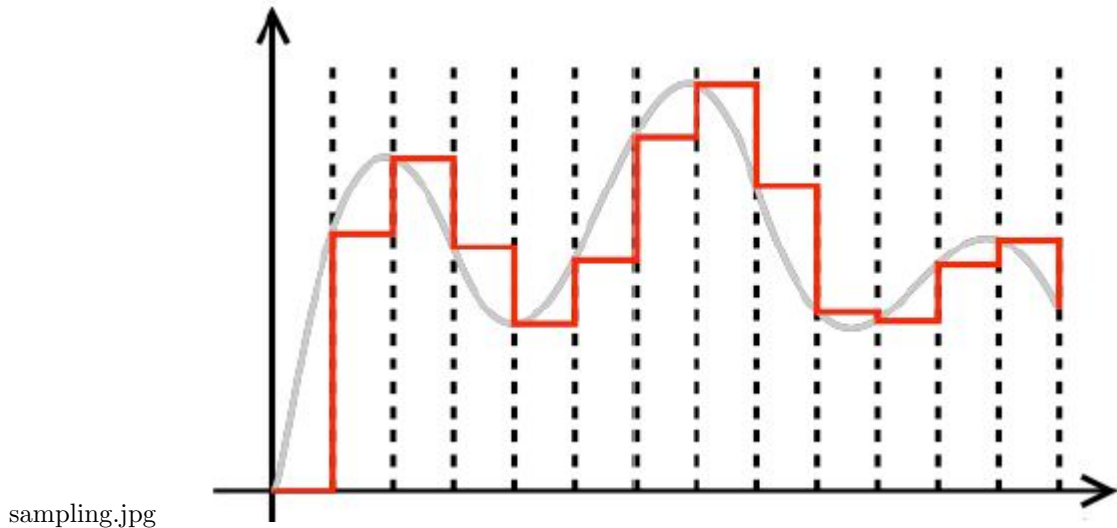


Figure 2.1: Illustration of Sampling in 1D

2.2 Non-Uniform Sampling

In this section we will explain multi coset sampling which is a Non-uniform sampling technique. Multi-coset sampling is a periodic nonuniform sub-Nyquist sampling technique for acquiring continuous-time spectrally-sparse signals.

A multiband signal $x(t)$ is a bandlimited, continuous-time, squared integrable signal that has all of its energy concentrated in one or more disjoint frequency bands (of positive Lebesgue measure). Denoting the Fourier transform of $x(t)$ by $X(j\omega)$,

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt$$

a bandlimited signal is one whose spectrum is bounded, i.e., $X(j\omega) = 0$ for $-\pi W \leq \omega \leq \pi W$ radians per second, for some positive real number W . Here, $W/2$ is the bandwidth of $x(t)$ and W is therefore the Nyquist frequency. The spectral support of a multiband signal is the union of the frequency intervals that contain the signals energy. A sparse multiband signal is thus a multiband signal whose spectral support has Lebesgue measure that is small relative to the overall signal bandwidth. If, for instance, all the active bands have equal bandwidth B Hz and the signal is composed of K disjoint frequency bands, then a sparse multiband signal is one satisfying $KB \ll W$.

Multi Coset Sampler: Multi-coset sampling (MC) is a periodic nonuniform sub-Nyquist sampling technique for acquiring sparse multiband signals [9]. For a fixed time interval T that is less than or equal to the Nyquist period and for a suitable positive integer L , Multi coset samplers sample $x(t)$ at the time instants $t = (kL + c_i)T$ for $1 \leq i \leq q, k = 0, 1, \dots$. The time offsets c_i are distinct, positive real numbers less than L and are known collectively as the multi-coset sampling pattern. The system thus collects $q \leq L$ samples in LT seconds, or equivalently, exhibits an average sampling rate of q/LT Hz. Here we set T equal to the Nyquist period $T = 1/W$, thereby referencing the systems sampling rate to the Nyquist rate. Multi-coset samplers are parameterized by $q, L, \text{ and } c_i$, and the system design depends on conditioning them properly to ensure successful recovery of $x(t)$

from the output samples. MultiCoset samplers are most easily implemented as multichannel systems where channel i shifts $x(t)$ by c_i/W seconds and then samples uniformly at W/L Hz see figure 2.2.

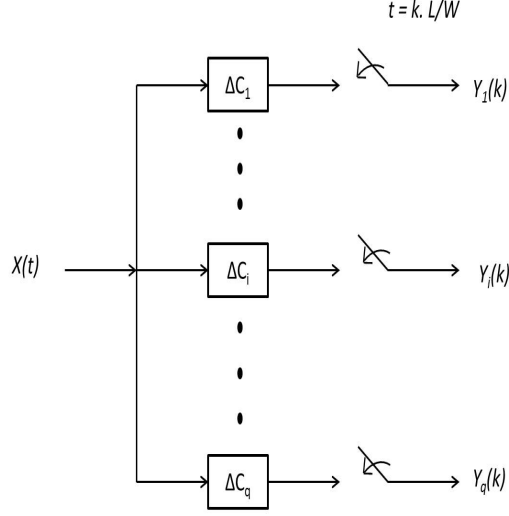


Figure 2.2: Multi-coset sampler implemented as a multi-channel system. In this case, the base sampling period equals the Nyquist rate W Hz.

2.2.1 MultiCoset Sampling

Let $x(t)$ be a sparse multiband signal. Then by inspection of 2.2 we have the following time and frequency domain relationships for the i^{th} channel, $i = 1, \dots, q$.

Shifting in time :

$$x(t + c_i/W) \quad \xleftrightarrow{FT} \quad X(j\omega)e^{jc_i/W\omega}$$

Sampling/Aliasing:

$$y_i(k) = x(kL/W + c_i/W)$$

DTFT of the above equation

$$Y_i(e^{j\omega L/W}) = \frac{W}{L} \sum_{m=-\lceil \frac{L}{2}(\frac{\omega}{\pi W}-1) \rceil+1}^{\lfloor \frac{L}{2}(\frac{\omega}{\pi W}+1) \rfloor} X(j\omega - 2\pi \frac{W}{L}m)e^{-j\frac{c_i}{W}(\omega - 2\pi \frac{W}{L}m)}$$

The summation limits are finite for a given ω because $x(t)$ is assumed bandlimited. Because $Y_i(e^{j\omega L/W})$ is periodic with period $2\pi W/L$, we can, without loss of information, restrict $Y_i(e^{j\omega L/W})$ to one period.

$$e^{-j\frac{c_i}{W}\omega}Y_i(e^{j\omega L/W}) = \frac{W}{L} \sum_{m=-\lceil \frac{L}{2}(\frac{\omega}{\pi W}-1) \rceil+1}^{\lfloor \frac{L}{2}(\frac{\omega}{\pi W}+1) \rfloor} X(j\omega - 2\pi \frac{W}{L}m)e^{-j\frac{2\pi}{L}c_i m}$$

for $i = 1, \dots, q$, where $1_{[\cdot]}$ denotes the indicator function. Note that the restriction to $[\pi W/L, \pi W/L)$ removes the dependence on ω in the summation limits since within this interval $Y_i(e^{j\omega L/W})$ is a linear combination of a particular (finite) set of spectral segments of $x(t)$. We can therefore write this

expression in a matrix-vector formulation

$$\mathbf{z}(\omega) = \phi \mathbf{s}(\omega)$$

where

$$z_i(\omega) = e^{-j\frac{c_i q}{W}\omega} Y_i(e^j \omega L/W) \mathbf{1}_{[-\frac{\pi W}{L}, \frac{\pi W}{L}]}$$

$$\phi_{i,l} = \frac{W}{L} e^{-j\frac{2\pi}{L} c_l m_l}$$

$$s_l(\omega) = X(\omega - 2\pi \frac{W}{L} m_l) \mathbf{1}_{[-\frac{\pi W}{L}, \frac{\pi W}{L}]}$$

for $i = 1, \dots, q, l = 1, \dots, L$, and $m_l = -\lfloor \frac{1}{2}(L+1) \rfloor + l$

2.3 Interpolative Basis(FEM)

Both uniform and Non-uniform sampling used to represent a class of functions with domain as the euclidean space. But to represent a class of functions on a topological space we need an interpolative basis. Using the interpolative basis we can get back the surface by consider only a subset of signal points on the topological surface. Finite Element Method (FEM) is in a topological domain, allows more complex element behavior to be modeled. The FEM was originally just an extension of matrix structural analysis, developed by structural engineers. It has since been used in just about every field where differential equations define the problem behavior. The basic idea of the finite element method is to break up a continuum into a discrete number of smaller "elements". These elements can be modeled mathematically by a stiffness matrix and are connected by nodes that have degrees of freedom.

