Image Superresolution through a Network of Wireless Cameras

IMAGE SUPERRESOLUTION THROUGH A NETWORK OF WIRELESS CAMERAS

By

MARC DIRECTO, B.Eng.Mgmt.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Applied Science

McMaster University

March 2004

MASTER OF APPLIED SCIENCE (2004) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: Image Superresolution through a Network of Wireless Cameras

AUTHOR: Marc Directo

B.Eng.Mgmt.

SUPERVISOR: David Capson, Shahram Shirani

NUMBER OF PAGES: xi, 72

Abstract

This thesis outlines a multiple-camera wireless image superresolution system which uses off-the-shelf components. The system presented demonstrates the reconstruction of a high-resolution image from multiple low-resolution images acquired from different wireless camera nodes. Each camera node participating in the system consists of a dedicated camera for image acquisition as well as a Bluetooth USB communications card for wireless transmission of data. Low-resolution images are captured at these nodes and are transmitted to the central vision server, where they are processed and registered onto a common projective plane. The registration process is arrived at through the RANdom SAmple Consensus (RANSAC) algorithm. Once the set of low-resolution images has been registered, a single high-resolution image is reconstructed. The super-resolution process used to obtain the high-resolution output is the Projection Onto Convex Sets (POCS) technique. Reconstruction results are presented.

Acknowledgements

I would like to thank Dr. David Capson and Dr. Shahram Shirani, my supervisors, for the opportunity they gave me and their patience and support. Their guidance throughout my research was invaluable. In addition I would like to thank the support staff of ECE. I would also like to thank Phil, Jeff, Derek, Abbas, Alireza, In Su, Steve, Phil and Kaveh who kept me company in the Machine Vision and Image Processing Lab. I'd like to thank my siblings, Michelle, Michael and Melanie, as well as my parents, for believing in me. I owe special thanks to the Schrattenholzer family for their care and hospitality. To my best friends and fellow grad students, Amie and James, thank you for sharing the grad/life experience with me. To anyone else who had patience with me while I completed my work I would like to express my gratitude. Last but not least, I would like to say thank you to Sandra, for keeping me grounded.

List of Abbreviations

SR	superresolution
hr	high-resolution image
lr	low-resolution image
L	number of low-resolution observations
$\mathbf{H}$	2D homography relating $\mathbf{x}$ and $\mathbf{x}'$ in the equation $\mathbf{H}\mathbf{x}_i = \mathbf{x}'_i$
P, P'	sets of feature points in images being registered
CM	candidate match list for image correspondence
Q, Q'	sets of putative correspondence points for images being registered
S, S'	sets of inliers for images being registered
O, O'	sets of outliers for images being registered
PSF	Point Spread Function
DLT	Direct Linear Transform
POCS	Projection onto Convex Sets
RANSAC	Random Sample Consensus
RTC pkt	Request-to-Connect packet
ACK pkt	Acknowledgment packet

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Superresolution

Digital imaging systems record incoming light rays onto an electronic sensor, typically a Complementary Oxide Semiconductor (CMOS) or Charged Coupled Device (CCD) sensor, to create an image. The spatial resolution of these sensors is fixed, creating a problem when an image with a higher spatial resolution is required (throughout this thesis, resolution is taken as spatial resolution). In order to increase the resolution of the image two solutions emerge; increase the image sensor size or interpolate between the pixels of the low resolution image to arrive at a high resolution estimate. These two alternatives are unattractive because larger sensors incur higher costs and obtaining a high resolution estimate from a single low resolution image is a highly ill-posed problem. Superresolution techniques are a way of overcoming these problems.

Superresolution (SR) is the process of reconstructing an image or obtaining video which is at a higher resolution than that of the capturing sensor. The idea behind the process is that a series of low-resolution (lr) images are available, which relate to an underlying high-resolution (hr) image. These low-resolution images can be obtained from existing inexpensive low-cost sensors or a sequence of video frames. As long as

the low-resolution observations are related by sub-pixel shifts, new information can be gleaned and processed to come up with a high resolution estimate.

High-resolution image reconstruction techniques [6][11] show promise in the application domain as low-cost, low-resolution imaging becomes increasingly more prevalent in today's consumer devices. This is evident from the integration of an imaging component with an increasing number of consumer devices such as cellular phones, PDA's, notebook computers, etc. At the root of harnessing these under-utilized imaging resources is the problem of organizing these devices into an intelligent network that can take these low-resolution observations and produce the desired high-resolution output. Wireless networking technologies give us a platform on which to implement such a system.

1.2 Review of Literature

Increasing the spatial resolution of an image can be achieved through single frame interpolation and restoration methods. It is known that this method is highly ill-posed and so multi-frame methods are more viable. A number of multi-frame SR techniques have been used.

Frequency Domain Tsai and Huang [10] proposed a frequency domain approach which worked on the principles of the spatial aliasing effect. The relationship between aliased discrete Fourier transform coefficients of observed low-resolution images and the continuous Fourier transform coefficients of the desired high-resolution image was exploited. This method only considered the case of global motion, and assumed the original *hr* image was bandlimited. Kim et al. [26] extended this technique to include sensor blur and noise. Varying blurs in each image was presented by Kim and Su [22].

Regularized Reconstruction Deterministic and stochastic regularization approaches exist for SR image reconstruction. With an insufficient number of *lr* images, the SR problem becomes problematic. Constrained least squares (CLS) and maximum a posteriori (MAP) techniques are among some of the regularization techniques. CLS can be arrived at through a Lagrangian decomposition of the problem [12]. The regularization parameter controls the tradeoff between the reconstruction results and the smoothness of the image. A deterministic iterative technique is then used to estimate the high resolution image. Bayesian techniques are used when the a posteriori probability density function (PDF) can be found. The MAP estimator maximizes the a posteriori PDF. Hardie et al. [2] used a MAP framework to estimate both the registration parameters and *hr* image. Cheeseman et al. [9] used a Bayesian estimation technique to reconstruct satellite images of planets.

Iterative Back-Projection An iterative back projection technique was formulated by Irani and Peleg [17]. Here, the high-resolution image is estimated by back projecting the difference between simulated low-resolution images and the actual low-resolution images.

Adaptive Filtering An adaptive filtering approach was taken by Elad and Feuer [7]. They proposed using least squares estimators based on adaptive filters, with a least means squares or recursive least squares algorithm. The *hr* image is estimated at each time using an iterative steepest descent method. It can handle linear space and time-varying blurring and arbitrary motion, which are both assumed known.

Projection Onto Convex Sets The method used in this thesis is the projection onto convex sets (POCS) approach. Stark and Oskoui [16] first suggested this recursive technique where the unknown image is restricted to the intersection of convex

sets that satisfy certain constraints. In the event of a nonempty intersection, a solution can be arrived at by alternating projections onto these convex sets. Tekalp et al. [18] [19] included observation noise in their formulation. Patti et al. [23] took into account the presence of sensor blurring. In another instance, Patti et al. [24] proposed a model to deal with the motion blur from nonzero aperture time. Patti and Altunbasak [1] allowed higher order interpolation methods to be used by improving the discretization process as well as reducing the amount of edge ringing present by modifying the constraint sets.

1.3 Approach

The goal of the research documented in this thesis is to use multiple cameras to view a scene, and by combining their mutual information, reconstruct a higher resolution image of the scene. One way to achieve this, without a calibrated setup of cameras, is to limit the type of image transformation to a planar homography. An example of a planar homography is the situation whereby a series of images are acquired from a single camera by rotating it around its centre. These captured images are related by planar homographies. Another case is the situation of two cameras with arbitrary viewing angles imaging a planar surface. The resulting images are also related by a planar homography.

It is the latter case which will form the basis of this work. The system consists of outlying camera nodes which take low-resolution images, and a server node which accepts the images and produces the high resolution estimate. The major steps are shown in figure 1.1 and outlined below :

Image Acquisition A series of low-resolution images, $(lr_0, \ldots, lr_{L-1})$, are acquired from client nodes. The images are transferred over a wireless connection to the server node.

Image Registration The Harris corner detector [20] is applied to each image and an initial set of correspondence matches are made. The Random Sample Consensus (RANSAC) algorithm [4] is then applied to iteratively find a set of inliers. The Direct Linear Transform (DLT) [28] is used to find the homography H which transforms the image for superresolution.

Image Superresolution The POCS algorithm is used to superresolve the images.

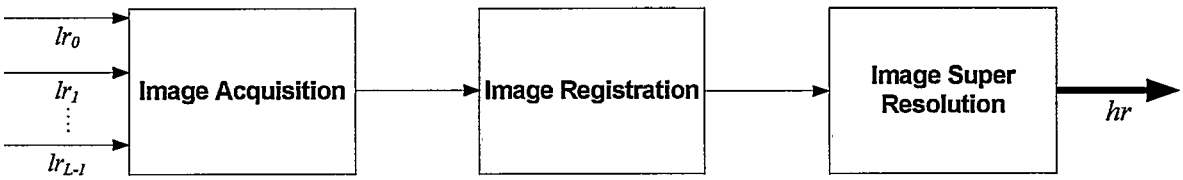


Figure 1.1: System Block Diagram

1.4 Organization Of Thesis

Chapter 2 discusses the principles behind the superresolution technique implemented. Algorithms for finding the correspondences between images and registering them are given. The superresolution technique of Projection onto Convex Sets is also introduced.

Chapter 3 outlines the architecture of the vision system. The short range wireless scheme is introduced. Communication setup, methods and protocols are described.

Chapter 4 gives the experimental results of the POCS algorithm as well as the vision system reconstruction results.

Chapter 5 discusses the results of the experiment and conclude with comments on future work.

Chapter 2

Image Superresolution

This chapter explores and analyzes the process of creating a superresolved image from a series of low spatially resolved images. The resolution gained from multi-image superresolution techniques is possible because of the additional information introduced by several low-resolution images. The low-resolution images to be captured will first be registered onto a common view. Once this has been achieved, the superresolution algorithm can be implemented to produce the high-resolution image estimate.

2.1 Image Registration

The image acquisition model used assumes that the images are related by a 2D planar homography. That is to say, given a set of points $\mathbf{x}_i$ in an image I_1, and a corresponding set of points $\mathbf{x}'_i$ in an image I_2, we must find the projective transformation which maps $\mathbf{x}_i$ to $\mathbf{x}'_i$, for all i. This projective transform is known as a 2D homography and the rest of this section will outline in detail how to find this homography.

Figure 2.1(a) shows two different cameras viewing a planar surface and the resulting image points formed on their respective image planes. These images are related by a homography. Figure 2.1(b) depicts the case of acquiring images by rotating a

camera around its centre resulting in two different image planes that are also related by a homography. The case which is of interest is the situation given in figure 2.1(a). To reduce scaling issues, the camera centres are taken to lie on a common line, parallel to the planar scene. One low-resolution image is produced for each camera introduced into the system. These low-resolution images, once warped to a common view, will be the underlying images for our high resolution estimate. The first image in a sequence of low resolution images is defined as lr_0, the base image, and each subsequent image lr_i, is registered onto the base image. For illustrative purposes, the remainder of section 2.1 deals with the general case of finding the homography H for an arbitrary projective transform from one of our low resolution images lr_i to the base image lr_0.

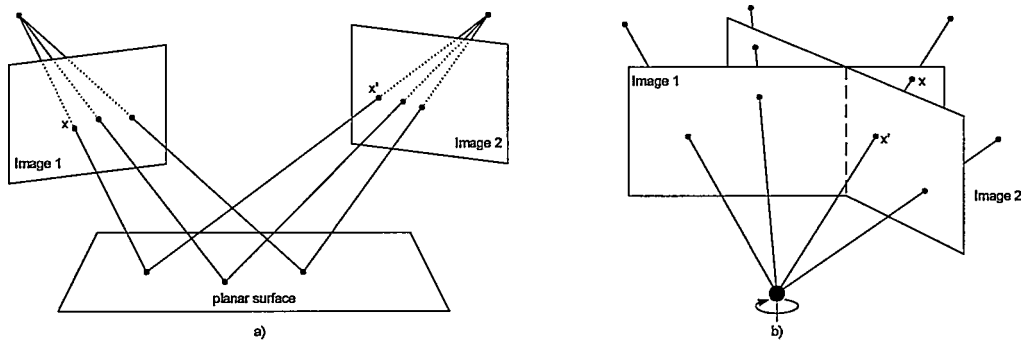


Figure 2.1: Examples of Homographies

2.1.1 Point Correspondence

The first step in registering the images is to compute feature points in both images. The hope is that some of the feature points constitute correspondences between the images. This stage is not meant to produce a perfect set of correspondence points, but rather to come up with a rough set from which only good correspondence points will be taken. These initial correspondences are found using a proximity and neighbourhood intensity metric.

Harris Feature Detector

To find feature points in an image, the method proposed in [20] is used. A local moment matrix M is formulated from the image gradients as

$$\mathrm{M} = \begin{pmatrix} (\frac{\partial I}{\partial m_1})^2 & (\frac{\partial I}{\partial m_1})(\frac{\partial I}{\partial m_2}) \\ (\frac{\partial I}{\partial m_1})(\frac{\partial I}{\partial m_2}) & (\frac{\partial I}{\partial m_2})^2 \end{pmatrix} \tag{2.1}$$

where $I(m_1, m_2)$ is the grey level at pixel (m_1, m_2). The feature response function is then given by

$$R = \det(\mathrm{M}) - k(\mathrm{tr}(\mathrm{M}))^2. \tag{2.2}$$

An alternative feature response function is given in [15] as

$$R_{alt} = \frac{\det(\mathrm{M})}{\mathrm{tr}(\mathrm{M}) + \epsilon} \tag{2.3}$$

where ϵ is the machine error (distance from 1.0 to the next largest floating point number). In the calculation of the Harris feature points, the alternative feature response function R_{alt} is used. Those values which are local maxima at (m_1, m_2) of R are defined as features. Then

$$P = \{p_a\}, \quad \forall a = 0 \ldots (A_0 - 1), \text{ where } p_a = (m_1, m_2) \tag{2.4}$$

$$P' = \{p'_{a'}\}, \quad \forall a' = 0 \ldots (A_i - 1), \text{ where } p'_{a'} = (m_1, m_2) \tag{2.5}$$

are the feature point sets from lr_0 and lr_i respectively.

Putative Correspondences

With points of interest found in both images, the task now becomes one of finding potential correspondence points between them. That is to say, find the features points in P that correspond to the same feature points in P'.

One of the feature points found in lr_0 is labelled $p_\triangle$. If we define W as a square search region centred at $p_\triangle$ in lr_i, with dimensions $(2w+1) \times (2w+1)$, then the list

of candidate matches for $p_\triangle$ will be those feature points p'_∇ found within W in lr_i or

$$CM(p_\triangle) = \left\{ \{p'_\nabla\}, \forall \nabla = 0 \ldots (A_i - 1) \;\middle|\; \begin{array}{c} m_{1,p_\triangle} - w \leq m_{1,p'_\nabla} \leq m_{1,p_\triangle} + w \\ \text{and} \\ m_{2,p_\triangle} - w \leq m_{2,p'_\nabla} \leq m_{2,p_\triangle} + w \end{array} \right\} \tag{2.6}$$

Each feature point p'_∇ in CM is then compared to $p_\triangle$ using the cross-correlation function of the n-pixel neighbourhood surrounding both $p_\triangle$ and p'_∇. The highest thresholded value is taken as a match. This brute force method of comparing each feature point found in CM is repeated for each p_j in lr_0. In the event that a feature in lr_i is claimed by more than one feature in lr_0, the feature which produced the highest cross-correlation takes precedence. The entire process is repeated with the roles of lr_0 and lr_i reversed.

To illustrate the process an example is given in figure 2.2. The two images lr_0 and lr_i are defined on a 10×10 grid with the image of a square filled in black. The square in image lr_i is shifted by 1 pixel to the right, and 2 pixels down from the square in image lr_0. The features detected in lr_0 are the corners of the black square and are denoted by upright triangles. Similarly, upside down triangles signify features found in lr_i. If we take the feature point located at (4,5) in lr_0, then W ($w = 7$) is the window which surrounds those pixels centred at (4,5) in lr_i (the thick dashed line). The candidate match list is then $CM = \{(6,6),(6,8)\}$. The thin dashed lines surrounding (4,5) in lr_0 and (6,6) in lr_i are the n-pixel neighbourhoods from which the cross-correlation will be computed from. In this case $n = 8$. The cross-correlation factor which is the highest and over a certain threshold is deemed as a corresponding point.