2.3.1 Finite Element Method Implication

It is a numerical procedure for obtaining approximate solutions to many of the problems encountered in engineering analysis. FEM use interpolative basis to reconstruct the total image. Here we see a problem which is solved using the FEM. Given an initial model representing general knowledge of the object, and incomplete or missing information about geometry or material properties. The method is based on iterative analysis of the difference between the actual and predicted behaviour. Large differences indicate that an objects properties are not captured properly by the model describing it. These error are due to flaws in the model parameter estimation such as geometry and material properties. 'P' is sparse points, 'Q' is set of correspondence [10] of guide search.

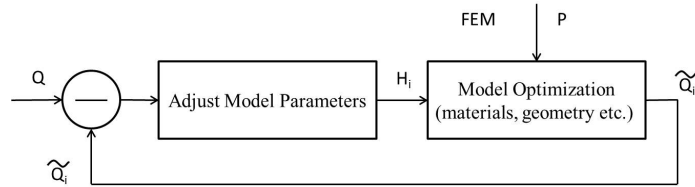


Figure 2.3: General approach

Example:-Hand Motion Analysis

Since large deformations occur during any hand motion, physical laws should be used to model not only the skeletal motion, but also the nature of the deformations in soft tissues. Research has shown that finite element theory can be used to model near-correct muscular motion. Frame-to-frame correspondence recovery is based on the iterative analysis of the directed Hausdorff distance between the model and the next frame in the sequence. Graphically interpreting nonlinear behavior through animation allows us to verify and visualize displacement results. An obvious high pressure around the base of the thumb is readily visible.

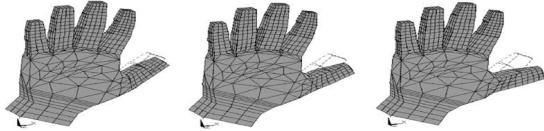


Figure 2.4: (a) Motion model of left hand and its analysis [10]

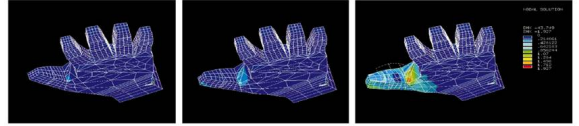


Figure 2.5: (b) Motion model of right hand and its analysis [10]

2.3.2 Relating FEM to Image Processing

FEM has many applications in the field of image processing. Many parts of the body will be reconstructed as 3D models with the help FEM from the slices of 2D images like CT-Scan, MRI-Scan etc. FEM model representation of surface is nothing but interpolative basis selection, where we select the node points to represent total object. For transmission of 3D data we need more bandwidth. In real world it is not possible to dedicate more bandwidth for single user. So, we do compression of this 3D data by modeling using FEM. We use the mesh nodes for represent surface, from which we will reconstruct the total object. Any point on 3D surface can be represent by using the function of nodes points. My teammate Mr. Srikanth is also subjecting the same problem with the help of compressive sensing framework. Our objective is to select the node points in such way that we will reconstruct the object with better representation. In the above hand model we discussed about the changing the properties by using the motion of hand [10]. But our problem is to change the node points such way that we will get the good representation of object.

2.4 Compressive Sensing

Compressive Sensing or Compressive Sampling is the recent advancement in signal processing. According to Compressive Sensing theory if a signal has a sparse representation in a particular basis then it can be recovered far less samples than the number of samples according to Nyquist Sampling theorem.

2.4.1 Compressive Sensing Theory

Consider a real-valued, finite-length, one-dimensional, discrete-time signal x , which can be viewed as an $N \times 1$ column vector in \mathbb{R}^N with elements $x[n], n = 1, 2, \dots, N$. (We treat an image or higher-dimensional data by vectorizing it into a long one-dimensional vector.). Any signal in can be repre-

sented in terms of a basis of $N \times 1$ vectors . For simplicity, assume that the basis is orthonormal. Using the $N \times N$ basis matrix $\psi = [\psi_1|\psi_2|\dots|\psi_N]$ with the vectors ψ_i as columns, a signal x can be expressed as

$$X = \psi s. \tag{2.1}$$

where s is the $N \times 1$ column vector of weighting coefficients $s_i = \langle x, \psi_i \rangle = \psi^T X$ and T denotes transposition. Clearly, x and s are equivalent representations of the signal, with x in the time or space domain and s in the ψ domain. The signal x is K -sparse if it is a linear combination of only K basis vectors; that is, only K of the s_i coefficients are non zero and $(N-K)$ are zero. The case of interest is when $K \ll N$. The signal x is compressible if the representation has just a few large coefficients and many small coefficients.

Transform Coding and its Inefficiencies: The fact that compressible signals are well approximated by K -sparse representations forms the foundation of transform coding . In data acquisition systems (for example, digital cameras) transform coding plays a central role: the full N -sample signal x is acquired; the complete set of transform coefficients s_i is computed via $s_i = \psi^T x$; the K largest coefficients are located and the $(N-K)$ smallest coefficients are discarded; and the K values and locations of the largest coefficients are encoded. Unfortunately, this sample-then-compress framework suffers from three inherent inefficiencies. First, the initial number of samples N may be large even if the desired K is small. Second, the set of all N transform coefficients s_i must be computed even though all but K of them will be discarded. Third, the locations of the large coefficients must be encoded, thus introducing an overhead.

Compressive Sensing Problem: Compressive sensing address these inefficiencies by directly acquiring a compressed signal representation without going through the intermediate stage of acquiring N samples. Consider a general linear measurement process that computes $M \ll N$ inner products between X and a collection of vectors $(\phi_j)_{j=1}^M$ as in $Y_j = \langle X, \phi_j \rangle$. Arrange the measurements Y_j in an $M \times 1$ vector Y and the measurement vectors ϕ_j^T as rows in an ϕ $M \times N$ matrix . Then, by substituting ψ from 2.1, Y can be written as

$$Y = \phi X = \phi \psi s = \Theta s$$

where $\Theta = \phi \psi$ is an $M \times N$ matrix. The measurement process is not adaptive, meaning that ϕ is fixed and does not depend on the signal X . The problem consists of designing a) a stable measurement matrix ϕ such that the salient information in any K -sparse or compressible signal is not damaged by the dimensionality reduction $X \in \mathbb{R}^N$ to $Y \in \mathbb{R}^M$ and b) a reconstruction algorithm to recover X from only $M \ll N$ measurements Y (or about as many measurements as the number of coefficients recorded by a traditional transform coder).

Designing a Stable Measurement Matrix:

The measurement matrix ϕ must allow the reconstruction of the length- N signal X from $M \ll N$ measurements (the vector Y). Since $M \ll N$, this problem appears ill-conditioned. If, however, X is K -sparse and the K locations of the nonzero coefficients in s are known, then the problem can be solved provided $M \geq K$. A necessary and sufficient condition for this simplified problem to be well

conditioned is that, for any vector v sharing the same K nonzero entries as s and for some $\epsilon > 0$

$$1 - \epsilon < \frac{\|\Theta v\|_2}{\|v\|_2} < 1 + \epsilon$$

That is, the matrix Θ must preserve the lengths of these particular K -sparse vectors. Of course, in general the locations of the K nonzero entries in s are not known. However, a sufficient condition for a stable solution for both K -sparse and compressible signals is that Θ satisfies (3) for an arbitrary $3K$ -sparse vector. This condition is referred to as the *Restricted Isometry property (RIP)*.

Designing a Stable Reconstruction Algorithm: The signal reconstruction algorithm must take the M measurements in the vector Y , the random measurement matrix ϕ and the basis ψ and reconstruct the length- N signal X or, equivalently, its sparse coefficient vector s . For K -sparse signals, since $M < N$ there are infinitely many \hat{s} that satisfy $\Theta\hat{s}=Y$. This is because if $\Theta s = Y$ then $\Theta(s+r) = Y$ for any vector r in the null space $N(\Theta)$ of Θ . Therefore, the signal reconstruction algorithm aims to find the signals sparse coefficient vector in the $(N - M)$ dimensional translated null space $H = N(\Theta) + s$.

- Minimum l_2 norm reconstruction:

Define the l_p norm of the vector s as $(\|s\|_p)^p = \sum_{i=1}^N |s_i|^p$. The classic approach to inverse problems of this is to find the vector in the translated null space with the smallest l_2 norm (energy) by solving

$$\hat{s} = \operatorname{argmin} \|\hat{s}\|_2 \text{ such that } \Theta\hat{s} = Y$$

This optimization has the convenient Closed form solution $\hat{s} = \Theta(\Theta\Theta^T)^{-1}Y$. Unfortunately, l_2 minimization will almost never find a K -sparse solution, returning instead a nonsparse \hat{s} with many nonzero elements.

- Minimum l_0 norm reconstruction:

Since the l_2 norm measures signal energy and not signal sparsity, consider the l_0 norm that counts the number of non-zero entries in s . (Hence a K -sparse vector has l_0 norm equal to K). The modified optimization

$$\hat{s} = \operatorname{argmin} \|\hat{s}\|_0 \text{ such that } \Theta\hat{s} = Y$$

can recover a K -sparse signal exactly with high probability. Unfortunately, solving (5) is both numerically unstable and not tractable, requiring an exhaustive enumeration of all $\binom{N}{K}$ possible locations of the nonzero entries in s .

- Minimum l_1 norm reconstruction:

Surprisingly, optimization based on the l_1 norm

$$\hat{s} = \operatorname{argmin} \|\hat{s}\|_1 \text{ such that } \Theta\hat{s} = Y$$

can exactly recover K -sparse signals and closely approximate compressible signals with high probability using only $M \geq cK \log(N/K)$ iid Gaussian measurements.

Here we looked at different ways of representing the signal i.e; from a finite discrete set of points

we can capture the whole object. We developed this framework that will be useful in later chapter which is a real life problem.

Chapter 3

3D Reconstruction

Here we will consider the real life application of reconstruction of 3D object. The problem is to reconstruct the 3D object faithfully without losing any information. Intuition says that information from a single perspective is not sufficient to reconstruct the whole 3D object. So we need a multicamera array for capturing the entire information about the object.

3.1 Problem statement

To reconstruct a 3D object from images (say M) taken from different views covering the whole 360 space around the object.

3.2 Camera Modelling

To reconstruct the 3D object we need to get back the depth information that was lost while capturing the object using the camera. To understand how a camera projects a point in the 3D world coordinate system into the image coordinate system we consider the Camera Modelling.

Camera projects a 3D point on to the 2D image plane from the first principles of optics. This is known as projective transformation that defines how real-world objects are projected on the image plane. Projective transformation is defined by camera intrinsic and extrinsic parameters. The four most important parameters define the focal length in x and y direction and the possible displacement of the image center away from the optic axis (known as principal point). The focal length ideally is the distance between the center of projection and the image plane. These parameters are summarized in the matrix, which is called the intrinsic camera matrix. The matrix of intrinsic parameters does not depend on the scene viewed and, once estimated, can be re-used (as long as the focal length is fixed (in case of zoom lens)). The following equation gives the basic structure projective geometry and the calibration parameters are useful in finding the 3D world coordinates.

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & | & t_1 \\ r_{21} & r_{22} & r_{23} & | & t_2 \\ r_{31} & r_{32} & r_{33} & | & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.1)$$

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} \left[\begin{array}{c|c} R & t \end{array} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.2)$$

where the 3×3 matrix K is dictated by the internal parameters of the camera, and the 3×4 matrix $[R|t]$ by external parameters.

where (u_0, v_0) denotes the image coordinate of the point where the principal axis meets the image plane, f_x and f_y are focal lengths along image coordinate axes, and γ is a skewness index. If an image from camera is scaled by some factor, all of these parameters should be scaled (multiplied/divided, respectively) by the same factor.

Further, $[R|t]$, where the 3×3 matrix R is unitary, and indicates 3D rotation operation, where 3×1 vector t collects three translation parameters along the three world coordinate axes.

3.3 Solution

Outline: The solution involves dealing with an inverse problem. We find set of corresponding points through all the input images. The visibility matrix takes care of points present only in a subset of images. The matrix containing corresponding points (x) now has to be factorized in such a way that we recover the structure of Projective motion \hat{P} and also projective shape $\hat{X}(x = PX)$. This is achieved by rank 4 factorization.

The Various steps that are required to obtain the 3D object is illustrated by using a flow chart:

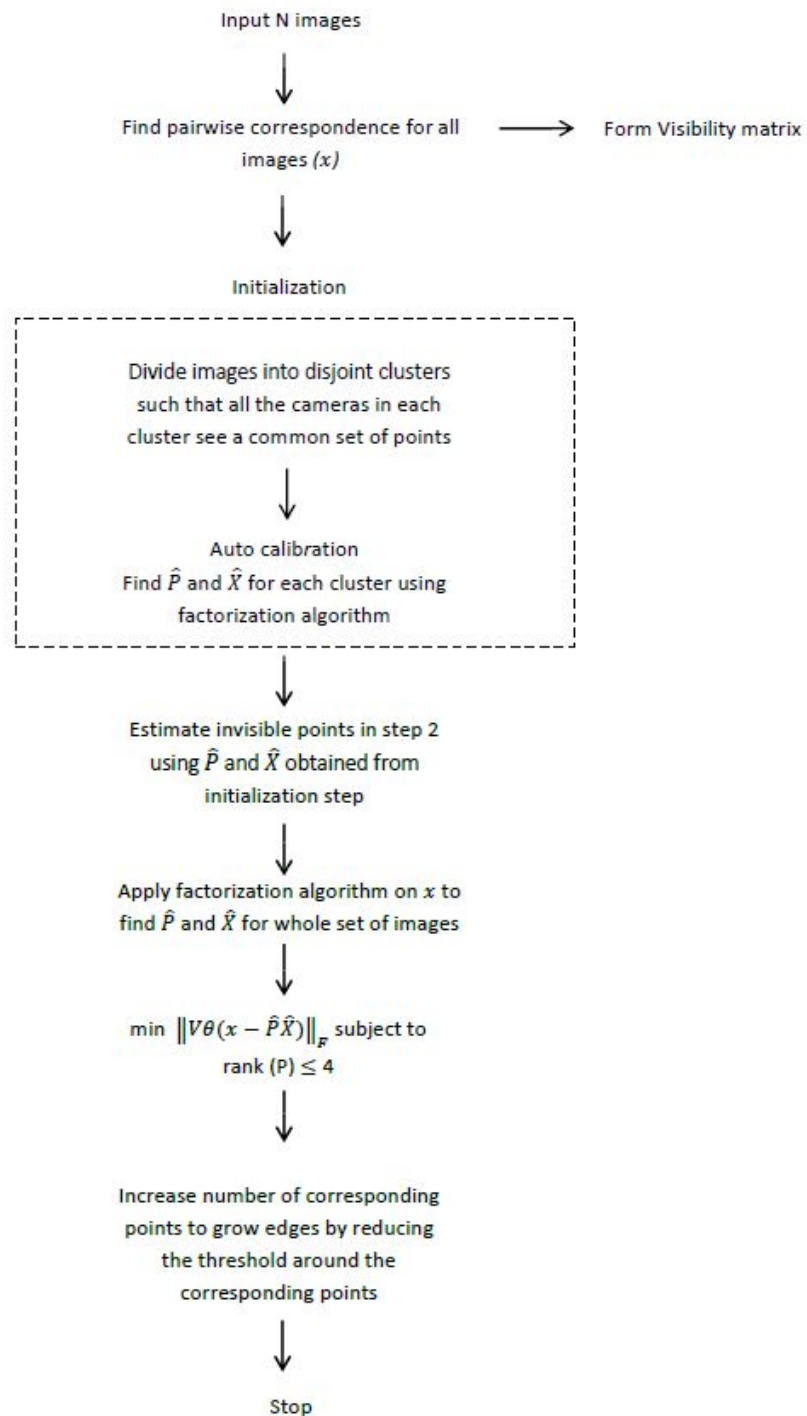
3.4 Point Correspondence

Finding point correspondences in two or more images has many applications such as image registration, object recognition and camera calibration. Correspondence matching consists of three steps :

- Interest point/Feature Point detection
- Feature Vector/Descriptor selection
- Matching

3.4.1 Interest Point Detection

Interest points are the points that are more or less easily differentiated from their surrounding points. Corners, blobs, T-junctions etc. are examples of such points. The obvious argument that can be made is, pixel values are very different at corners or edges than at the background. A large value of derivative, taken in either X or Y direction, indicates possibility of an edge and a large derivative taken in XY direction indicates presence of a corner. These are, of course, very crude methods of detection. Harris Corner Detector, Canny edge detector etc. are some widely used schemes. For detection of blob like regions and Hessian matrices are popular; that too detect points and edges.



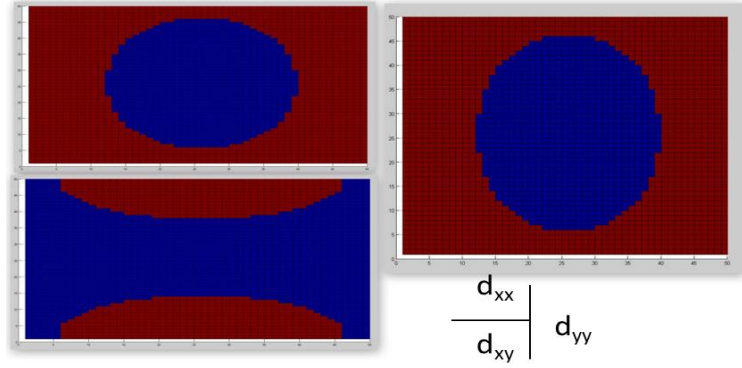


Figure 3.1: Masks used for interest point detection

Some methods use trace of Hessian matrix and some use determinant of it.