If a correspondence is found from $p_\triangle$ to p'_∇, it is recorded in the sets

$$Q = \{q_i\}, \quad \forall i = 0 \ldots (B-1), q_i = (m_1, m_2) \tag{2.7}$$

$$Q' = \{q'_i\}, \quad \forall i = 0 \ldots (B-1), q'_i = (m_1, m_2) \tag{2.8}$$

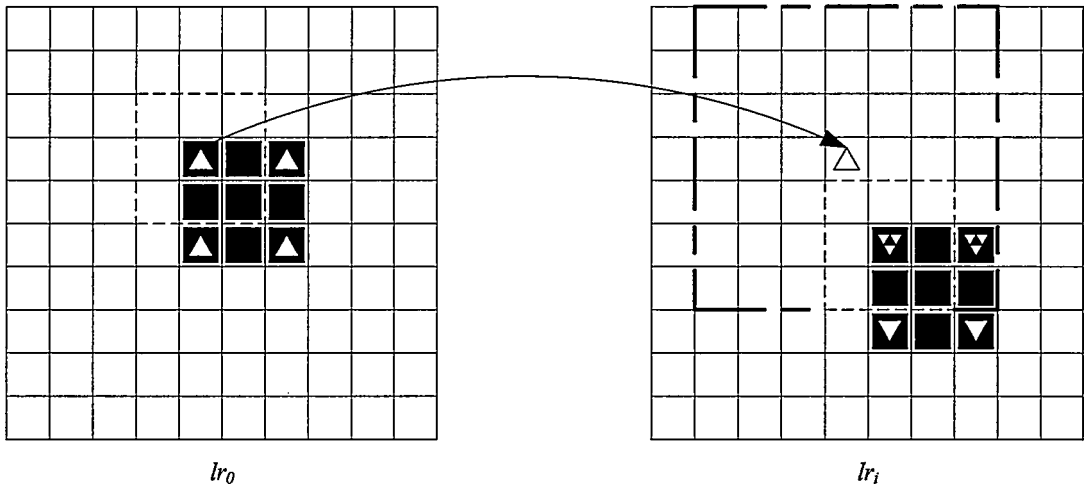


Figure 2.2: Finding Correspondences

Where Q are the points from P and Q' are the points from P'. These constitute our putative correspondences. It is of import to note that these putative matches need not be perfect. It will be the task of the RANSAC algorithm to find those matches that are good and discard mismatches. In order to properly classify inlying data it is necessary to review how a homography between two images is found.

2.1.2 Projective Geometry

Homogeneous Points and Lines

In the discussion of projective geometry, it is beneficial that the concept of homogeneous coordinates is introduced.

The registration process involves representing a Euclidean point (x, y) in the projective plane. This can be done by adding a third coordinate of 1 to the Euclidean pair. The point (x, y) in Euclidean space becomes $(x, y, 1)$ in homogenous coordinates of the projective space $\mathbb{P}^2$. This homogenous coordinate can be thought of as a vector in $\mathbb{R}^3$ that represents the Euclidean point in $\mathbb{R}^2$. In fact, this third coordinate is

arbitrary since it is a vector representation. To summarize

$$(x, y) \leftrightarrow (x, y, 1) = (zx, zy, z). \tag{2.9}$$

The origin ($z = 0$) is excluded in this definition. Then the projective space $\mathbb{P}^2$ is the set of homogeneous vectors in $\mathbb{R}^3$ (excluding $(0, 0, 0)^T$).

To move from the projective space to Euclidean space, the first two coordinates are divided through by the third coordinate :

$$(x, y, z) \leftrightarrow \left(\frac{x}{z}, \frac{y}{z}\right) \tag{2.10}$$

To summarize as in figure 2.3, points and lines in $\mathbb{P}^2$ are represented by lines and planes in $\mathbb{R}^3$.

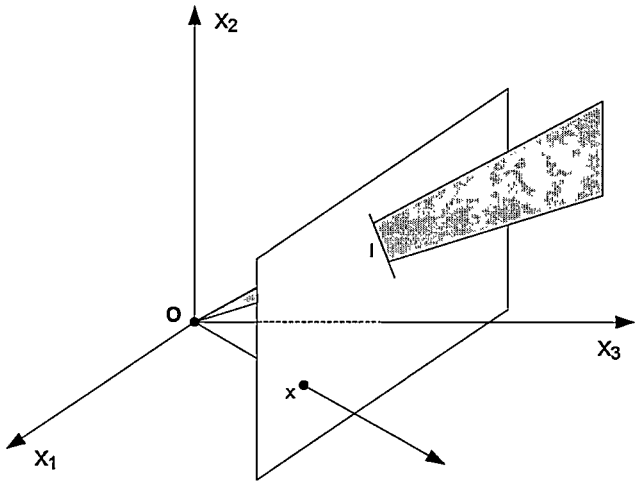


Figure 2.3: **Model of the Projective Space.** *Points (x) and lines (l) in* $\mathbb{P}^2$ *are represented by lines and planes, respectively, through the origin in* $\mathbb{R}^3$.

2D Homography

Given the set of correspondence points q_i from lr_0, the homogeneous coordinates $\mathbf{x}_i$ can be produced. In a similar fashion the points $\mathbf{x'}_i$ are the homogeneous coordinates

of q_i', the correspondence points in lr_i. The goal is to compute the 3×3 matrix $\mathbf{H}$ such that :

$$\mathbf{H}\mathbf{x}_i = \mathbf{x}_i' \quad \forall i \tag{2.11}$$

The number of correspondence points required to find the matrix $\mathbf{H}$ can be found by analyzing the components of $\mathbf{H}$. $\mathbf{H}$ has 9 entries but in equation 2.11, the equality is only up to scale. This leaves 8 degrees of freedom. Since each correspondence accounts for 2 degrees of freedom (one in each of the x and y directions), the minimum number of point correspondences necessary is 4.

If a minimum number of correspondence points exist, a simple linear method of finding $\mathbf{H}$ is given by the DLT algorithm [28]. Recall from equation 2.11, that the homography $\mathbf{H}$ relates the points $\mathbf{x}_i$ in one image, to the points $\mathbf{x'}_i$ in another. Since we are working in the projective space $\mathbb{P}^2$, we use the homogeneous points $\mathbf{x}_i$ and $\mathbf{x'}_i$. In fact, equation 2.11 is a vector equality and so the left and right sides need only have the same direction and may differ by a non-zero scaling factor. Rewriting equation 2.11 in the form of a vector cross product yields

$$\mathbf{x'}_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0} \tag{2.12}$$

$$\begin{aligned}
\mathbf{H}\mathbf{x}_i &= \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \\
&= \begin{pmatrix} h_1 x_i + h_2 y_i + h_3 z_i \\ h_4 x_i + h_5 y_i + h_6 z_i \\ h_7 x_i + h_8 y_i + h_9 z_i \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{h}^{1\mathsf{T}} \mathbf{x}_i \\ \mathbf{h}^{2\mathsf{T}} \mathbf{x}_i \\ \mathbf{h}^{3\mathsf{T}} \mathbf{x}_i \end{pmatrix}
\end{aligned} \tag{2.13}$$

Letting $\mathbf{h}^{j\top}$ represent the j-th row of matrix $\mathbf{H}$. Rewriting $\mathbf{x'}_i = (x'_i, y'_i, z'_i)^\top$, the cross product from equation 2.12 becomes

$$\begin{aligned}
\mathbf{x'}_i \times \mathbf{H}\mathbf{x}_i &= \begin{vmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \\ x'_i & y'_i & z'_i \\ \mathbf{h}^{1\top}\mathbf{x}_i & \mathbf{h}^{2\top}\mathbf{x}_i & \mathbf{h}^{3\top}\mathbf{x}_i \end{vmatrix} \\
&= \begin{pmatrix} y'_i\mathbf{h}^{3\top}\mathbf{x}_i - z'_i\mathbf{h}^{2\top}\mathbf{x}_i \\ z'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\top}\mathbf{x}_i - y'_i\mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix} \\
&= \begin{pmatrix} -z'_i\mathbf{h}^{2\top}\mathbf{x}_i + y'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ z'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ -y'_i\mathbf{h}^{1T\top}\mathbf{x}_i + x'_i\mathbf{h}^{2\top}\mathbf{x}_i \end{pmatrix}
\end{aligned}$$

Using $\mathbf{h}^{j\top}\mathbf{x}_i = \mathbf{x}_i^\top\mathbf{h}^j$, the above separates into three equations in $\mathbf{H}$.

$$\begin{aligned}
\mathbf{x'}_i \times \mathbf{H}\mathbf{x}_i &= \begin{pmatrix} -z'_i\mathbf{x}_i^\top\mathbf{h}^2 + y'_i\mathbf{x}_i^\top\mathbf{h}^3 \\ z'_i\mathbf{x}_i^\top\mathbf{h}^1 - x'_i\mathbf{x}_i^\top\mathbf{h}^3 \\ -y'_i\mathbf{x}_i^\top\mathbf{h}^1 + x'_i\mathbf{x}_i^\top\mathbf{h}^2 \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{0}^\top - z'_i\mathbf{x}_i^\top\mathbf{h}^2 + y'_i\mathbf{x}_i^\top\mathbf{h}^3 \\ z'_i\mathbf{x}_i^\top\mathbf{h}^1 + \mathbf{0}^\top - x'_i\mathbf{x}_i^\top\mathbf{h}^3 \\ -y'_i\mathbf{x}_i^\top\mathbf{h}^1 + x'_i\mathbf{x}_i^\top\mathbf{h}^2 + \mathbf{0}^\top \end{pmatrix}
\end{aligned}$$

$$\begin{bmatrix} \mathbf{0}^\top & -z'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ z'_i\mathbf{x}_i^\top & \mathbf{0}^\top & x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{2.14}$$

This equation has the form $\mathbf{G}_i\mathbf{h} = \mathbf{0}$, which is linear in $\mathbf{h}$. It is important to note that the third row is not linearly independent (it is the sum of x'_i times the first row and y'_i times the second). The third row can be omitted for simplicity and the equation

for one point correspondence can be written as

$$\begin{bmatrix} \mathbf{0}^\top & -z'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ z'_i\mathbf{x}_i^\top & \mathbf{0}^\top & x'_i\mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{2.15}$$

Thus, each point correspondence yields two equations in $\mathbf{H}$. This reconfirms that each point correspondence has two degrees of freedom and accounts for two entries in $\mathbf{H}$. Four point correspondences produces $\mathbf{Gh} = \mathbf{0}$, where $\mathbf{G}$ comprises the equations we get from each of the four point correspondences and has dimensions 8×9. Since $\mathbf{H}$ is defined only up to scale, the solution $\mathbf{h}$ is sufficient and a scale may be arbitrarily chosen by imposing $||\mathbf{h}|| = 1$.

In the case where there are $n \geq 4$ point correspondences, the system becomes over-determined. If the coordinates of the correspondence points are exact, an exact solution apart from the trivial solution will be found. However, in most real world cases, there is measurement noise that will prohibit an exact solution from being discovered. In this case, *only* the trivial solution will yield an exact solution which is of no interest. The problem then becomes one of finding a vector $\mathbf{h}$ that minimizes the quotient $\frac{||\mathbf{Gh}||}{||\mathbf{h}||}$. The solution is the unit eigenvector of $\mathbf{G}^\top\mathbf{G}$ with the smallest eigenvalue or the unit singular vector corresponding to the smallest singular value of $\mathbf{G}$. This algorithm is known as the DLT and is given in algorithm 1.

Algorithm 1 DLT Algorithm

Input: $n \geq 4$ 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$

1. For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ compute $\mathbf{G}_i$ from equation 2.15.
2. Assemble the n 2×9 matrices $\mathbf{G}_i$ into a single $2n \times 9$ matrix $\mathbf{G}$.
3. Obtain the SVD of $\mathbf{G}$. If $A = UDV^\top$ with $\mathbf{G}$ diagonal with positive diagonal entries, arranged in descending order down the diagonal, then $\mathbf{h}$ is the last column of V.

2.1.3 RANSAC Algorithm

With the information needed to find **H** (the correspondence points), and the method by which to calculate **H** (the DLT), a robust estimation of **H** can be found. In finding **H** using the DLT, the assumption of only one type of error, the measurement noise, was made. The other error that is of concern is the possibility that a correspondence could be wrong. These incorrect correspondences are deemed *outliers* to the data and can have a detrimental effect on the estimation of **H**. It is important that these outliers are found and discarded, and that only *inliers* are used in our calculations.

The robust estimator we will be using is called the RANdom SAmple Concensus (RANSAC) algorithm [4][27]. It is especially good at dealing with a large proportion of outliers. The idea behind RANSAC is to use a minimal set of data initially, and when presented with the opportunity, expand on the set with good data.

As shown earlier in the case of a 2D homography, the minimum set required is 4 point correspondences. Taking a random sample of 4 correspondence points, the putative point correspondence sets Q and Q_i, we can calculate an **H**. An adequate random sample is one that is not degenerate and has a good spatial distribution over the image. If three of the four points in an image are collinear, then a homography cannot be determined. In addition, points chosen over a more spatially distributed area will exhibit better results because of the local nature of the projective transform. This can be implemented by tiling the image and choosing points in differing tiles by weighting the random number generator. The support for this **H** can be found by applying the homography to the set and determining the number of point correspondences that lie within a certain distance threshold to it. Those points that are within the threshold t are deemed inliers. An important question becomes what distance threshold to use?

Distance threshold

This distance t is chosen such that with probability α, the point correspondence is an inlier to the data. If the assumption is made that the measurement error is Gaussian with zero mean and standard deviation σ, the value of the perpendicular distance d can be estimated with a χ^2_m variable with m degrees of freedom (a sum of squared Gaussian variables). The cumulative distribution of a χ^2_m variable is $F_m(k^2) = \int_0^{k^2} \chi^2_m(\xi)d\xi$. This yields the probability that the value of a χ^2_m random variable will be less than k^2. From [28] the following result emerges

$$q \overset{\mathbf{H}}{\leftrightarrow} q' \begin{cases} \textit{inlier} & d^2 < t^2 \\ \textit{outlier} & d^2 \geq t^2 \end{cases} \quad \text{with } t^2 = F_m^{-1}(\alpha)\sigma^2 \tag{2.16}$$

With $\alpha = 0.95$, then from [28] the value of t^2 for a homography is $5.99\sigma^2$.

The output of the RANSAC algorithm is an initial estimate of $\mathbf{H}$ as well as the sets

$$S = \{\mathbf{x}_i\} \text{ if } \{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\} \text{ is classified an inlier} \tag{2.17}$$

$$S' = \{\mathbf{x}'_i\} \text{ if } \{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\} \text{ is classified an inlier} \tag{2.18}$$

$$O = P - S \tag{2.19}$$

$$O' = P' - S' \tag{2.20}$$

2.1.4 Guided Matching

With an accurate estimate of $\mathbf{H}$ the list of inliers can be augmented and in turn a more refined estimate of $\mathbf{H}$ achieved. This technique is similar to the image correspondence step described earlier in this chapter and makes a more refined search over a smaller number of points, namely the outliers of the data. If one of the classified outlying

Algorithm 2 RANSAC Algorithm

Input: 2 images with putative set of point correspondences
Initialize: $N = \infty$ and *sample_count* $= 0$
while $N >$ sample_count **do**

1. Select an adequate random sample of 4 correspondences and compute $\mathbf{H}$ using the DLT.
2. Calculate the distance d for each putative correspondence.
3. Compute the number of inliers consistent with $\mathbf{H}$ by the number of correspondences for which $d^2 < t^2 = 5.99\sigma^2$ pixels.
4. $e = 1 -$ (number of inliers)/(total number of correspondences)
5. $N = \log(1-p)/\log(1-(1-e)^4)$
6. sample_count++

end while

feature points in O is considered, one can predict where this feature point $\mathbf{x}_j$ should map to in lr_i using $\mathbf{H}\mathbf{x}_j = \widehat{\mathbf{x}}_j$. By searching a small window centred around $\widehat{\mathbf{x}}_j$ in lr_i, it is possible to find feature points in O' that would be potential matches. The same auto-correlation function is used to determine if any points form a correspondence pair with $\mathbf{x}_j$. Each point in O is tested and the new inlying correspondence features are added to S and S' and subtracted from O and O'. Similarly, the same reasoning can be applied to points $\mathbf{x'}_j$ in O' which should map to lr_0 using $\widehat{\mathbf{x}'}_j = \mathbf{H}^{-1}\mathbf{x'}_j$. The necessary adjustments are again made to S, S', O and O'. The DLT is used again with S and S' to estimate $\mathbf{H}$, and the guided matching process is repeated, until the size of S and S' stabilizes. Applying the homography $\mathbf{H}$ to lr_i is the final step in the registration process. The image is now ready to be passed to the superresolution algorithm.