The concept behind Hessian matrix is as follows: Given a point $X = (x, y)$ in an image I , the Hessian matrix $H(X, \sigma)$ at X with scale σ is:

$$\mathcal{H}(X, \sigma) = \begin{pmatrix} L_{xx}(X, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(x, \sigma) \end{pmatrix}$$

where $L_{xy}(X, \sigma)$ is convolution of second order derivative of Gaussian kernel with image I at point X . The determinant value of Hessian matrix indicates presence of an Interest point; if the value is above some threshold. The threshold selection is an important step that determines number of points detected. The sign of the determinant indicates nature of the point(e.g.dark point on lighter background).We also quantize the LoG values to integers,then the filter masks look like what is shown below:

Our method is very similar to Speeded Up Robust Features[1] but it excludes many of SURF's artifacts at present making it simpler for our purpose. We take a Gaussian Kernel at fixed variance and find determinant of Hessian matrix at each point in the images at hand. After observing the values of determinants for the particular image database we fix a threshold for recording a point as an Interest point.Original SURF[1] performs this task at multiple scales and localizes the point if it is detected at three scales. The interest point detected for an image is shown below:



points.jpg

Figure 3.2: Interest Points Detected for a Toy image

3.4.2 Building Descriptor:

Once an Interest point is detected, next task is to assign a feature vector or descriptor to it, which is needed for Matching step. For finding a feature vector a neighbourhood around every Interest point is selected. Then various properties of these neighbourhood are extracted that form the feature vector. SURF [1] suggests to find a dominant orientation (using a scale dependant circular neighbourhood) around an Interest point before finding a feature vector. This gives rotation invariance to the descriptor in matching step. For our application we do not practice this part. Next, the neighbourhood selection is advised to be scale dependant (the scale at which Interest point was detected); which is fixed in our case.

Having selected a neighbourhood we take sum of pixelwise differences in x and y directions along each row and column of neighbourhood matrix, respectively. These are recorded as $\sum dx$ and $\sum dy$ respectively. In order that these differences should not cancel each other (giving a sum zero for a row or column, falsely showing lack of distinguishing properties around Interest point); we also record sum of absolute values of these differences which are $\sum |dx|$ and $\sum |dy|$ respectively. This constitutes a $9 \times 4 \times 3$ descriptor in our case, which takes colour information into account. This offers more robustness.

$$\text{Feature Vector} = \begin{pmatrix} \sum dx \\ \sum dy \\ \sum |dx| \\ \sum |dy| \end{pmatrix}_{9 \times 4 \times 3}$$

The colour part is absent in original SURF for computation purpose. Histogram of the neighbourhood is also a factor in feature vector which is used by SIFT and GLOH in different ways.

3.4.3 Matching

Matching step requires calculating distance between feature vectors of Interest points in two images. Choice of distance measure and threshold for recording a correspondence match determines quality of overall scheme. Distance measure can be Euclidean distance, Mahalanobis distance, etc. We use sum of squared differences method. We select one point each in two images between which point correspondences are to be found. Then we take total 108 (= $9 \times 4 \times 3$; size of feature vector for each point) differences and add them. This gives a score for two points in reference. In this way we try to match a point with each and every point in next image. The pair which gives least SSD score is recorded as corresponding points pair.

3.5 Correspondence Matching in Multiple Images

Here we explain the method to find correspondences in multiple images. An Interest point is first selected in an image, then with a full search in next image we try to find corresponding point. During next iteration we select only those points in the second image which have correspondence in first image and do a full search in third image and so on for multiple images. This is achieved by simple management of multidimensional flags assigned to each Interest point. This also reduces

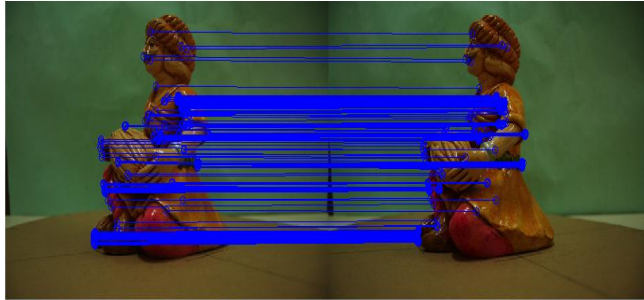


Figure 3.3: point correspondence in two images

computation time as we perform a selective search except for first iteration. We end up with point correspondences in k images out of N images entered. Figure below shows the correspondence for 4 images. It is intuitive that as the number of images increases the point correspondences in multiple

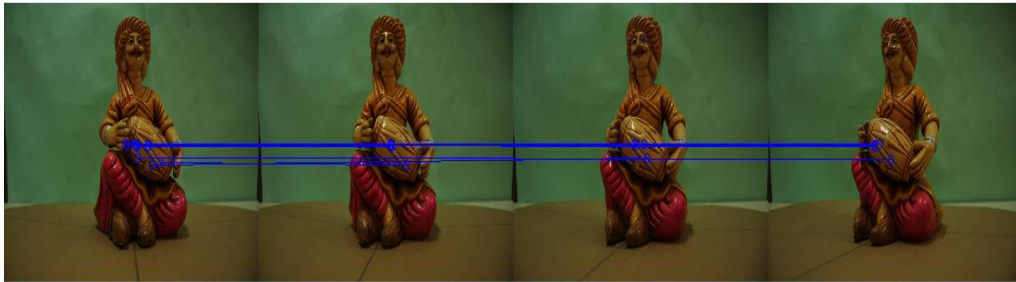


Figure 3.4: Correspondence Matching for 4 images

images decreases. In order to get more corresponding points we first select a small neighborhood around the corresponding point and perform the above steps by reducing the threshold value for matching around the small neighbourhood of the corresponding point in the multiple images and again perform the matching so that we get more corresponding points.

3.6 Auto Calibration of Multi-Camera Array

For 3D reconstruction using images from multiple views, simultaneous camera calibration and localization of multiple camera array is the most rudimentary and most significant step in order to extract the 3D attributes from the 2D images. Many researchers from Computer Vision, Image Processing domains tried to solve this problem, but still there is no elegant solution to this problem. To calibrate a single camera, one would have to determine the 5 internal and 4 external parameters that govern image formation. In a multi-camera network, this problem is further compounded, since for want of efficiency each camera cannot be calibrated individually.

The calibration of a network of cameras employs concepts of multi-view geometry[2], like epipolar

geometry and projective transformations in addition to the basic physics of image formation. However, most traditional calibration algorithms are photogrammetric, i.e. they use a calibration target like a checkerboard. In [8] two step calibration method was used to compute the intrinsic, extrinsic and localization parameters with the help of virtual object(marker detection) and Vision Graphs. First step constitutes of calculating intrinsic and extrinsic parameters of each camera using Tsai's method [5], in the second step external calibration using virtual calibration object and vision graphs to find 6 external parameters describing orientation and position of each camera with regard to selected reference camera.

This is not a scalable solution for a multi-camera array since the camera networks might be large and not all cameras in the network may be able to view the target. Also, this method has the basic limitation of using a cumbersome target and an elaborate set-up as the size of the target will also be a problem. This necessity of someone being physically present at the scene with a calibration target makes the process of multi-camera network deployment and data acquisition tedious. Hence, camera calibration would become robust if we were to somehow extract the internal and external parameters of the constituent cameras in the network from the 3D scene that is being captured itself, i.e. auto-calibration.

HP coliseum [6] uses a cube with four colored squares on each face (totaling 24 colors plus black and white) as a calibration object. The face components supply the determining the calibration parameters. Lens distortion correction is computed by determining the radial polynomial that straightens the target faces' black boundaries. Intrinsic parameters are found using Zhang's [3] method. External parameters are estimated in a two-stage process that starts with initial adjacent-pair pose estimates using a nonlinear variant of a stereo solver [7].

Although these traditional methods are accurate, they require physical access to the observed space and involve an offline precalibration stage for every configuration of the network. This is impractical and costly in most remote applications or deployments in hazardous locations. Deviating from conventional methods that employ calibration targets, self-calibration algorithms that compute the parameters of the camera from the scene or a set of uncalibrated images has been attempted and developed to an extent. Hartley [17] presents a stratified approach, assuming that the intrinsic parameters are constant, where an affine calibration stage is used to compute a rectifying homography H . These methods require computation of correspondences between images and requires much more overlap between cameras than might be available in camera networks. In such a setting, the factorization approach to the problem of extracting 3D shape information and camera parameters simultaneously proposed by Han and Kanade [4] is quite appealing.

3.6.1 Factorization Algorithm

The factorization-based method recovers shape and motion of the 3D object from multiple uncalibrated perspectives. They accomplish the task of computing 3D object shape and camera parameters by tracking a set of feature points on the object in all the multiple views. However, the basic drawback of this approach is the limitation that not all the feature points that have been marked for tracking may be visible in all the cameras in the network. Assuming that we have M cameras and N object points the overall projective transformation matrix is given by

$$\begin{aligned}
W_s &= \begin{bmatrix} s_{11} \begin{bmatrix} x_{11} \\ y_{11} \end{bmatrix} & \cdots & s_{1N} \begin{bmatrix} x_{1N} \\ y_{1N} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ s_{M1} \begin{bmatrix} x_{M1} \\ y_{M1} \end{bmatrix} & \cdots & s_{MN} \begin{bmatrix} x_{MN} \\ y_{MN} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_M \end{bmatrix} \begin{bmatrix} X_1 & \cdots & X_N \end{bmatrix} \\
\Rightarrow W_s &= \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_M \end{bmatrix} \begin{bmatrix} X_1 & \cdots & X_N \end{bmatrix} \tag{3.3}
\end{aligned}$$

Kanade's Iterative Projective Factorization Algorithm

1. Set $s_{ij} = 1$, for $i = 1 \dots n$ and $j = 1 \dots m$;
2. Compute the current scaled measurement matrix W_s by Equation (3.3);
3. Perform rank4 factorization on W_s , generate the projective shape and motion;
4. Reset $s_{ij} = P_i^{(3)} X_j$ where $P_i^{(3)}$ denotes the third row of the projection matrix P_i ;
5. If s_{ij} 's are the same as the previous iteration, stop; else go to step 2.

The goal of the projective reconstruction process is to estimate the values of the projective depths (s_{ij} 's) which make Equation (3.3) consistent.

The factorization of Equation (3.3) recovers the motion and shape up to a 4×4 linear projective transformation H :

$$W_s = PX \tag{3.4}$$

$$= PHH^{-1}X \tag{3.5}$$

$$= \hat{P}\hat{X} \tag{3.6}$$

where $\hat{P} = PH$ and $\hat{X} = H^{-1}X$

P and X are referred to as the projective motion and the projective shape. Any non-singular 4×4 matrix could be inserted between P and X to get another motion and shape pair.

3.6.2 Auto Calibration of Multiple View camera array

Let x be the measurement matrix of M cameras stacked that captures 3D points X which are visible from more than one cameras.

If we assume that the distance between the object center and the camera is large, then the scaling factor is independent of the position of the 3D object point and the camera projection can be modeled as:

$$s_i x_{ij} = P_i X_j$$

Case1: All points are visible to all cameras

Now, if all the N tracked feature points are visible from all the M cameras in the network, the global camera projection can be modeled as:

$$\begin{bmatrix} x_{11} & \dots & x_{1N} \\ y_{11} & \dots & y_{1N} \\ 1 & \dots & 1 \\ \vdots & & \vdots \\ x_{M1} & \dots & x_{MN} \\ y_{M1} & \dots & y_{MN} \\ 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_1}K_1[R_1|t_1] \\ \frac{1}{s_2}K_2[R_2|t_2] \\ \vdots \\ \frac{1}{s_M}K_M[R_M|t_M] \end{bmatrix} \begin{bmatrix} X_1 & \dots & X_N \\ Y_1 & \dots & Y_N \\ Z_1 & \dots & Z_N \\ 1 & \dots & 1 \end{bmatrix} \quad (3.7)$$

$$x = PX \quad (3.8)$$

Problem statement: To choose $\{P, X\}$ such that $\|x - PX\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$.

Solution: Choose P, X such that $\|x - \hat{P}\hat{X}\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$.

where \hat{P}, \hat{X} are found by singular value decomposition of x and subsequently picks the best rank-4 estimate to obtain the solution. In this preliminary case, Han and Kanade's algorithm can be implemented and the global projection matrix can be obtained by rank-4 decomposition. The brief description of algorithm is as follows

$$\text{From SVD, } x_{3M \times N} = U_{3M \times 3M} \Sigma_{3M \times 3M} V^T_{3M \times N}$$

where the singular values in Σ are arranged in descending order. To obtain the rank-4 decomposition estimate, we write

$$\hat{\Sigma} = \begin{bmatrix} \Sigma_{11} & 0 & 0 & 0 \\ 0 & \Sigma_{22} & 0 & 0 \\ 0 & 0 & \Sigma_{33} & 0 \\ 0 & 0 & 0 & \Sigma_{44} \end{bmatrix}$$

$$\text{where, } \Sigma = \begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{3M,3M} \end{bmatrix}$$

If $U = [u_1, u_2, \dots, u_{3M}]$, then $\hat{U} = [u_1, u_2, u_3, u_4]$. Similarly, if $V = [v_1, v_2, \dots, v_{3M}]$, then $\hat{V} = [v_1, v_2, v_3, v_4]$. Now, solving for $x = U\Sigma V^T = PX$,

$$\text{we write, } \hat{P} = \hat{U}\hat{\Sigma}H \quad \text{and} \quad \hat{X} = H^{-1}\hat{V}^T \quad \text{for every invertible } H$$

We now have $\hat{x} = \hat{P}\hat{X}$, thus for every choice of $(P, X) = (\hat{P}, \hat{X})$, we evaluate the frobenius norm to determine how much error the rank-4 decomposition would entail,

$$\begin{aligned}\|x - \hat{P}\hat{X}\|_F &= \|U\Sigma V^T - \hat{U}\hat{\Sigma}\hat{V}^T\|_F \\ &= \|\hat{U}^c\hat{\Sigma}^c\hat{V}^{cT}\|_F \\ &= \sum_{k=5}^{3M} \Sigma_{kk}\end{aligned}$$

where, $U = [\hat{U}|\hat{U}^c]$ and $V = [\hat{V}|\hat{V}^c]$. Thus, the task is now to choose P, X such that $\|x - \hat{P}\hat{X}\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$.

Case 2: All Points are visible to more than one cameras

In this case all points may not be visible to all cameras. Let Θ be the visibility matrix which defines what features points are visible to what camera. Now the observation matrix cannot be modeled as equation (3.3), this is because x will be having holes if any particular camera cannot see any point in X

Problem Statement: To choose $\{P, X\}$ such that $\|\Theta \odot (x - PX)\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$

Solution: Choose P, X such that $\|\Theta \odot (x - \hat{P}\hat{X})\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$.

where \hat{P}, \hat{X} are estimated by singular value decomposition (rank 4 decomposition) of x . The detailed algorithm to compute \hat{P}, \hat{X} is as follows

Initialization step: we break the M cameras into q clusters, such that every camera in the j^{th} cluster can see N_j tracked feature points. We now apply Han and Kanade's method to cluster j to obtain the initial estimates of \hat{P}_j and \hat{X}_j .

$$\text{From SVD, } x_{3M_j \times N_j} = U_{3M_j \times 3M_j} \Sigma_{3M_j \times 3M_j} V_{3M_j \times N_j}^T$$

where the singular values in Σ are arranged in descending order. For want of simplicity, we shall drop the index j referring to the j^{th} cluster, keeping in mind that this same procedure is adapted for every cluster. To obtain the rank-4 decomposition estimate, we write

$$\hat{\Sigma} = \begin{bmatrix} \Sigma_{11} & 0 & 0 & 0 \\ 0 & \Sigma_{22} & 0 & 0 \\ 0 & 0 & \Sigma_{33} & 0 \\ 0 & 0 & 0 & \Sigma_{44} \end{bmatrix}$$

where, $\Sigma = \begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{3M,3M} \end{bmatrix}$

If $U = [\underline{u}_1, \underline{u}_2, \dots, \underline{u}_{3M}]$, then $\hat{U} = [\underline{u}_1, \underline{u}_2, \underline{u}_3, \underline{u}_4]$. Similarly, if $V = [\underline{v}_1, \underline{v}_2, \dots, \underline{v}_{3M}]$, then $\hat{V} = [\underline{v}_1, \underline{v}_2, \underline{v}_3, \underline{v}_4]$. Now, solving for $x = U\Sigma V^T = PX$,

$$\text{we write, } \quad \hat{P} = \hat{U}\hat{\Sigma}H \quad \text{and} \quad \hat{X} = H^{-1}\hat{V}^T \quad \text{for every invertible } H$$

We now have $\hat{x} = \hat{P}\hat{X}$, thus for every choice of $(P, X) = (\hat{P}, \hat{X})$, we evaluate the Frobenius norm to determine how much error the rank-4 decomposition would entail,

$$\begin{aligned} \|x - \hat{P}\hat{X}\|_F &= \|U\Sigma V^T - \hat{U}\hat{\Sigma}\hat{V}^T\|_F \\ &= \|\hat{U}^c\hat{\Sigma}^c\hat{V}^{cT}\|_F \\ &= \sum_{k=5}^{3M} \Sigma_{kk} \end{aligned}$$

where, $U = [\hat{U}|\hat{U}^c]$ and $V = [\hat{V}|\hat{V}^c]$. Thus, the task is now to choose P, X such that $\|x - \hat{P}\hat{X}\|_F$ is minimized subject to the constraint that $\text{rank}(P) \leq 4$. At the end of the first step, we have \hat{P}_j and \hat{X}_j , for every cluster $j = 1, 2, \dots, q$. Our goal is to populate the global image point matrix $x_{3M \times N}$ by using the directly observed image coordinates of the points that are visible and by estimating the coordinates of those points which are otherwise invisible to a given camera. In the second step, estimating invisible points is done by calculating point correspondences using fundamental matrices, trifocal tensors or multifocal tensors as per the situation. Since the first step gives us \hat{P} from which the corresponding camera matrix P can be obtained and the knowledge of the camera matrices is used to generate the fundamental matrix or the trifocal or multifocal tensors required to generate the point correspondences.