2.2 Superresolution

Once the image registration process is complete, the high-resolution estimate can be formed using the POCS algorithm for each of the L low-resolution images.

2.2.1 Projection Onto Convex Sets

In the POCS formulation [21] [25], an unknown signal s, corresponding to a measured signal x, is to be found. It is assumed that s is part of an appropriate Hilbert space $\mathcal{H}$. With k pieces of a priori information and constraints, there are k corresponding closed convex sets

$$C_i \in \mathcal{H}, \quad i = 1, 2, \ldots, k. \tag{2.21}$$

The unknown signal s will be found in the set

$$C_0 \doteq \bigcap_{i=1}^{k} C_i \tag{2.22}$$

providing C_0 exists. The goal is to find the closest point s to our measured signal x in C_0. Associated with each convex set C_i is a projection operator P_i. The composition of each of the projections results in the composition operator

$$P = P_k P_{k-1} \ldots P_1. \tag{2.23}$$

The recursive update on s is

$$s^{[j+1]} = P s^{[j]}. \tag{2.24}$$

A relaxed projection operator can be introduced to improve the convergence rate. This relaxed projection operator extends the projection beyond the boundaries of the sets and is given by

$$T_i = 1 + \lambda_i (P_i - 1) \quad i = 1, 2, \ldots, m, \tag{2.25}$$

leading to the composition operator

$$T = T_m T_{m-1} \cdots T_1, \tag{2.26}$$

and the more general recursive update

$$s^{[k+1]} = T s^{[k]}. \tag{2.27}$$

The initial estimate s^0 can be chosen from $\mathcal{H}$. An example of this can be seen in figure 2.4.

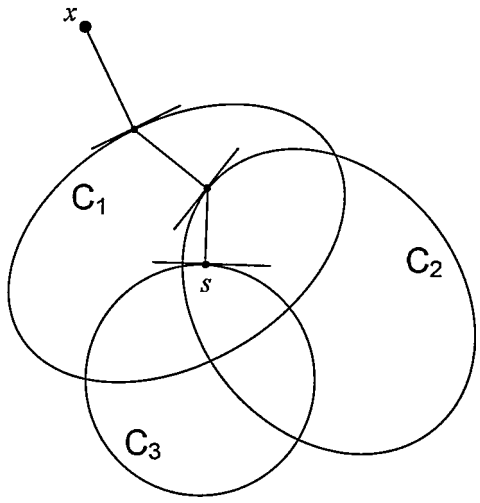


Figure 2.4: **POCS Example** - *Successive projections onto convex sets.*

2.2.2 Image Capture Process

The image capture process can be modelled by a linear shift varying (LSV) process that includes camera and scene object motion, blur and low-resolution sampling. The sensor aperture time is taken to be zero. The mapping from a high-resolution image to a low-resolution image is arrived at by computing a weighted sum of the high-resolution image pixels, whose weights come from the space-invariant point spread function (PSF). The centre of the PSF is found through the motion vector that

relates the low-resolution image to the high-resolution image. Defining this PSF as $h(m_1, m_2, k; n_1, n_2)$ we can summarize the image capture process as

$$lr_k(m_1, m_2) = \sum_{n_1} \sum_{n_2} h(m_1, m_2, k; n_1, n_2) hr(n_1, n_2). \tag{2.28}$$

Figure 2.5 depicts the image capture process.

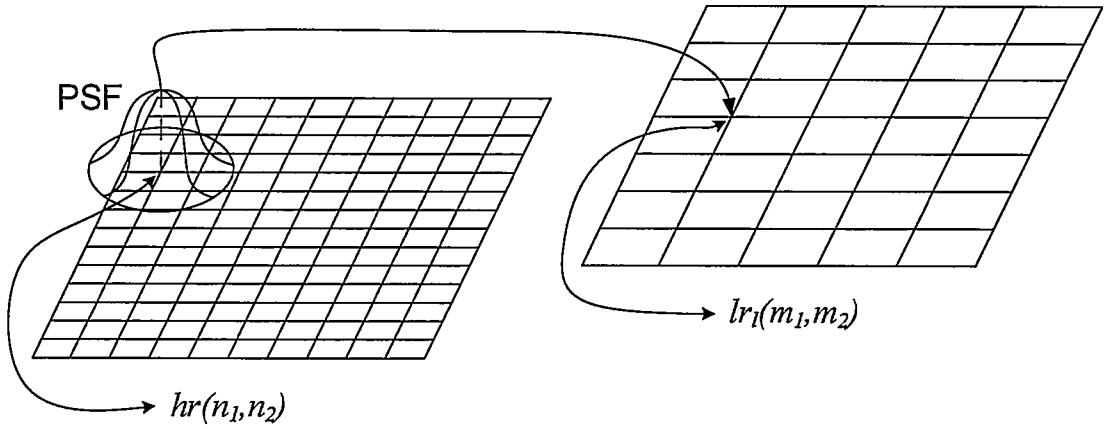


Figure 2.5: **Modelling the image capture process -** *Mapping from high-resolution image to low-resolution image*

2.2.3 POCS Applied to Image Superresolution

Application of the POCS method to the SR problem involves defining a different closed convex set for each low-resolution pixel (m_1, m_2) observed in lr_i :

$$\begin{aligned} C_{m_1,m_2,l} &= \{hr(n_1, n_2) \mid |r(m_1, m_2, l)| < \delta\}, \\ &0 \leq n_1 \leq N_1 - 1, \\ &0 \leq n_2 \leq N_2 - 1, \end{aligned} \tag{2.29}$$

and δ is equal to $c\sigma_v$, where σ_v is the standard deviation of the noise, and $c \geq 0$ is an appropriate statistical confidence bound. The residual $r(m_1, m_2, l)$ is defined as

$$r(m_1, m_2, l) \doteq lr(m_1, m_2, l) - \sum_{n_1} \sum_{n_2} h(m_1, m_2, l; n_1, n_2) hr(n_1, n_2). \tag{2.30}$$

This is simply the image intensity of the low-resolution image at (m_1, m_2) minus the value of the image capture process on hr taken at (n_1, n_2). The projection operation onto these convex sets is given by

$$\mathbf{P}_{(m_1,m_2,k)}[hr(n_1,n_2)] = \tag{2.31}$$
$$\begin{cases} hr(n_1,n_2) + \frac{r(m_1,m_2,k) - \delta\, h(m_1,m_2,k;n_1,n_2)}{\sum_{n_1,n_2} |h(m_1,m_1,k;n_1,n_1)|^2} & \text{, if} \quad r(m_1,m_2,k) > \delta \\ hr(n_1,n_2) & \text{, if} -\delta < r(m_1,m_2,k) < \delta \\ hr(n_1,n_2) + \frac{r(m_1,m_2,k) + \delta\, h(m_1,m_2,k;n_1,n_2)}{\sum_{n_1,n_2} |h(m_1,m_1,k;n_1,n_1)|^2} & \text{, if} \quad r(m_1,m_2,k) < \delta \end{cases}$$

The amplitude constraint C_A is given with

$$C_A = \{hr(n_1,n_2) \mid 0 \leq hr(n_1,n_2) \leq 255 \quad \text{for} \quad 0 \leq n_1 \leq N_1, 0 \leq n_2 \leq N_2\} \tag{2.32}$$

The projection operator onto C_A can then be defined as

$$\mathbf{P}_A[hr(n_1,n_2)] = \begin{cases} 0 & \text{, if} \quad hr(n_1,n_2) < 0 \\ hr(n_1,n_2) & \text{, if} \quad 0 \leq hr(n_1,n_2) \leq 255 \\ 255 & \text{, if} \quad hr(n_1,n_2) > 0 \end{cases} \tag{2.33}$$

Combining the projections from equations 2.31 and 2.33, and using the relaxed projection operator, the recursive update for the estimate of the high-resolution image $hr(n_1, n_2)$ becomes

$$hr^{j+1}(n_1,n_2) = T_A \prod_{\substack{(0,0) \leq (m_1,m_2) \leq (M_1,M_2) \\ k=1,\ldots,L}} T_{m_1,m_2,k}[hr^j(n_1,n_2)] \tag{2.34}$$

$$j = 0, 1, \ldots, \quad 0 \leq n_1 \leq N_1 - 1, \quad 0 \leq n_2 \leq N_2 - 1$$

A suitable initial estimate hr^0, to use in the image SR process is one of the low-resolution images that has been bilinearly interpolated to the resolution desired, and in the experimental setup lr_0 is used. The algorithm for implementing POCS is given in algorithm 3 [14]. The PSF h is modelled as a Gaussian kernel.

Algorithm 3 Projection Onto Convex Sets - POCS

Input: L low-resolution images

1. Bi-linearly interpolate the reference frame to get an initial estimate of hr, the high-resolution reconstruction.
2. For each low-resolution lr_i
   (a) Compute the motion parameters for the current low resolution image, lr_i, to lr_0.
   (b) For each low-resolution pixel in the current low resolution image, lr_i
      i. Compute the mapping from lr_i to hr and apply the image capture process to hr to obtain an estimate of the pixel intensity at the lr coordinates.
      ii. Calculate the residual and back-project the error onto hr using equation 2.31

Chapter 3

System Architecture

3.1 Network of Cameras

Creating a single high-resolution image from a number of low-resolution images is realized through a network of cameras communicating with a vision server as depicted in figure 3.1. For flexibility, camera nodes communicate with the vision server through a wireless interface, namely Bluetooth. A full description of the Bluetooth protocol would be beyond the scope of this document, but a brief overview is given in a later section. A wireless network allows for simplified physical setup, as wiring from nodes to servers is not required. Moreover, the wireless interface gives our proposed system flexibility in the addition, removal, and (to a certain extent) the placement of camera nodes.

3.1.1 Camera Node

Each camera nodes' primary function is image acquisition. Upon receipt of an image request, it is the task of the camera nodes to acquire an image and transmit it to the vision server. This simple image capture task is achieved through use of a simple,

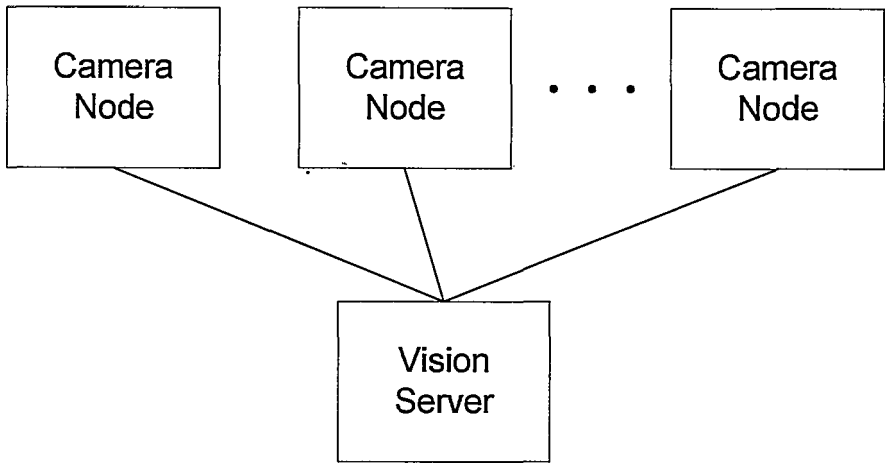


Figure 3.1: **Vision System Model** - *The vision system consists of outlying Camera Nodes communicating with the Vision Server.*

readily available Logitech Quickcam Camera. Wireless communication is achieved with point-to-point Ericsson Bluetooth modules. Both of these units connect to a workstation through Universal Serial Bus (USB) as in figure 3.2 (a).

3.1.2 Vision Server

The vision server is the core of the superresolution system. It organizes the image capture from camera nodes, and processes the images using the superresolution algorithm outlined in chapter 2. Similar to the camera nodes, wireless communication takes place through an Ericsson Bluetooth Module but with multi-point capabilities to accommodate communications with multiple camera nodes. As with the camera nodes, the Bluetooth module connects to a workstation through USB shown in figure 3.2 (b).

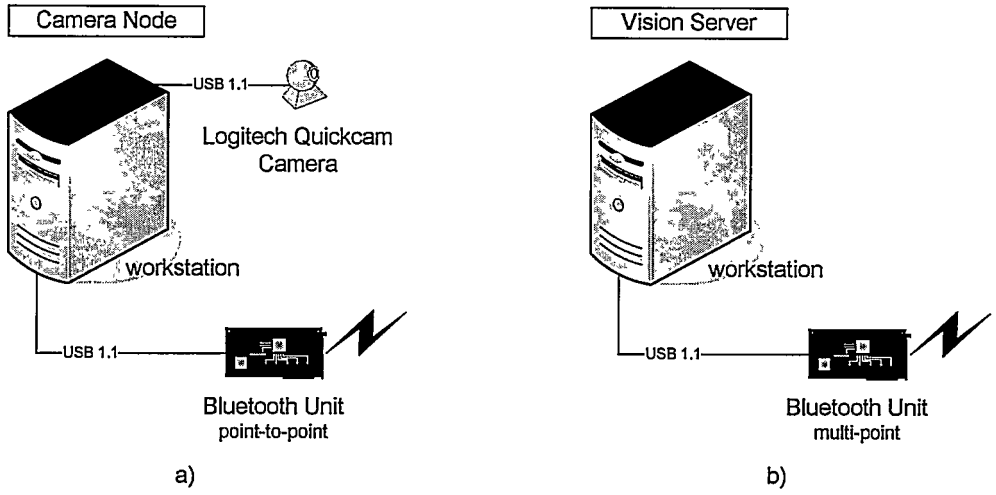


Figure 3.2: **Network Components** : *a) Camera Node - a camera node consists of an image capture device and wireless network card. b) Vision Server - the vision server also has a wireless network card and is responsible for organizing image capture between nodes and recreating the high-resolution image*

3.2 Bluetooth Overview

Bluetooth is a specification for wireless voice and data communications [3]. Its creators from Ericsson laboratories sought to find a wireless communications standard that on one hand could offer seamless cable replacement, and on the other hand provide new services and capabilities not currently available with existing technologies at the time. Some of Bluetooth's characteristics include :

- **Short Range** - Devices can communicate without wires over an air-interface within a 10 meter radius. As a consequence, Bluetooth communications requires very low power consumption which makes it suitable for small, portable devices.
- **Voice and Data** - With the convergence of both voice and data communications, Bluetooth facilitates the unifying of these worlds by allowing communications using either or both of these types of network traffic.

- **Unlicensed Band -** Many countries have restrictions on frequency spectrum usage. Bluetooth uses a frequency spectrum that is unlicensed throughout the world allowing a Bluetooth user to operate a device anywhere in the world without any modifications.

- **Open Specification -** To foster growth and widespread acceptance the Bluetooth standard is publicly available and royalty free.

These attributes make Bluetooth an attractive communication medium to implement the proposed superresolution system.

3.2.1 Technical Features of Bluetooth

Bluetooth wireless communication uses Radio Frequency (RF) technology. With RF technologies the problem of line of sight between a transmitter and receiver becomes irrelevant as RF waves can penetrate most obstacles such as clothing, walls, bodies doors, etc.

Bluetooth occupies the industrial, scientific and medical (ISM) frequency spectrum of 2.4 GHz which is unlicensed around the world. There are some restrictions on its use including :

- Frequency spectrum is divided into 79 channels

- Bandwidth is limited to 1 MHz per channel

- Frequency Hop Spread Spectrum (FHSS) must be employed

The main reason for these restrictions is to limit the amount of interference problems due to the availability of the spectrum to competing and complementary technologies. For simplicity, Bluetooth radios use Gaussian frequency-shift keying (GFSK) as their modulation technique. Using this scheme, the raw link speed achieved is 1 Mbps.