Visibility Matrix Θ : While the point correspondences are being computed, a mask $\Theta_{3M \times N}$ is generated whose entries in the i^{th} column is 1 if the tracked feature point is visible to the i^{th} camera, else it is set to 0. This mask Θ is important as it helps us calculate the error metrics that will be used to verify whether or not the iterative procedure is indeed converging and also to help decide when to stop the iteration.

In the third step, the global image point matrix $x_{3M \times N}$ is created using both the directly observable points and the global camera projection matrix is created by stacking individual camera matrices in the order corresponding to that of the estimated invisible points. global image matrix. Once this global image point matrix is obtained, in step 4, step 1 is again computed using the new global image point matrix.

Error Criteria: The error criterion is computed by applying a mask to the regular error minimization constraints that were used to determine \hat{P} and \hat{X} . Thus, we evaluate the Frobenius norm to determine how much error the invisible point estimation carries,

$$\begin{aligned} \|\Theta \cdot (x - \hat{P}\hat{X})\|_F &= \|\Theta \cdot (U\Sigma V^T - \hat{U}\hat{\Sigma}\hat{V}^T)\|_F \\ &= \|\Theta \cdot (\hat{U}^c\hat{\Sigma}^c\hat{V}^{cT})\|_F \end{aligned}$$

Once the Frobenius norm is calculated, the error metric for successive iterations is compared and if it converging, after suitable number of iterations, the process is halted. If the error metric is not

within bounds, the next iteration goes back to step 2 and estimates the invisible points by using the camera matrices obtained from the fresh \hat{P} and \hat{X} evaluated in step 4 of the previous iteration. With these new estimates, the global image coordinate matrix and the global camera matrix are evaluated and step-1 is evaluated to obtain the fresh set of \hat{P} and \hat{X} that will be used in the next iteration. The error metric is calculated and is within bounds or converging, the process is halted, else the iteration goes back to step-2 and runs all over again.

At the end of the last iteration when the error metric is finally found to be converging and well within a certain threshold, the freshly updated \hat{P} and \hat{X} matrices represent the camera matrices and the world coordinates of the N tracked feature points. Thus, employing this iterative procedure will enable us to not only perform self-calibration and obtain the camera parameters of all the M cameras in the network but also obtain the shape information of the 3D object as the N tracked feature points will help determine the shape of the object.

3.7 Generating Manifold

From the AutoCalibration we obtain the projective motion \hat{P} and projective shape \hat{X} which is the 3D coordinates of the corresponding points. We plot those 3D coordinates to obtain the surface of the 3D object as shown in the previous section. But we require more number of 3D points to depict the surface of the 3D object. The 3D points obtained from the auto calibration are not sufficient to generate the surface. To obtain more 3D points we need to generate more number of corresponding points using SURF method as discussed in section 2. We illustrate this by considering two images as shown in the 3.5 below.

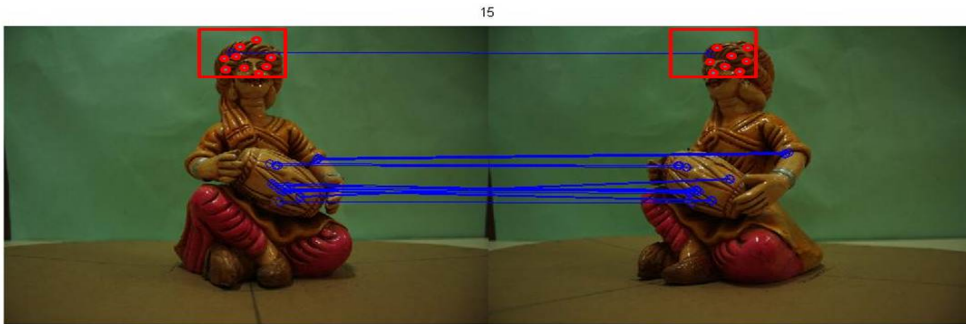


Figure 3.5: Generating more feature points

As shown in the figure we generate more interest points by selecting a neighbourhood around the corresponding points by reducing the filter threshold around the small neighborhood of the corresponding point. We then obtain the corresponding points by using the new interest points by using the SURF technique as discussed in section 2. Figure below shows the more corresponding points generated by reducing the threshold.

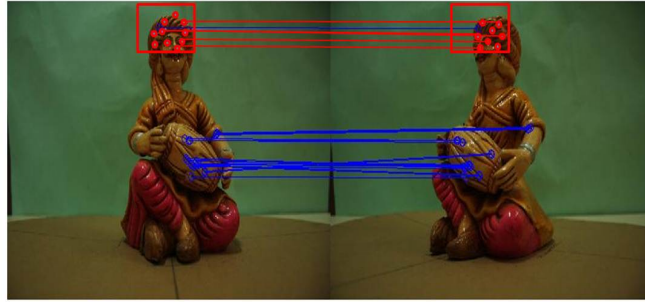


Figure 3.6: Increased Corresponding Points

We now perform the auto calibration method using the new corresponding points and again estimate both Projective motion \hat{P} and Projective shape \hat{X} . We repeat this steps until we get sufficient number of 3D points to get smooth surface of the 3D object.

Chapter 4

Compressive Sensing Framework of 3D Reconstruction

4.1 3D Compression: A Compressive Sensing Framework

From the aforementioned chapter we obtain the manifold of the 3D object.

Compressed sensing(CS) suggests that a signal, sparse in some basis, can be recovered from a small number of random projections. So, the problem in the compressive sensing framework we assume that there exists an orthonormal basis in which the signal on the manifold has a sparse representation. The assumption is valid because most of the natural images have sparse representation when represented in a particular basis. The basis under which the signal on the manifold has a sparse representation has to be constructed. We can use the wavelets [14] that are widely used in the image processing for compression and the Contourlets[15] which have added directionality compared to the wavelets. Assuming that we have a basis under which the signal on the manifold has a sparse representation the 3D reconstruction problem can be stated in the Compressive Sensing domain:

4.1.1 Problem Statement:

To compress the obtained signal say X on the manifold so that we can have minimal representation.

Solution: Construct a basis for the signal on the manifold so that it can have sparse representation in that particular domain.

Mathamatical Formalism: Let X be the signal on the manifold, then let ϕ be the random measurement matrix such that

$$\begin{aligned}\mathbf{X}_d &= \phi\mathbf{X} \\ \implies \mathbf{X}_d &= \phi\psi\hat{\mathbf{X}}\end{aligned}$$

In particular, suppose that there exists an orthonormal basis ψ such that $\mathbf{X} = \psi\hat{\mathbf{X}}$, where $\hat{\mathbf{X}}$ is K -sparse, i.e., the vector $\hat{\mathbf{X}}$ has only K nonzero entries. Then a solution exists if the following

conditions are met:

$$(1 - \epsilon)\|\hat{\mathbf{X}}\|_2 \leq \|\phi\psi\hat{\mathbf{X}}\|_2 \leq (1 + \epsilon)\|\hat{\mathbf{X}}\|_2$$
$$K \leq \text{length}(\phi\psi\hat{\mathbf{X}}) \leq \text{length}(\hat{\mathbf{X}}),$$

where ϵ is a loss factor, usually chosen to be small.

At this point it is difficult to demonstrate the solution to the above problem as we don't have a manifold that accurately depicts the 3D object and also the basis under which the object has a sparse representation has to be constructed. For the basis construction we can use the wavelet transform or the contourlet transform. We show the wavelet decomposition and the contourlet decomposition of the 2D images by which we can infer that the manifold can also have sparse representation in that basis (which has to be constructed).

We consider the analogous problem of compressive image superresolution where we have low resolution image which can be considered as a manifold for the 3D reconstruction problem and a the basis as a wavelet basis (daubechies 8) under which the image has a compact representation and by framing the problem in the compressive sensing domain we obtain a high resolution image (upsampled by a factor of 4) by using greedy algorithms and show that the same can be applied for the 3D Compression.

Chapter 5

Image Super Resolution

To solve the actual 3D problem we require the manifold of the 3D object and a basis on that manifold which we are not able to demonstrate at present. So we consider the analogous problem of 2D image Super Resolution where the main challenge is to recover the high frequency information that was lost in the process of generation of low resolution images. The goal of Image Super Resolution is to recover the missing information in a way that approximates the original high resolution image by posing it in the compressive sensing domain and demonstrate how powerful the compressive sensing can be. The basic idea is that after reconstruction the high resolution image will be sparse in a transform domain and we can therefore use the compressed sensing theory to directly solve for the sparse coefficients from the low-resolution image.

5.1 Super Resolution Problem in Compressive Sensing Domain

The theory of compressive sensing demonstrates how a subsampled signal can be faithfully reconstructed through non-linear optimization techniques as discussed in section 2.4.

5.2 Problem Statement:

Let \mathbf{X} represent the desired high resolution image as an n - dimensional vector $\in \mathbb{R}^n$ and $\tilde{\mathbf{X}} \in \mathbb{R}^m$ represent the low-resolution input. We want to estimate the high-resolution signal from the low resolution input $\tilde{\mathbf{X}} \in \mathbb{R}^m$ where $m \ll n$. We assume that $\tilde{\mathbf{X}}$ has been acquired from the original through a linear downsampling measurement process written as:

$$\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X} \tag{5.1}$$

where \mathbf{S} is a sampling matrix that performs the linear measurements on \mathbf{X} . Initially this seems like an impossible feat since the m samples of $\tilde{\mathbf{X}}$ yield a $(n - m)$ dimensional subspace of possible solutions for the original \mathbf{X} that match our given observations. So we apply a

key assumption of compressed sensing that the transformed version of the signal, $\tilde{\mathbf{X}}$, is k sparse under some basis ψ , meaning that it has at most k non-zero coefficients in that basis i.e; $\|\hat{\mathbf{X}}\|_0 \leq k$. This is not an unreasonable assumption, since we know that high-resolution image will be a “real world image”, and so it will be compressible in a transform domain(wavelet). Now we write 5.1 as:

$$\tilde{\mathbf{X}} = \mathbf{S}\psi\hat{\mathbf{X}} = \Theta\hat{\mathbf{X}} \quad (5.2)$$

where $\Theta = \mathbf{S}\psi$ is a general $m \times n$ measurement matrix. The conditions that have to be satisfied are $m \geq 2k$ and Θ should satisfy the *Restricted Isometry Property*

$$1 - \epsilon < \frac{\|\Theta\hat{\mathbf{X}}\|_2}{\|\hat{\mathbf{X}}\|_2} < 1 + \epsilon \quad (5.3)$$

then we can find the desired $\hat{\mathbf{X}}$ by solving the l_1 optimization problem

$$\min\|\hat{\mathbf{X}}\|_1 \text{ such that } \tilde{\mathbf{X}} = \Theta\hat{\mathbf{X}}. \quad (5.4)$$

This can be done with methods such as *basis pursuit* and *greedy algorithms*[12]. We cannot use 5.2 directly in compressed sensing because they do not meet the 5.3. In order to fulfill the condition, we modify the equation 5.2 by filtering the high-resolution image before downsampling. In other words, we can write our desired high resolution image as \mathbf{X}_s which is filtered by matrix Φ to result in a blurred, high resolution version $\mathbf{X}_b = \Phi\mathbf{X}_s$. This blurred version is then downsampled by equation 5.1:

$$\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X}_b = \mathbf{S}\psi\hat{\mathbf{X}} \quad (5.5)$$

We are using a Gaussian filter as our filter Φ . By expressing the high resolution image in wavelet basis we can modify equation 5.5 as:

$$\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X}_b = \mathbf{S}\psi\hat{\mathbf{X}}_s. \quad (5.6)$$

With this formulation in hand, we can now solve for \mathbf{X}_s by posing it as a compressed sensing problem by assuming that its transform $\hat{\mathbf{X}}_s$ is sparse in the wavelet domain:

$$\min\|\hat{\mathbf{X}}_s\|_1 \text{ such that } \tilde{\mathbf{X}} = \Theta\hat{\mathbf{X}}_s. \quad (5.7)$$

5.3 Solution:

As stated earlier there are greedy methods by which we can approximate a solution to the optimization problem. Given an initial low-resolution image $\tilde{\mathbf{X}}$, we would like to solve for the wavelet transform of the sharp, high-resolution image $\hat{\mathbf{X}}_s$ as in equation 5.7. The idea is once we solve for image $\hat{\mathbf{X}}_s$, we can take its inverse wavelet transform $\psi\hat{\mathbf{X}}_s$ to recover our high-resolution image \mathbf{X}_s . To do this, we use the *Orthogonal Matching Pursuit* greedy algorithm.

Algorithm for Orthogonal Matching Pursuit:

Input parameters: We are given the measurement matrix Θ , the vector $\tilde{\mathbf{X}}$ and the error threshold ϵ_0

Initialization: Initialize $k=0$, and set

- The initial solution $\hat{\mathbf{X}}_s^0 = 0$
- The initial residual $\mathbf{r}^0 = \tilde{\mathbf{X}} - \Theta\hat{\mathbf{X}}_s^0 = \tilde{\mathbf{X}}$
- The initial support $I^0 = \text{support}\{\hat{\mathbf{X}}_s\} = \emptyset$

Main Iteration: Increment k by 1 and perform the following steps:

- **sweep:** Compute the errors $e(j) = \min\|\theta_j z_j - r^{k-1}\|_2^2$ for all j using the optimal choice $z_j^* = \theta_j^T r^{k-1} / \|\theta_j\|_2^2$. where θ_j corresponds to column j of the measurement matrix Θ
- **Update Support:** Find a minimizer j_0 of $e(j) : \forall j \notin I^{k-1}, e(j_0) \leq e(j)$, and update $I^k = I^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\hat{\mathbf{X}}_s^k$, the minimizer of $\|\Theta\hat{\mathbf{X}}_s - \tilde{\mathbf{X}}\|_2^2$ subject to $\text{support}\{\hat{\mathbf{X}}_s\} = I^k$
- **Update Residual:** Compute $\mathbf{r}^k = \tilde{\mathbf{X}} - \Theta\hat{\mathbf{X}}_s^k$
- **Stopping Rule:** If $\|\mathbf{r}^k\| < \epsilon_0$, stop. Otherwise, apply another iteration.

Output: The approximate solution is $\hat{\mathbf{X}}_s^k \approx \hat{\mathbf{X}}_s$ obtained after k iterations.

Once we obtain $\hat{\mathbf{X}}_s$ we can take its inverse wavelet transform $\psi\hat{\mathbf{X}}_s$ to obtain the high-resolution image \mathbf{X}_s .

Chapter 6

Results and Conclusion

6.1 Point Correspondence

1. Pair wise correspondence between the images captured by 2^{nd} and 3^{rd} cameras using SURF feature extraction method

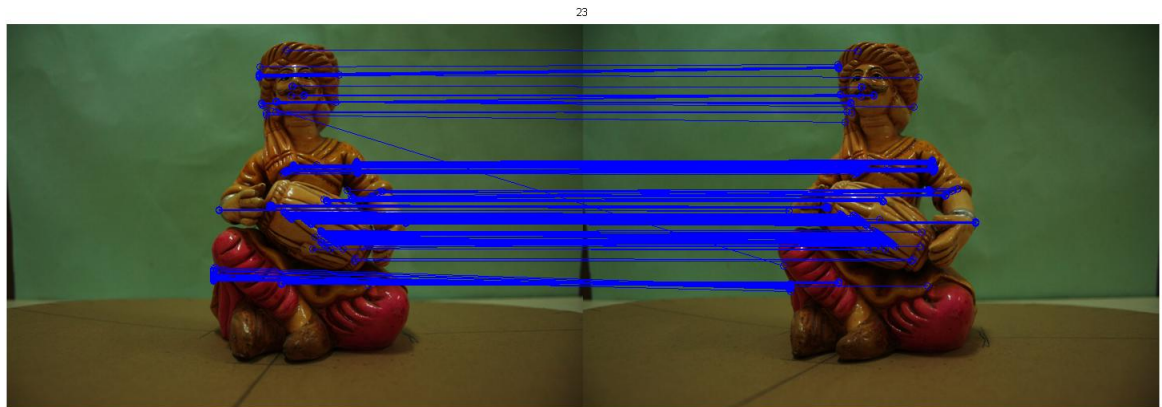


Figure 6.1: Pair wise correspondence between 2^{nd} and 3^{rd} cameras

Point correspondence between the images captured by 7^{th} and 8^{th} cameras using the SURF method is shown in the fig:6.2 below