Bluetooth transmit power consumption ranges from 1 mW (typical Bluetooth radio) to 100 mW.

3.2.2 Bluetooth Networks

Bluetooth devices form a wireless network called a piconet. Devices in a piconet can take on one of two roles : master or slave. Within a piconet there can only be one master, and up to seven active slaves and up to 255 parked slaves. See figure 3.3 for an example. The master in a piconet governs the synchronization of the FHSS

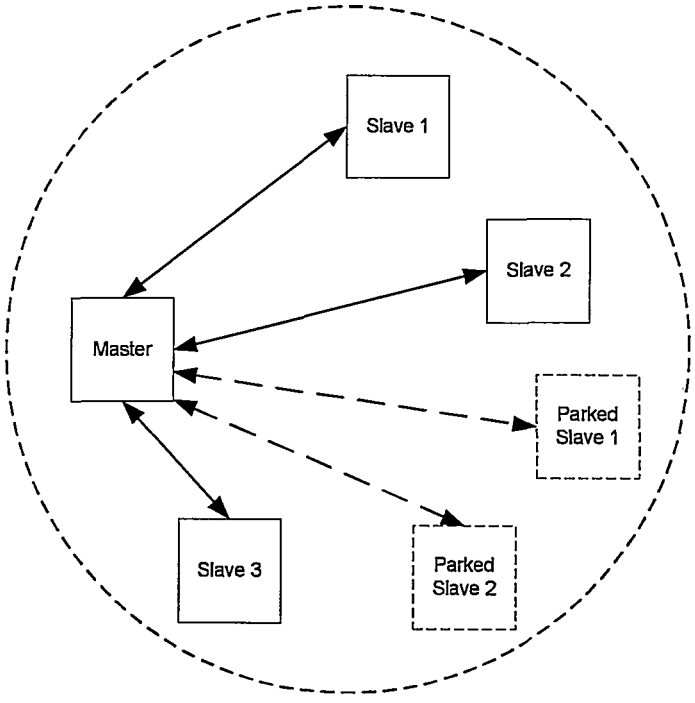


Figure 3.3: **A Simple Piconet** - *A simple piconet consisting of one master and three slave devices. The master dictates the synchronization pattern in a piconet.*

scheme used in the network. It determines the frequency hopping pattern and the phase for the hopping sequence. All slaves connected to the master in a piconet hop together in unison with the master. This master-slave relationship is primarily used to describe lower-level communications. From an application view-point, Bluetooth

devices generally operate as peers, and the designation of master or slave is irrelevant.

When piconets overlap in time and space, a *scatternet* is formed. A device can be a member of several piconets, and in fact, a master of one piconet, can take the role of slave in another. This allows for devices to easily maintain multiple connections or move between networks. An example of a scatternet is given in figure 3.4. The vision system presented in this thesis uses a simple piconet.

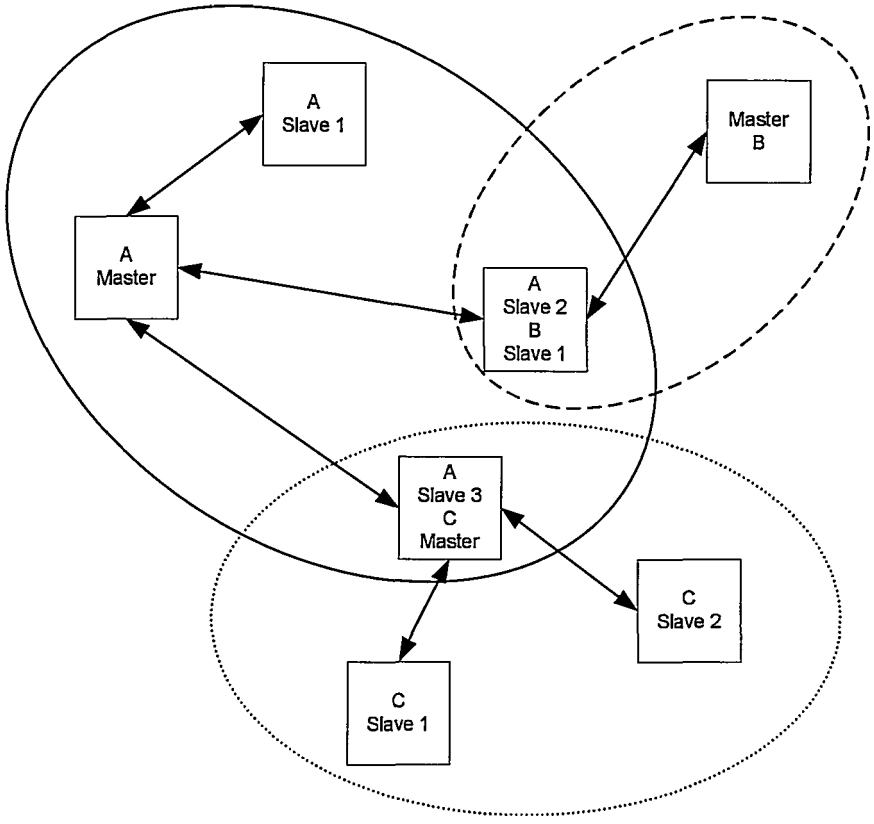


Figure 3.4: **A Scatternet -** *Overlapping piconets in space and time form a scatternet.*

3.2.3 Bluetooth Communications

The Bluetooth channel is accessed through a time division duplex (TDD), packet-based polling scheme. The channel is divided in time, into time slots that are 625 μs

in length. A packet transmission occupies one, three, or five time slots depending on packet size. The 2.4 GHz ISM band is split into j 1-MHz channels. The frequency channels covered are in the range $2,402 \leftrightarrow 2,402 + j$ MHz, and at any one time the frequency $2,402 + j$ is occupied, where $k = 0, 1, \ldots, j - 1$ and $j = 79$ in most countries.

When a master and slave are communicating, they alternate the use of the channel through transmitting in either an even time slot in the case of the master, or an odd time slot in the case of the slave (see figure 3.5). Each time slot occupies one of the j frequencies given above and the hopping sequence is a pseudo-random pattern determined by the master. Each slave communicating with the master follows the hopping sequence determined by the master. If a packet is longer than one time slot in length (three and five time slots are other possible lengths), then the frequency does not change until the packet is done transmitting (see figure 3.6).

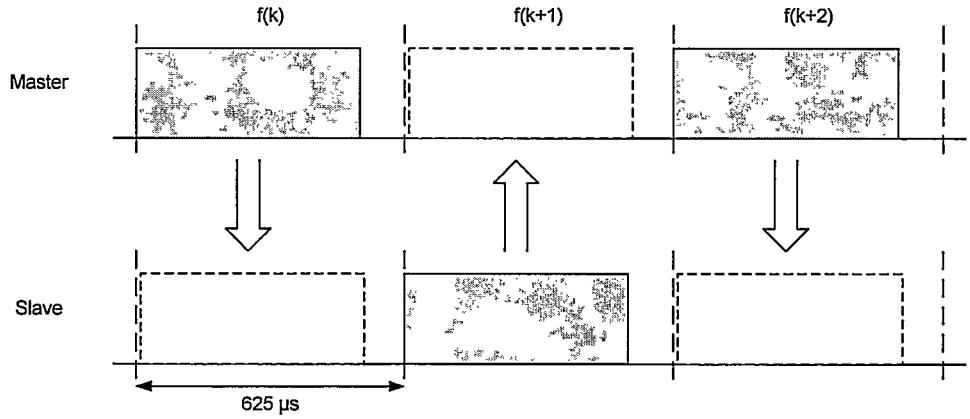


Figure 3.5: **Bluetooth Time Slots** - *Bluetooth time slots are* $625\mu s$ *in length. Masters and slaves alternate packet delivery through even and odd packet time slots.*

Communication Links

Between the master and slave, there are two types of links that can be established:

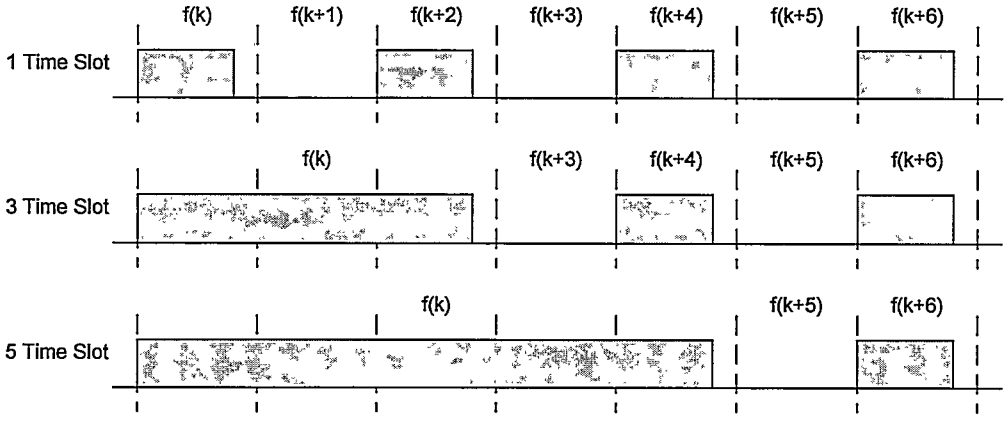


Figure 3.6: **Differing Bluetooth Packet Lengths** - *Packet transmissions occupying 1, 3, or 5 times slots.*

- Synchronous Connection-Oriented (SCO) link. SCO links are used primarily in time dependent applications such as voice or audio transmissions. SCO links can be viewed as circuit-switched connection from a master to a slave, with guaranteed packet delivery times. A master can support up to three SCO connections.

- Asynchronous Connection-Less (ACL) link. ACL links are connection-less oriented exchanges and can be viewed as a packet-switched connection between a master and slave. ACL packets can be transmitted in any slot not reserved for an SCO link. ACL connections are used exclusively in the system. There are seven different types of ACL packet types outlined in the table 3.1

The RFCOMM layer is responsible serial port abstraction in the Bluetooth layer stack. The RFCOMM layer is able to transmit and receive user data up to 32767 bytes (32K bytes) in length. Using DH5 packets, and 32K byte packets, a data rate of approximately 400 kb / s symmetrically can be achieved.

Table 3.1: Bluetooth Packet Types

Packet Type	Payload (Bytes)	Time Slots	FEC Encoded	CRC bits
DM1	18	1	2/3	16
DH1	28	1	NO	16
DM3	123	3	2/3	16
DH3	185	3	NO	16
DM5	226	5	2/3	16
DH5	341	5	NO	16
AUX1	30	1	NO	NONE

3.3 Communication Protocol

3.3.1 Initial Connection Procedures

The design of the multi-node Bluetooth card (used by the vision server) prohibits other cards from making incoming connections. This forces the vision server to establish links to camera nodes. The high-level connection procedure for a camera node to connect to the vision server starts with a camera node coming online and searching for the vision server. Once the server is found, the camera node sends a Request-To-Connect (RTC) packet that signals to the server that it would like to connect. The server, upon receiving the RTC packet, communicates back to the originating camera node and requests that a connection be made at the baseband layer of the Bluetooth Protocol stack. A list of connected nodes is kept at the server. Figure 3.7 illustrates the initial connection procedure.

3.3.2 Obtaining An Image Set

When a frame is to be superresolved, the server traverses its list of connected nodes, and sends a Request-For-Image (RFI) to all connected nodes to let them know that an image is to be taken. Upon receiving an RFI, a camera node captures an image and transmits it back to the server. Once all the nodes have submitted an image

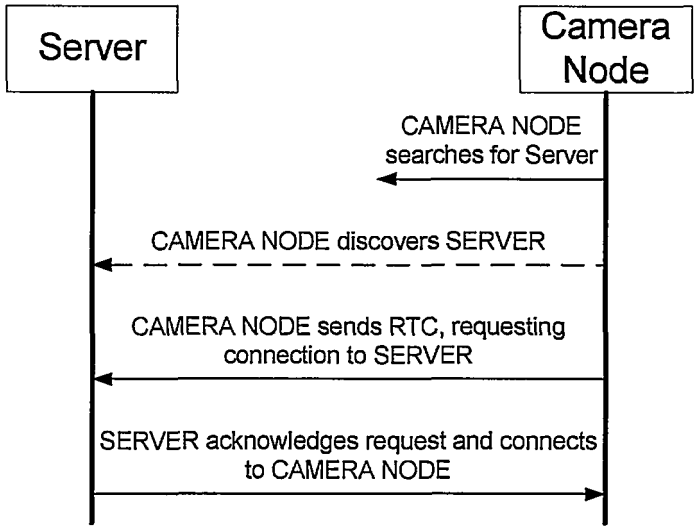


Figure 3.7: **Initial Connection Procedure** - *The initial connection procedure for a camera node to connect to the vision server.*

for the low-resolution set, the vision server collects the information and performs the superresolution algorithm to determine HR, the high-resolution estimate of the scene captured by the cameras. Figure 3.8 shows the image capture procedure.

3.3.3 Sending An Image

To transmit an image to the server, a camera node must first receive an RFI. Upon reception, it replies with an acknowledgment packet (RFI ACK) to notify the server that it will be sending the image. The RFI ACK packet contains how many bytes the server should expect to receive. Transmission of the image involves a serial dump of the bytes of the image file in a left-to-right, top-to-bottom scan of the image. Figure 3.9 illustrates the image transmission procedure.

3.3.4 Superresolving the Image Set

Once each camera node has transmitted its image of the scene, the image set is ready to be superresolved at the server. In the image capture process, the image from the

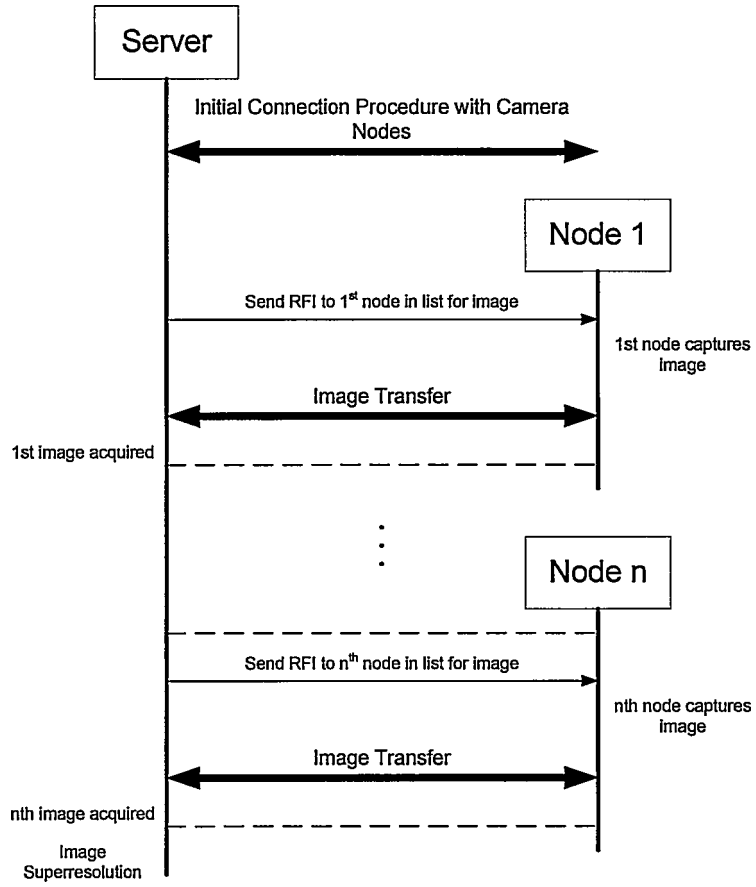


Figure 3.8: **Image Capture Procedure** - *The image capture procedure to obtain a low-resolution image set to superresolve.*

first camera node is designated as the base image, lr_0. Subsequent images are added to the set in the order of acquisition. The images are then input into the image superresolution algorithm outlined in chapter 2, with the output being a single hr image.