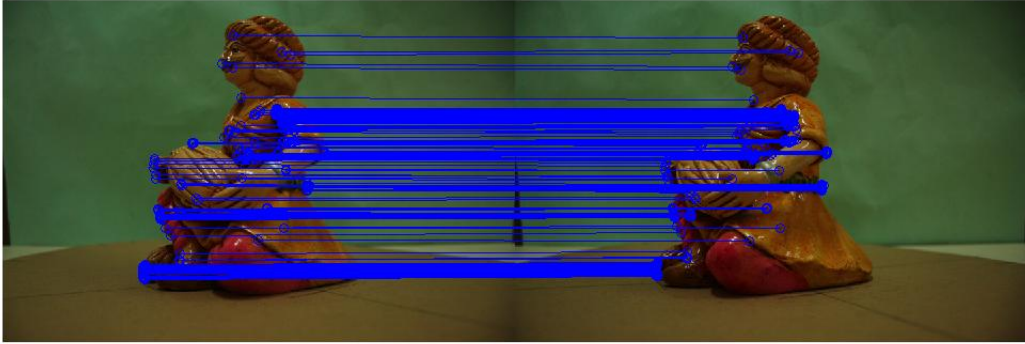


Figure 6.2: Pair wise correspondence between 7th and 8th cameras

2. 4 camera cluster correspondence (points visible to all cameras in cluster using SURF feature extraction method)

Considering the images captured by 4 cameras as a single cluster and finding the point correspondence between them. The corresponding points that we obtain are points that are visible all the 4 cameras in the cluster.

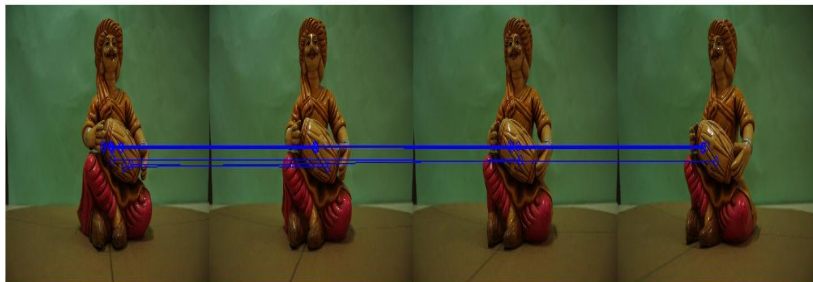


Figure 6.3: Cluster correspondence

6.2 Auto Calibration

6.2.1 Kanade's Factorization

3D reconstruction of the observed feature points using Kanade's Factorization algorithm



Figure 6.4: Reference Object

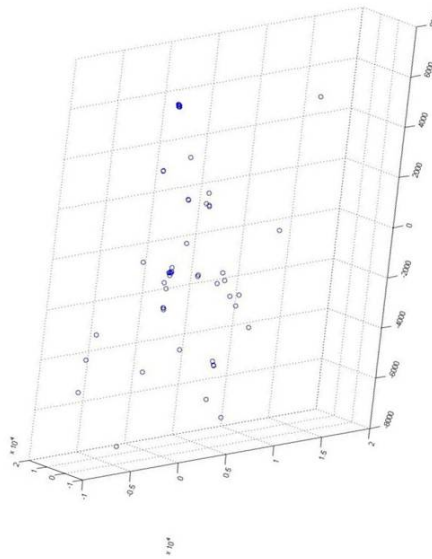


Figure 6.5: Reconstruction 3D points of the object feature points

6.2.2 Proposed method(Notion of Visibility)

3D reconstruction of the observed feature points using proposed method(Visibility Matrix)



Figure 6.6: Reference Object

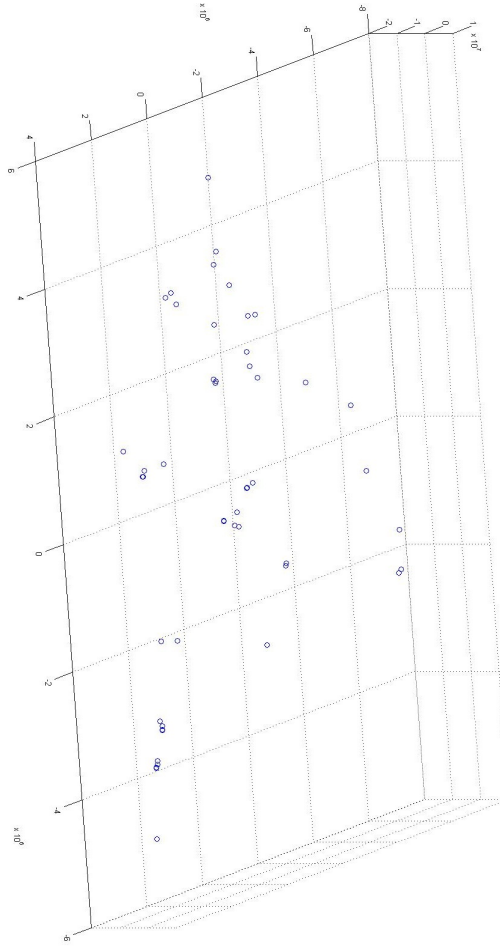


Figure 6.7: Reconstruction 3D points of the observed feature points

6.3 SuperResolution

High-resolution image by using the compressive sensing framework:

1. Consider the low resolution image 6.8 as input and the high-resolution image obtained by upsampling the low-resolution image by a factor of 4 .
2. The High-resolution image obtained by considering a 256×256 low-resolution image as input and upsampling it by a factor of 4.



Figure 6.8: 128×128 low-resolution image



Figure 6.9: 512×512 high-resolution image



Figure 6.10: 256×256 low-resolution image

Sparse Recovery



Figure 6.11: 1024×1024 high-resolution image

Bibliography

- [1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF:Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, 2008 Vol. 110, No. 3, pp. 346-359,
- [2] R. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision," *Cambridge University Press*, Second Edition, March 2004.
- [3] Z.Zhang, "A Flexible New Technique for Camera Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Nov 2000 , vol.22, no.11, pp. 1330- 1334,
- [4] Mei Han, Takeo Kanade, "Creating 3D models with uncalibrated cameras," *IEEE Workshop Applications of Computer Vision*, 2000, vol., no., pp.178-185, 2000.
- [5] R.Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robotics and Automation* vol. 3, no. 4, pp. 323-344, Aug.1987.
- [6] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. Gross, W. Culbertson, T.Malzenbender, " The coliseum immersive teleconferencing system," *In Proceedings of International Workshop on Immersive Telepresence France 2002*.
- [7] Longuet-Higgins,H " A Computer Algorithm for Reconstructing a Scene From Two Projections," *Nature*,1981.
- [8] G. Kurillo, L. Zeyu, R. Bajcsy, "Wide-area external multi-camera calibration using vision graphs and virtual calibration object," *Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008 , pp.1-9, 7-11 Sept. 2008.
- [9] Venkataramani, R.; Bresler, Y.; , "Optimal Sub-Nyquist Nonuniform Sampling and Reconstruction for Multiband signals," *International Conference on Signal Processing*, 2001. *ICIP 98* . , vol.2, no., pp.752-756 vol.2, 4-7 Oct 1998.
- [10] Tsap, L.V.; Goldgof, D.B.; Sarkar, S.; , "Nonrigid motion analysis based on dynamic refinement of finite element models," *Computer Vision and Pattern Recognition*, June 1998. *Proceedings*, vol., no., pp.728-734, 23-25 Jun 1998.
- [11] R. Vasudevan, G. Kurillo, E. Lobaton, T. Bernardin, O. Kreylos, R. Bajcsy, K. Nahrstedt, "High-Quality Visualization for Geographically Distributed 3-D Teleimmersive Applications," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp.573-584, June 2011.

- [12] Michael Elad, *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing*, 1st Edition, Springer, 2010.
- [13] E. J. Candes, M. B. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine* vol.25, no.2, pp.21-30, March 2008.
- [14] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd. Ed., Academic Press, 1998.
- [15] Do, M.N.; Vetterli, M.; , "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol.14, no.12, pp.2091-2106, Dec. 2005
- [16] R. Hartley, "Self-Calibration from Multiple Views with a Rotating Camera," *Proc. Third European Conf. Computer Vision*, pp. 471-478, May 1994.
- [17] R.I. Hartley, "An Algorithm for Self-Calibration from Several Views," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 908-912, June 1994.