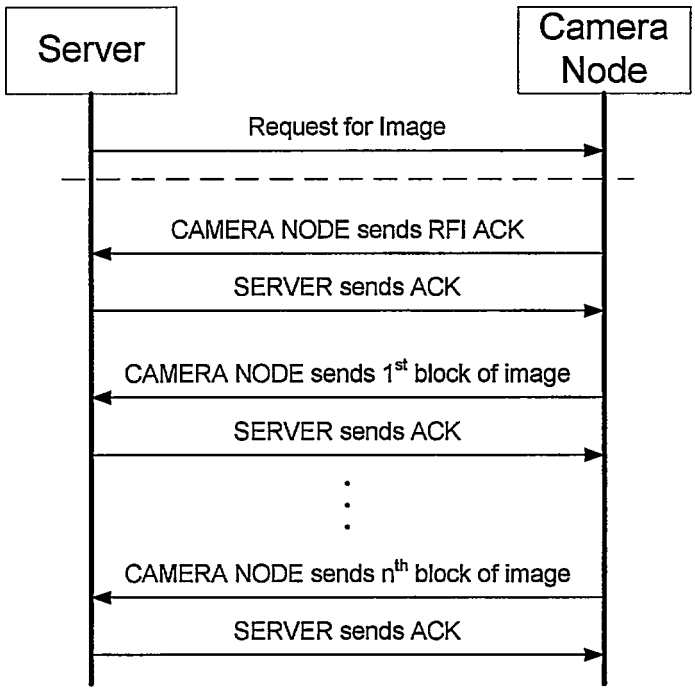


Figure 3.9: **Sending an Image** - *The image transmission procedure from vision server request to camera node transmission.*

Chapter 4

System Performance

The performance of the system can be characterized by how effective the superresolution algorithm performs in creating the high-resolution output image. The performance of the POCS algorithm is analyzed by simulating *lr* observations through the image capture process on a *hr* image, and applying the SR algorithm to the *lr* image set. The performance of the vision system as a whole is also examined.

4.1 POCS Performance

4.1.1 Artificial Low Resolution Images

The POCS algorithm was tested to evaluate the efficacy of the superresolution results in a quantitative manner. To simulate low-resolution observations, an initial high-resolution image was captured and artificial low-resolution images were produced. The image capture process from equation 2.28 was simulated by applying a Gaussian blurring kernel g to the high-resolution image hr, to produce a blurred hr' where

$$g(n_1, n_2) = e^{\frac{n_1^2+n_2^2}{2\sigma^2}} \tag{4.1}$$

$$hr' = \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} hr(k,l)g(n_1 - k, n_2 - l). \tag{4.2}$$

To produce the simulated low-resolution images for the image set, hr' is downsampled at differing offsets by a factor D in both directions. The low-resolution images are given by

$$lr_{d_1,d_2}(m_1, m_2) = hr'((2m_1 - 1) + d_1, (2m_2 - 1) + d_2) \tag{4.3}$$

$$D \in \mathbb{N}$$

$$d_1 = 0, 1, \ldots, D - 1$$

$$d_2 = 0, 1, \ldots, D - 1.$$

Three examples are shown where high-resolution images at a resolution of 640×480 pixels were taken. Example 1 in figure 4.1 is from the 1951 United States Air Force test resolution pattern [8] and is referred to as the *Res-Chart* image. Example 2 in figure 4.2 shows a portrait of students in the Machine Vision and Image Processing Lab and is referred to as the *Faces* image. Example 3 shown in figure 4.3 is the rear of a car and is referred to as the *Car* image. The results of the blurring and downsampling procedure follow. The size of the Gaussian blurring kernel was set with a window size of 5×5 pixels, with a downsampling factor $D = 2$. The resulting low-resolution image sets consist of 4 images with a resolution of 320×240. One image from each of the low-resolution sets can be found in figures 4.4, 4.5, 4.6.

4.1.2 POCS

The POCS algorithm was implemented on the low-resolution image sets. The number of low-resolution images used to reconstruct the high-resolution estimate varied from 2 to 4 images. The size of the POCS PSF was also tested at window sizes of 3×3 and 5×5. Finally the number of POCS iterations tested were 1, 5, 10, 15 and 20

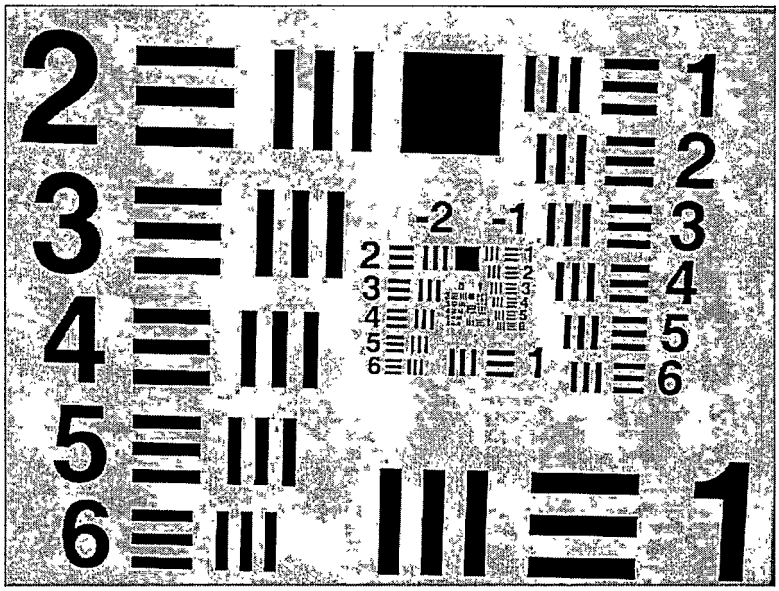


Figure 4.1: **Example 1 - High-resolution Res-Chart image** *at a resolution of 640 × 480*

iterations. The measure of fidelity is the mean squared error (MSE) between our reconstructed image and the original high-resolution image from figures 4.1, 4.2, 4.3 (here, the error is the difference in raw pixel intensity between the images). The MSE between the original image and the first frame bilinearly interpolated from each set was also calculated for comparison purposes. The reconstruction result that yielded the lowest MSE for each image tested are given in figures 4.7, 4.9, 4.11. Figures 4.8, 4.10, 4.12 show the bilinearly interpolated first frame from each image set for comparison.

The full MSE results can be found in table 4.1. It is important to note that in most cases, the MSE for the POCS restored image was less than that of the bilinearly interpolated image. The POCS restored images are also sharper and reveal some finer details (notice the inner set of numbers on the res-chart image, the detail in the teeth and shirt patterns for the faces image, and the detail in the license plate for the car image).

Figure 4.2: **Example 2 - High-resolution Faces image** *at a resolution of 640 × 480*

Figure 4.3: **Example 3 - High-resolution Car image** *at a resolution of 640 × 480*

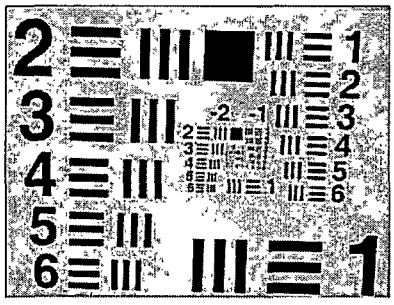


Figure 4.4: **Example 1 - Low-resolution Res-Chart image** *at a resolution of 320 × 240*

Figure 4.5: **Example 2 - Low-resolution Faces image** *at a resolution of 320 × 240*

Figure 4.6: **Example 3 - Low-resolution Car image** *at a resolution of 320 × 240*

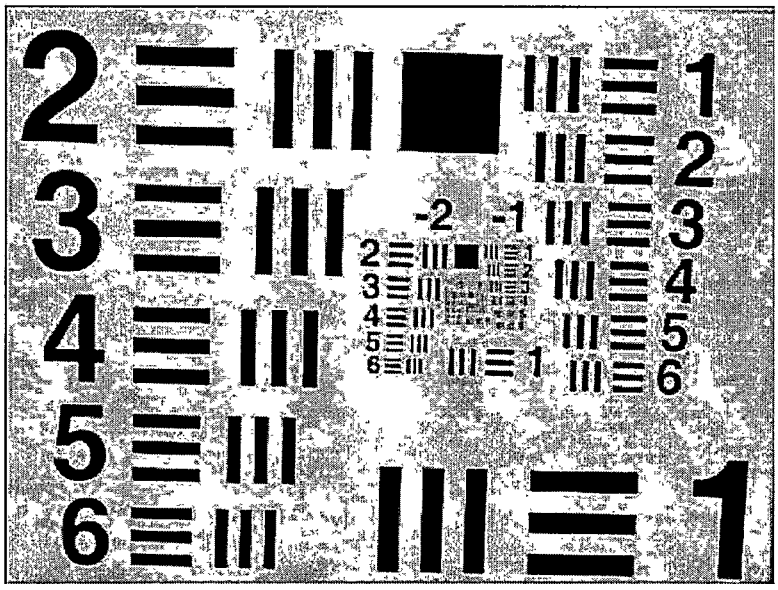


Figure 4.7: **Best POCS reconstructed Res-Chart image** - *Reconstructed Res-Chart image using 4 low-resolution images. The POCS algorithm with a PSF window size of* 3×3 *pixels at 20 iterations was used.*

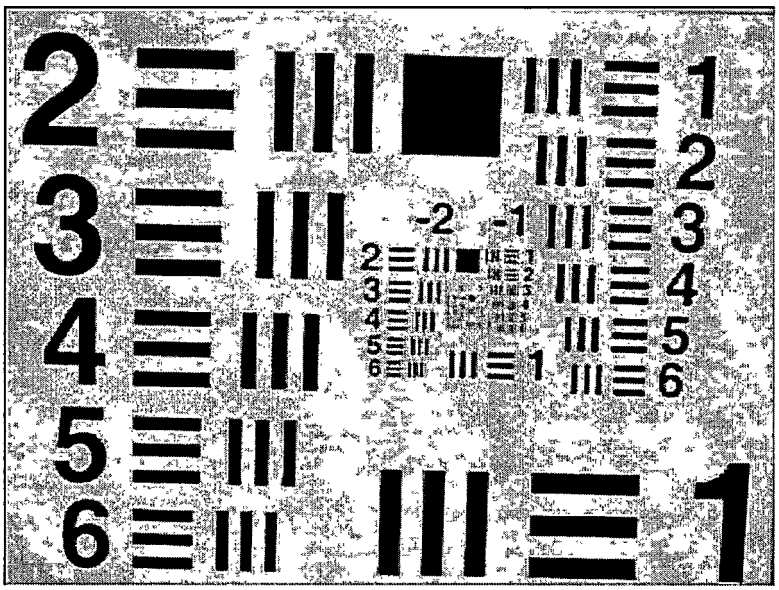


Figure 4.8: **Bilinearly interpolated Res-Chart image** - *Reconstructed Res-Chart image using bilinear interpolation from a downsampled high-resolution image.*

Figure 4.9: **Best POCS reconstructed Faces image** - *Reconstructed Faces image using 4 low-resolution images. The POCS algorithm with a PSF window size of* 5×5 *pixels at 20 iterations was used.*

Figure 4.10: **Bilinearly interpolated Faces image** - *Reconstructed Faces image using bilinear interpolation from a downsampled high-resolution image.*

Figure 4.11: **Best POCS reconstructed Car image** - *Reconstructed Car image using 4 low-resolution images. The POCS algorithm with a PSF window size of* 5×5 *pixels at 20 iterations was used.*

Figure 4.12: **Bilinearly interpolated Car image** - *Reconstructed Car image using bilinear interpolation from a downsampled high-resolution image.*

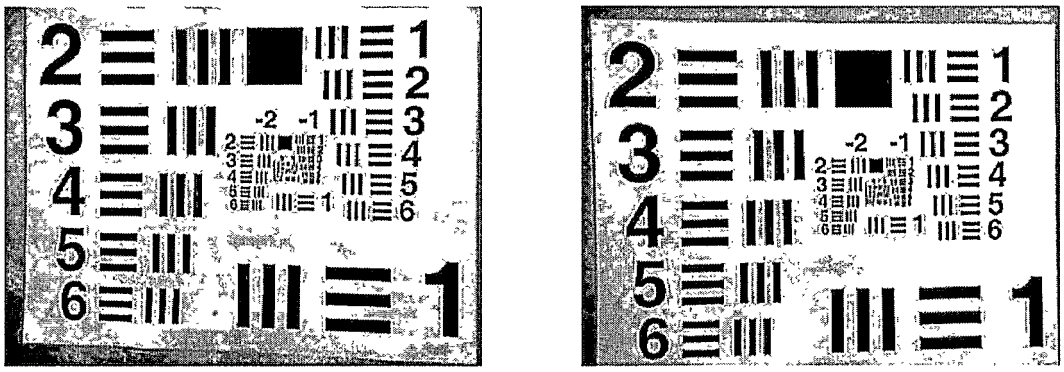
Table 4.1: MSE for POCS Restored Images.

	Images	Iterations					B.I.
		1	5	10	15	20	
ResChart Window 3	2	392.43	396.26	396.26	396.26	396.26	566.2068
	3	477.81	489.18	489.20	489.20	489.20	
	4	236.00	235.16	235.16	235.16	235.16	
ResChart Window 5	2	385.02	414.03	414.27	414.28	414.28	566.2068
	3	554.87	601.75	609.91	613.57	615.40	
	4	253.87	254.06	256.08	256.98	257.38	
Faces Window 3	2	172.70	175.38	175.38	175.38	175.38	215.3483
	3	176.49	178.67	178.69	178.69	178.69	
	4	129.95	129.83	129.83	129.83	129.83	
Faces Window 5	2	168.94	180.00	180.31	180.33	180.33	215.3483
	3	184.55	193.30	193.74	193.86	193.91	
	4	129.48	129.47	129.56	129.60	129.62	
Car Window 3	2	152.49	152.19	152.18	152.18	152.18	214.94
	3	199.29	202.96	203.02	203.06	203.08	
	4	99.26	98.97	99.03	99.06	99.07	
Car Window 5	2	152.05	157.17	157.43	157.45	157.45	214.94
	3	228.71	242.71	243.30	243.57	243.75	
	4	104.04	102.46	102.70	102.86	102.99	

4.2 System Performance

In this section, the performance of the system is analyzed. The output of intermediate steps for the image registration process is shown, as well as the final high-resolution estimate. The cameras capture images at a resolution of 320 × 240 and the high-resolution estimates are reconstructed at a resolution of 640 × 480. The Bluetooth hardware was limited to transmission packet sizes of 100 Bytes from a client node to the server. The transmitted images produce a payload of 76800 Bytes resulting in 768 image packets sent from the client to the server to transmit one image.

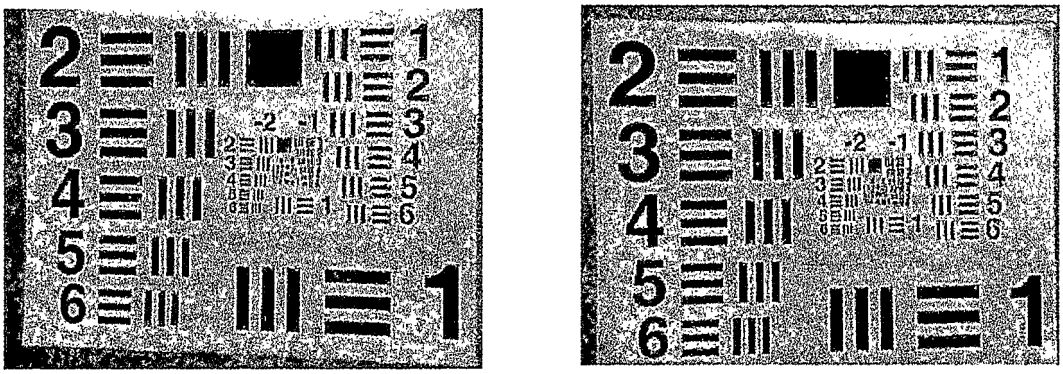
Three different image sets were tested and their output follows. The 1951 USAF resolution test pattern was again used for one of the image sets and are referred to as the Res-Chart image set. Another image is taken from a poster of the Schloss Neuschwanstein Castle [13] and is referred to as the Castle image set. The final example is a poster of the Escher work 'Waterfall' [5], and is referred to as the Escher image set. The Res-Chart low-resolution images can be found in figure 4.13. The



(a) (b)

Figure 4.13: **Low-resolution Res-Chart image set from Logitech cameras** *: (a) image 1 - base image, (b) image 2.*

Harris features can be found in figure 4.14. Image 1 contained 167 feature points while image 2 had 136 feature points. The initial set of 71 correspondences found can be found in figure 4.15(a). Notice the obvious mismatched points. Figure 4.15(b)

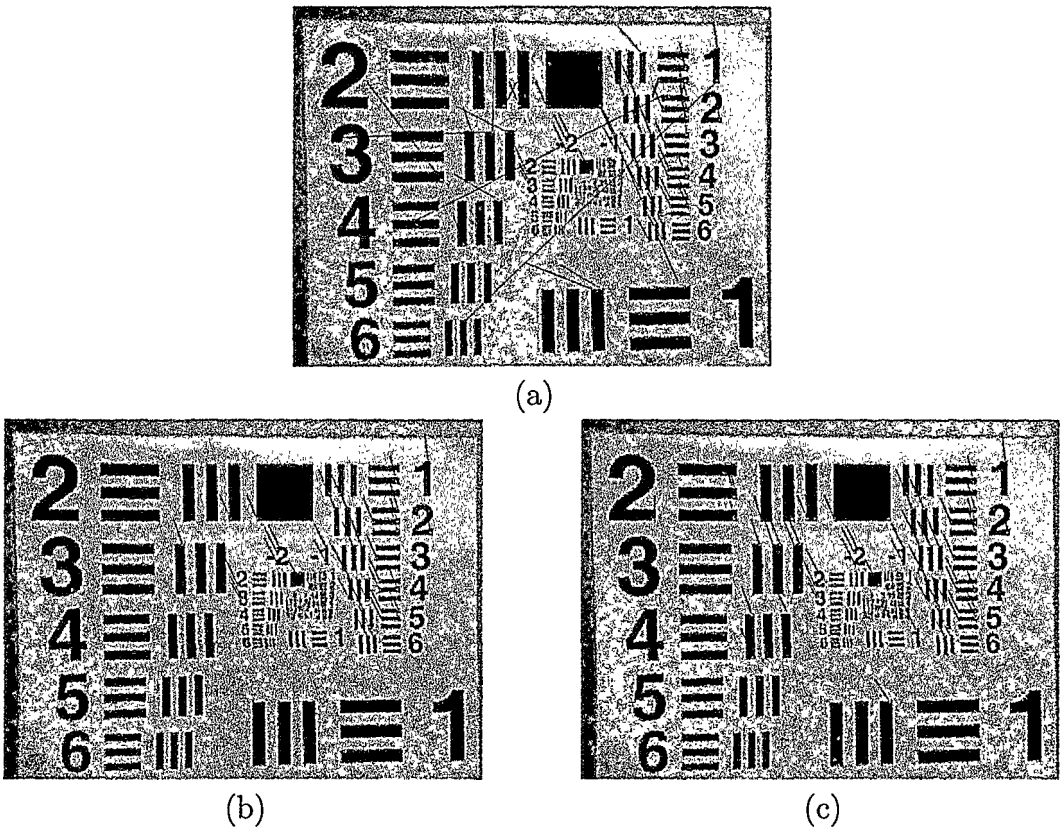


(a) (b)

Figure 4.14: **Harris feature points for Res-Chart image set** *: (a) image 1 - base image, (b) image 2.*

shows the initial set of 48 inlying correspondence points after the RANSAC algorithm. This step categorized 23 of the correspondences as outliers. The final set of 66 inliers that are settled upon after the guided matching step are shown in figure 4.15(c). The homography that transforms image 2 to image 1 was calculated with the warped image in figure 4.16(a), and the warped image interpolated in figure 4.16(b). The reconstructed bilinearly interpolated image can be compared with the POCS restored image in figures 4.17 and 4.18. Crops of the high-resolution images are given in figure 4.19(a) and (b). The lines are sharper and more distinct in the POCS reconstructed image. A fast fourier transform (FFT) analysis of the crop was performed to visually compare the high frequency components of the content of the images. The FFT of the cropped image shows more high-frequency components present in the POCS restored image. Table 4.3 and figure 4.20 show the actual time and percentage breakdown of each task in the system. Image transmission and correspondence matching accounted for 78% of the total time for the system to produce the high-resolution image estimate. A further 15% was required to establish the connections necessary, while the remaining 7% was used to do the majority of the image processing calculations.

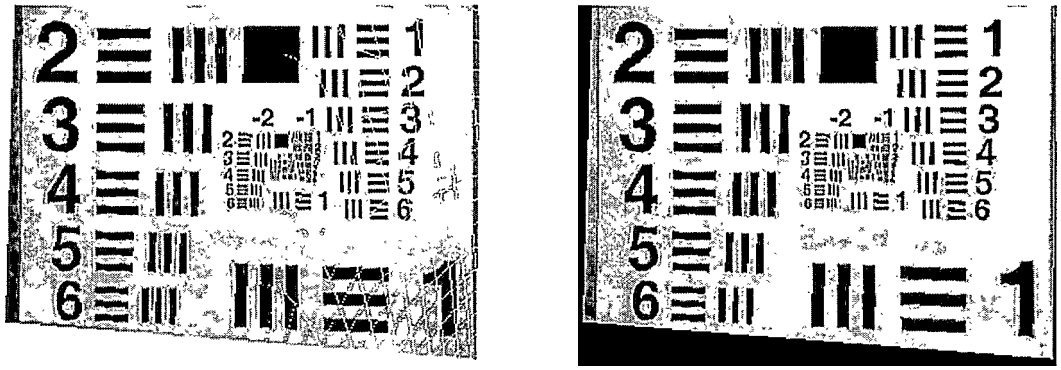
The Castle image set produced 165 and 182 feature points in image 1 and image



(a) (b) (c)

Figure 4.15: **Inliers for Res-Chart image set** *: (a) putative correspondences, (b) initial set of inliers, (c) final set of inliers.*

2 respectively. The low-resolution images along with their feature points extracted can be found in figures 4.21 and 4.22. From these feature points, 124 prospective correspondences were found with a final count of 121 inlying data points. The correspondence matches, initial inliers and final inliers are shown in figure 4.23 with a summary found in table 4.4. The warped image resulting from the homography is given in figure 4.24(a), and the warped image interpolated in figure 4.16(b). The reconstructed bilinearly interpolated Castle image can be compared with the POCS restored image in figures 4.25 and 4.26. Crops of the high-resolution images are can be seen figure 4.27(a) and (b). The POCS reconstruction again appears sharper with



(a) (b)

Figure 4.16: **Warped image 2 from Res-Chart image set** : *(a) warped, (b) warped and interpolated.*

Table 4.2: Interest Points for Resolution Chart Image Set

	Image 1	Image 2
Feature Points	167	136
Putative Correspondences	71	
Initial Inliers	48	
Final Inliers	66	

more contrast. The FFT of the cropped image shows more high-frequency components present in the POCS restored image. Table 4.5 and figure 4.28 give a breakdown of the timing of the system. Image transmission and correspondence matching accounted for 84% of the total time for the system to produce the high-resolution image estimate. A further 10% was required to establish the connections necessary, while the remaining 6% was used to do the majority of the image processing calculations.

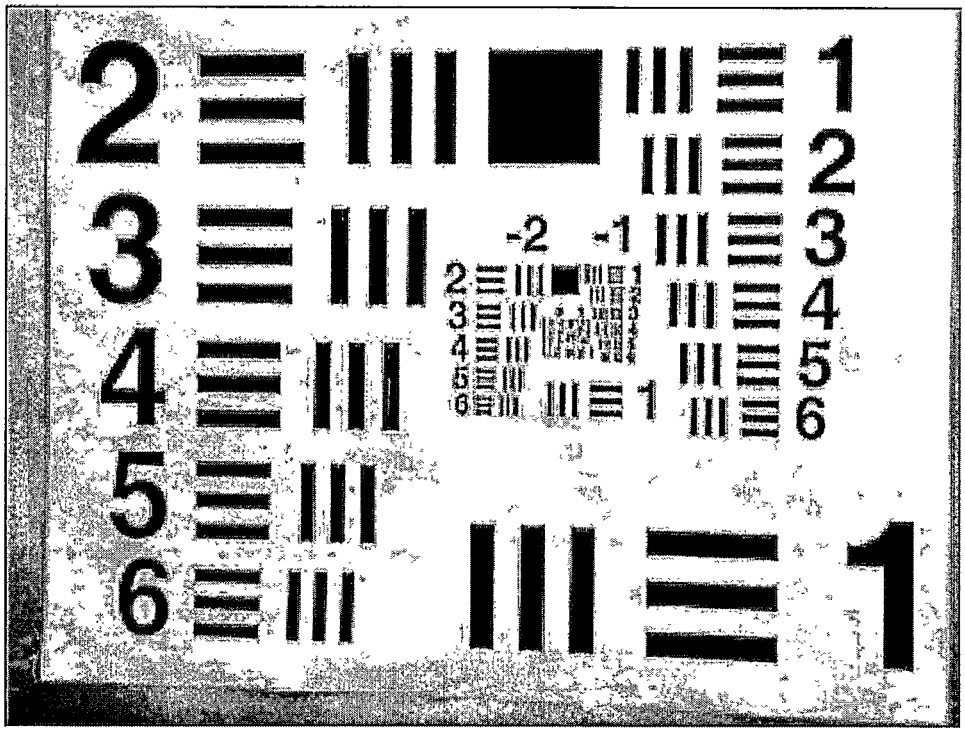


Figure 4.17: Bilinearly interpolated base image from Res-Chart image set.

Table 4.3: Timing for Res-Chart Image Set

Procedure	**Time** (s)
Connection	44.44
Image Transfer	112.99
Correspondence Matching	114.39
RANSAC	2.61
Guided Matching	0.343
Warping	9.735
Superresolution	8.734
Total	**293.242**

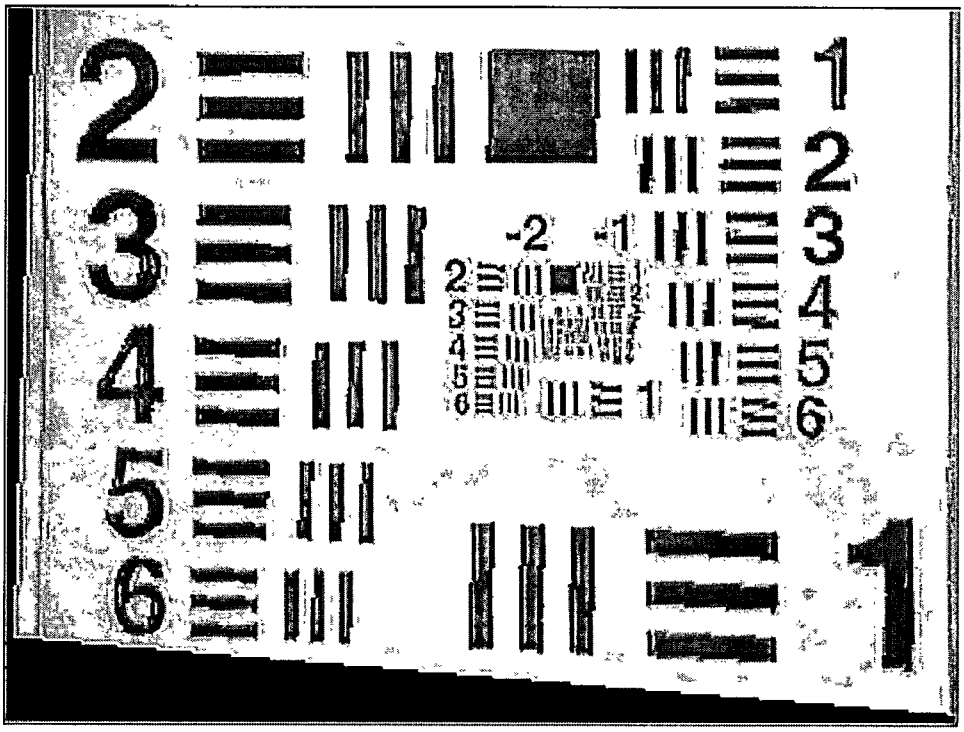


Figure 4.18: POCS restored base image from Res-Chart image set.

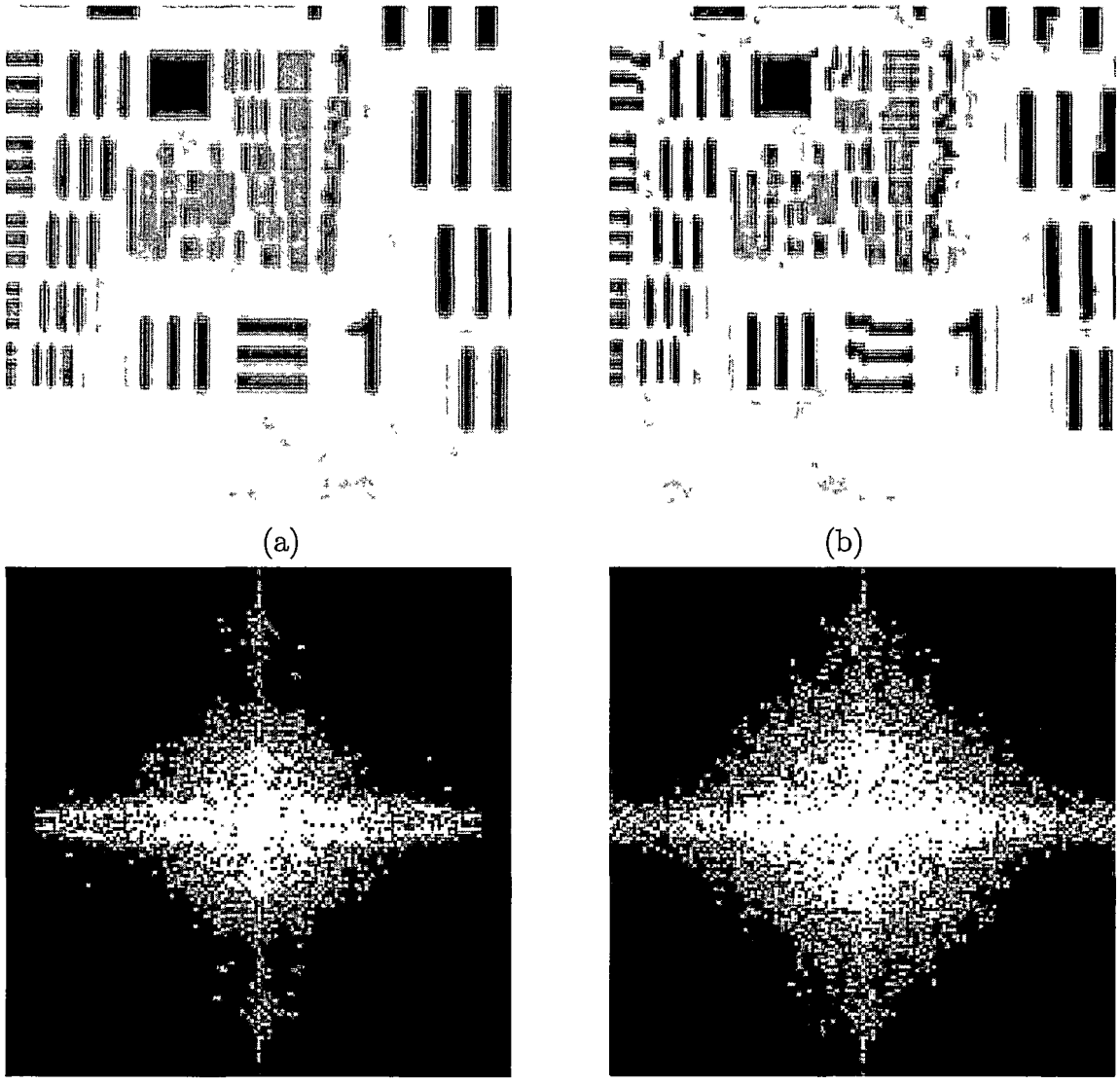


Figure 4.19: **Crops and FFT analysis of bilinearly interpolated and POCS restored Res-Chart image set** : *Crops of bilinearly interpolated and POCS restored Res-Chart image [(a) and (b) respectively]. FFT analysis on the cropped segments [(c) and (d) respectively].*

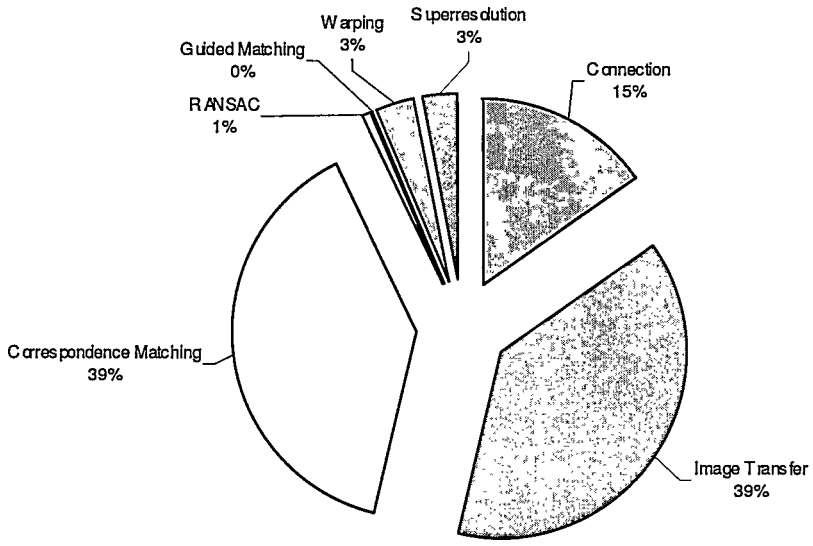
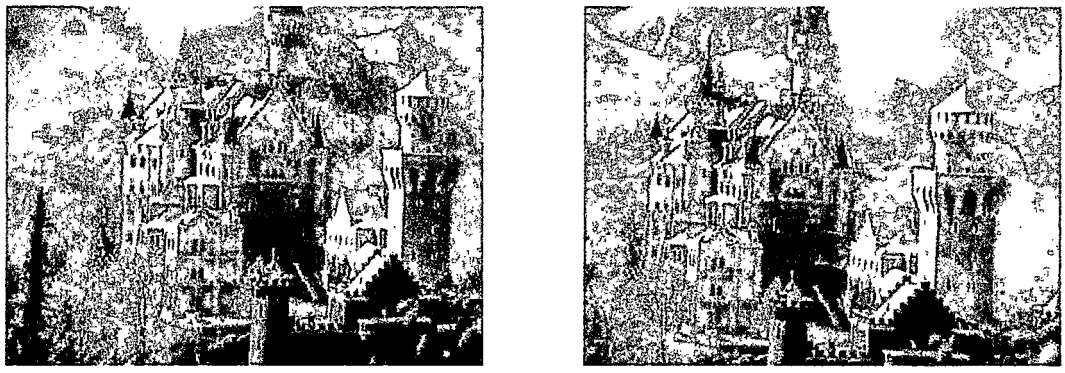
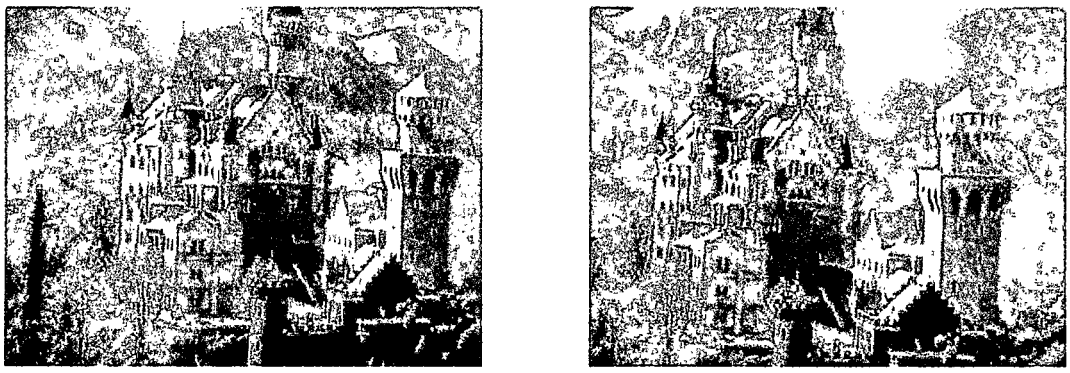


Figure 4.20: Timing Analysis for Res-Chart Image Set



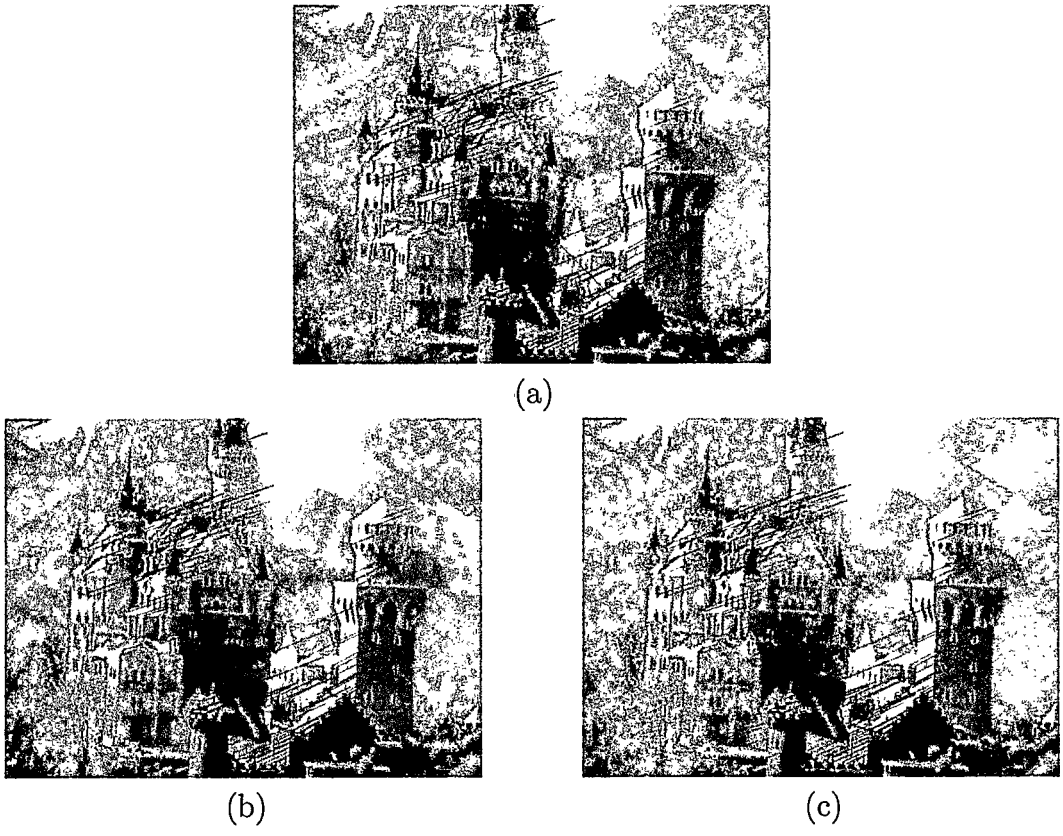
(a) (b)

Figure 4.21: **Low-resolution castle image set from Logitech cameras** *: (a) image 1 - base image, (b) image 2.*



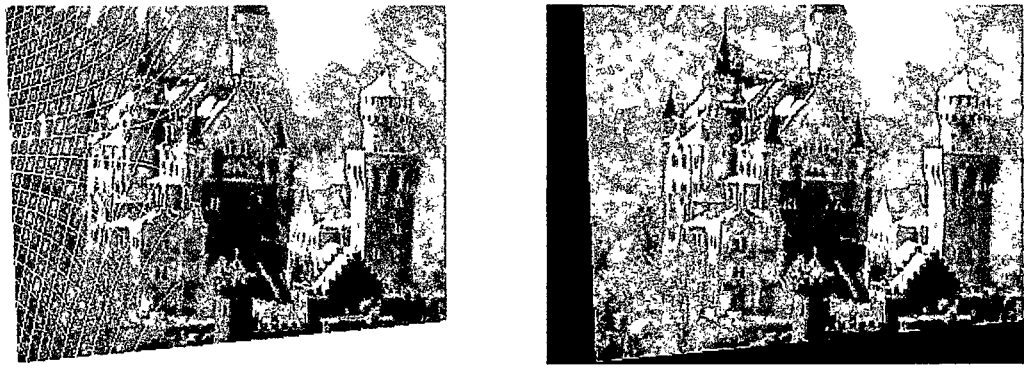
(a) (b)

Figure 4.22: **Harris feature points for Castle image set** *: (a) image 1 - base image, (b) image 2.*



(a) (b) (c)

Figure 4.23: **Inliers for Castle image set** : *(a) putative correspondences, (b) initial set of inliers, (c) final set of inliers.*



(a) (b)

Figure 4.24: **Warped image 2 from Castle image set** : *(a) warped, (b) warped and interpolated.*

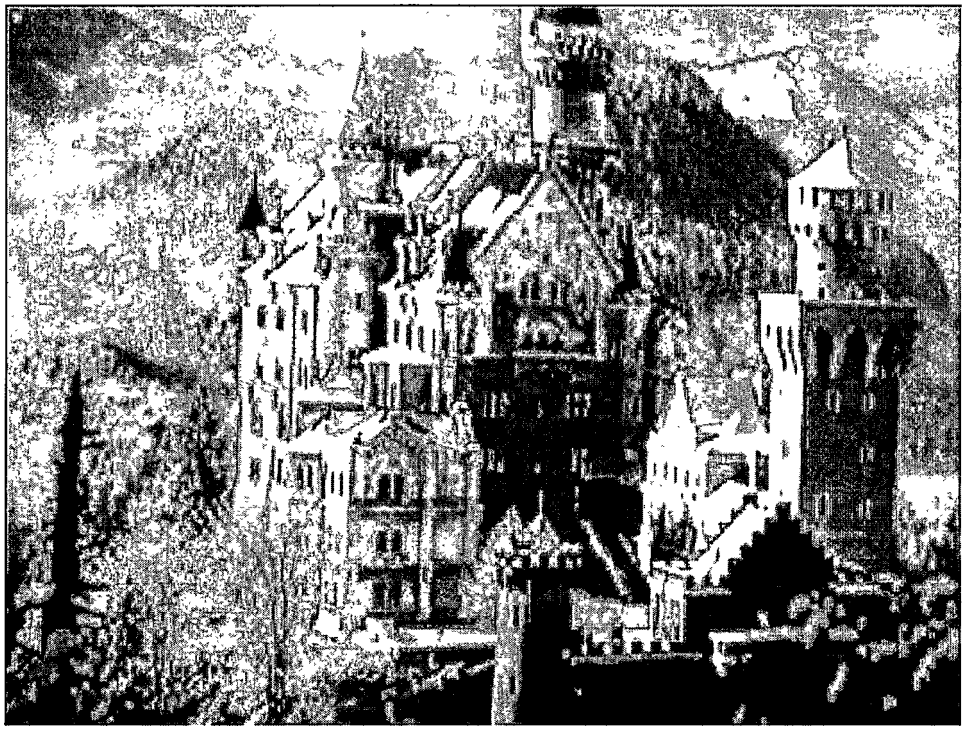


Figure 4.25: Bilinearly interpolated base image from Castle image set.

Table 4.4: Interest Points for Castle Image Set

	Image 1	Image 2
Feature Points	165	182
Putative Correspondences	124	
Initial Inliers	116	
Final Inliers	121	

Table 4.5: Timing for Castle Image Set

Procedure	**Time** (s)
Connection	32.65
Image Transfer	116.76
Correspondence Matching	149.469
RANSAC	1.031
Guided Matching	0.344
Warping	8.531
Superresolution	8.781
Total	**317.566**

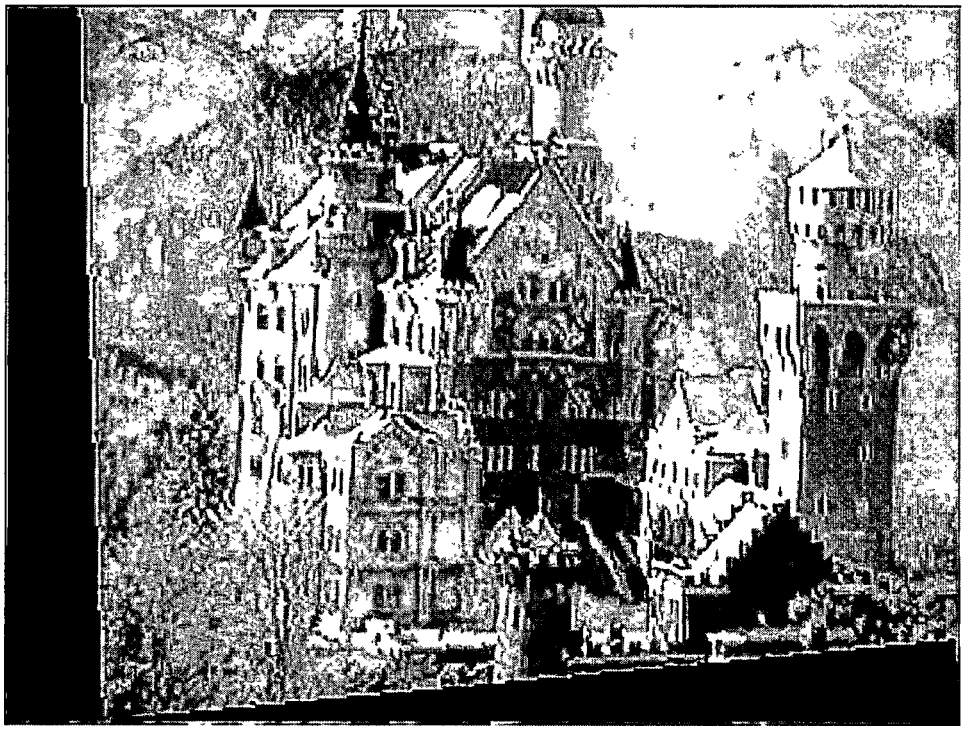


Figure 4.26: POCS restored base image from Castle image set.

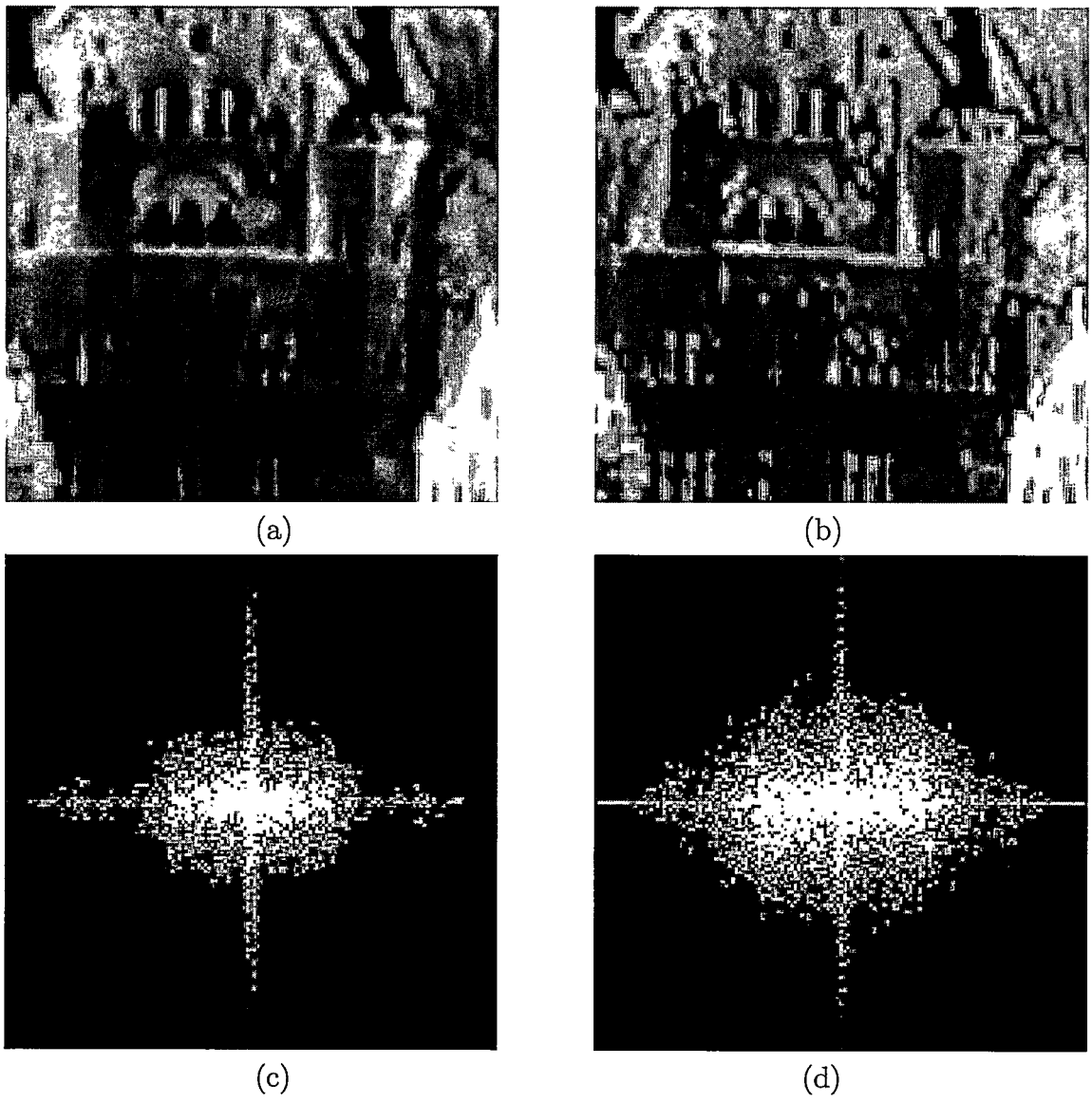


Figure 4.27: **Crops and FFT analyis of bilinearly interpolated and POCS restored Castle image** *: Crops of bilinearly interpolated and POCS restored Castle image [(a) and (b) respectively]. FFT analysis on the cropped segments [(c) and (d) respectively].*

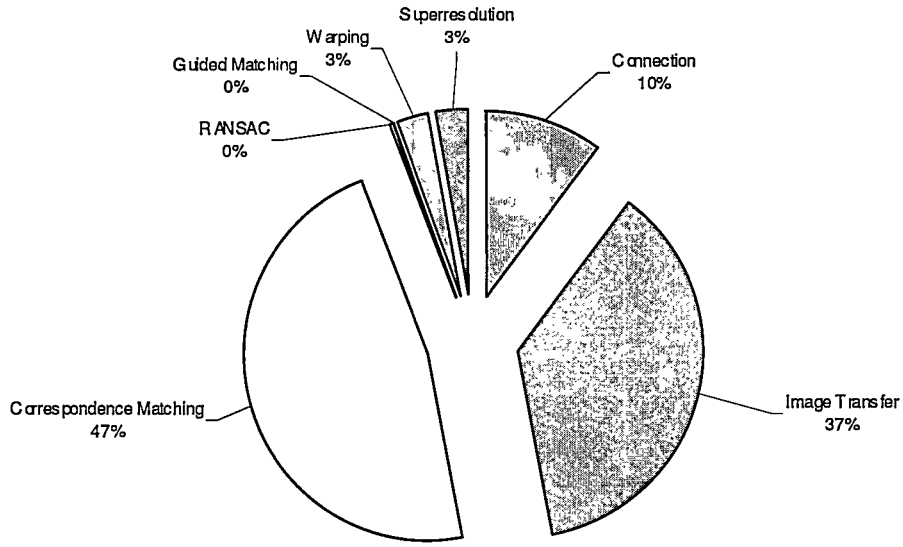


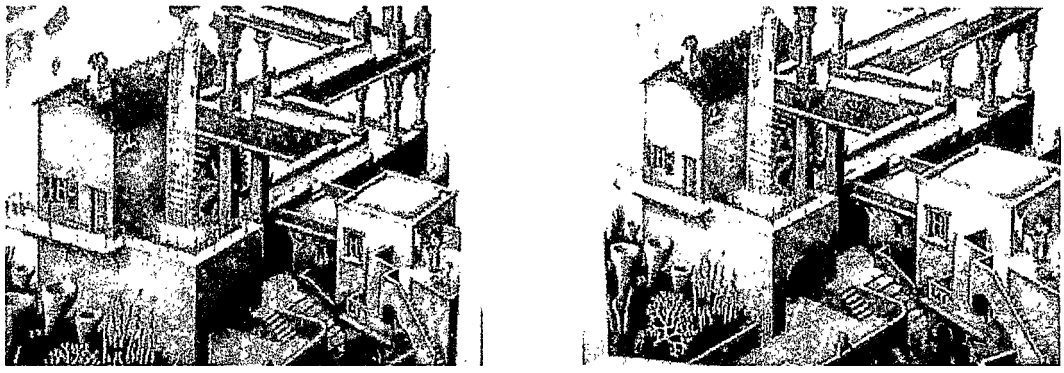
Figure 4.28: Timing Analysis for Castle image set.

The Escher image set produced 214 and 229 feature points in image 1 and image 2 respectively. The low-resolution images along with their feature points extracted can be found in figures 4.29 and 4.30. From these feature points, 123 prospective correspondences were found with a final count of 120 inlying data points. The correspondence matches, initial inliers and final inliers are shown in figure 4.31 and summarized in table 4.6. The warped image resulting from the homography is given in figure 4.32(a), and the warped image interpolated in figure 4.32(b). The reconstructed bilinearly interpolated Escher image can be compared with the POCS restored image in figures 4.33 and 4.34. Crops of the high-resolution images are can be seen figure 4.35(a) and (b). The POCS reconstruction shows finer detail, especially in the water wheel. The FFT of the cropped image shows more high-frequency components present in the POCS restored image. Table 4.7 and figure 4.36 give a breakdown of the timing of the system. Image transmission and correspondence matching accounted for 87% of the total time for the system to produce the high-resolution image estimate. A further 8% was required to establish the connections necessary, while the remaining 5% was used to do the majority of the image processing calculations.

For the convenience of the reader, table 4.8 provides a summary of all the system performance timings for comparison.

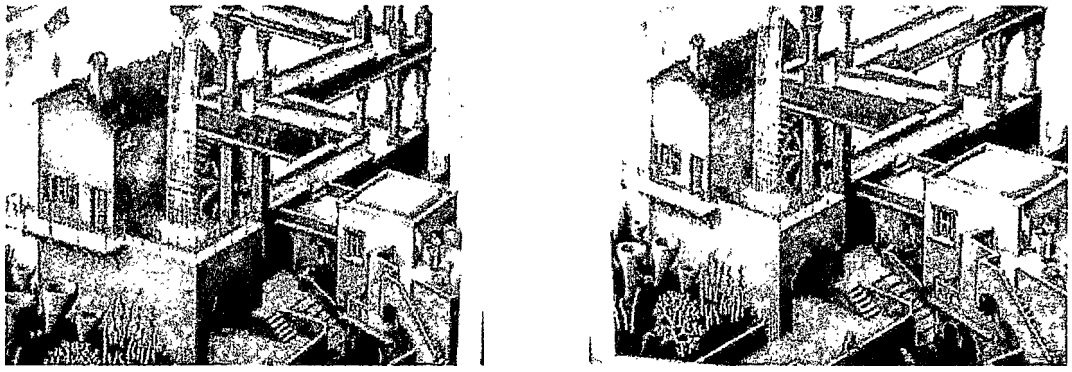
Table 4.6: Interest Points for Escher Image Set

	Image 1	Image 2
Feature Points	214	229
Putative Correspondences	123	
Initial Inliers	115	
Final Inliers	120	



(a) (b)

Figure 4.29: **Low-resolution Escher image set from Logitech cameras** *: (a) image 1 - base image, (b) image 2.*

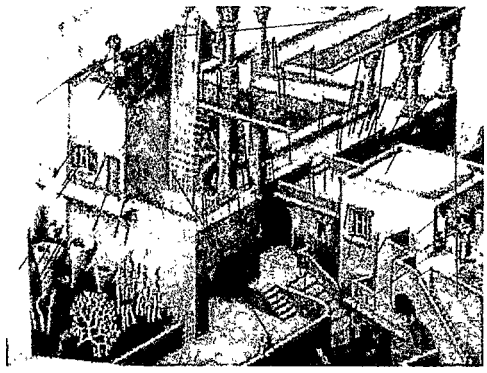


(a) (b)

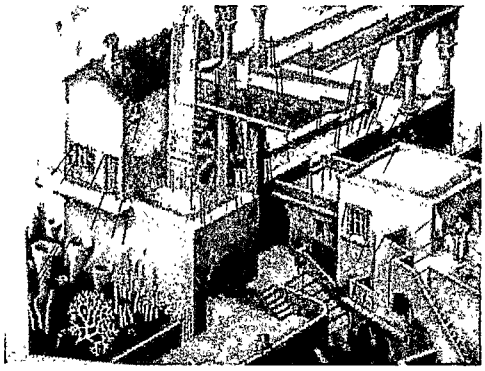
Figure 4.30: **Harris feature points for Escher image set** *: (a) image 1 - base image, (b) image 2.*

Table 4.7: Timing for Escher Image Set

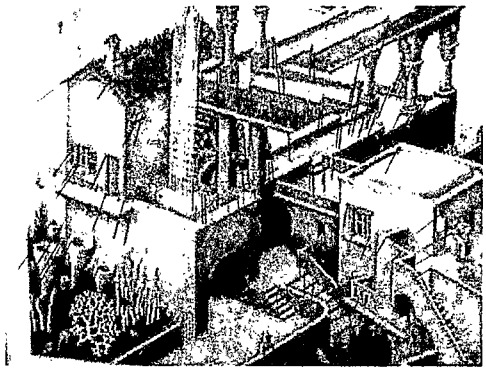
Procedure	**Time** (s)
Connection	31.23
Image Transfer	115.92
Correspondence Matching	200.703
RANSAC	0.781
Guided Matching	.359
Warping	10.985
Superresolution	8.640
Total	**368.618**



(a)

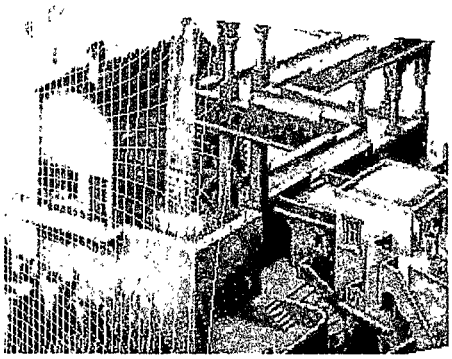


(b)

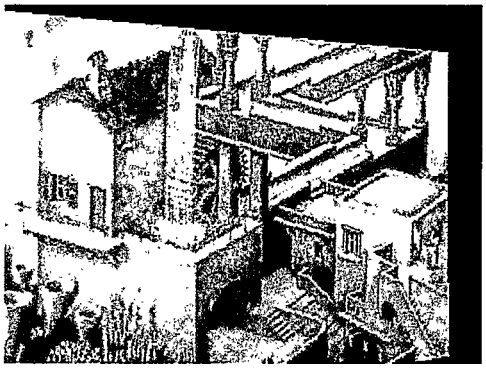


(c)

Figure 4.31: **Inliers for Escher image set** *: (a) putative correspondences, (b) initial set of inliers, (c) final set of inliers.*



(a)



(b)

Figure 4.32: **Warped image 2 from Escher image set** *: (a) warped, (b) warped and interpolated.*

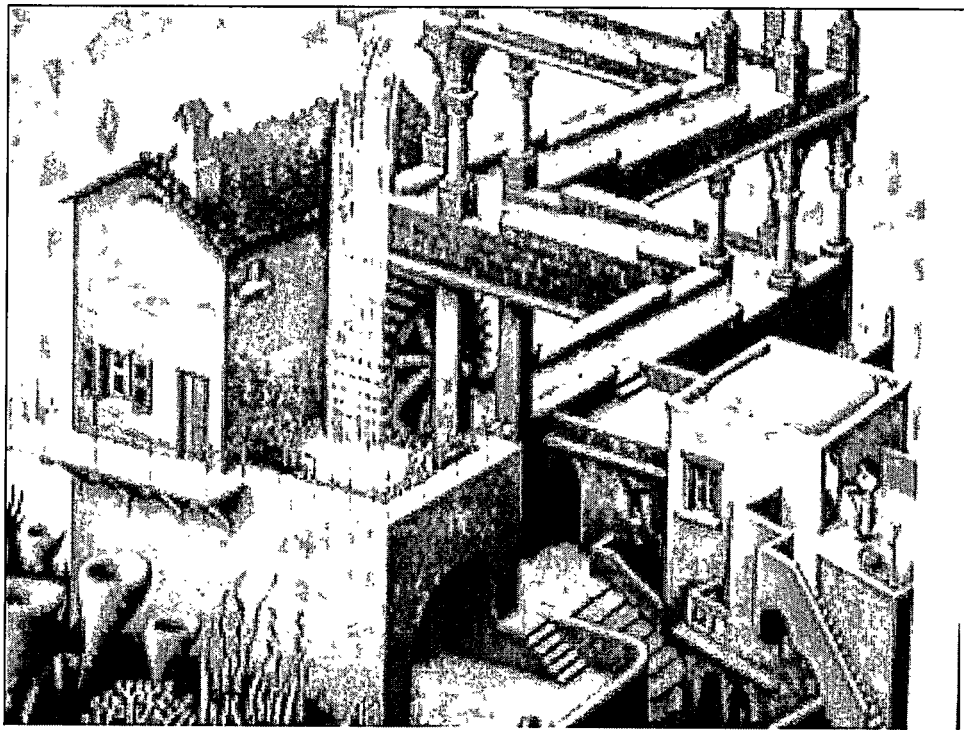


Figure 4.33: Bilinearly interpolated base image from Escher image set.

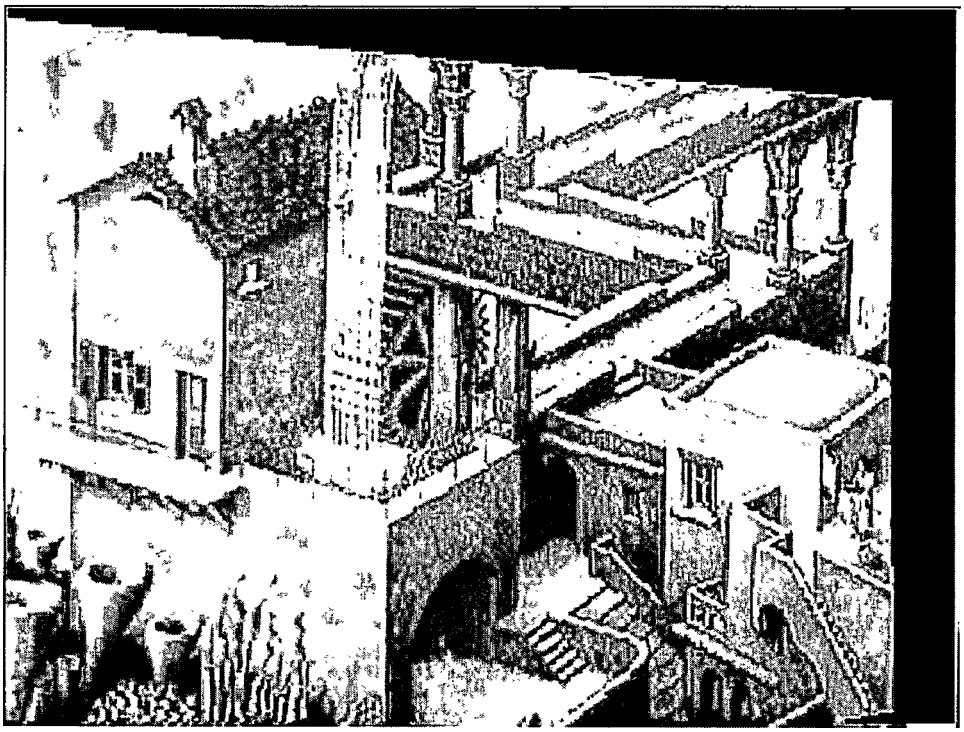


Figure 4.34: POCS restored base image from Escher image set.

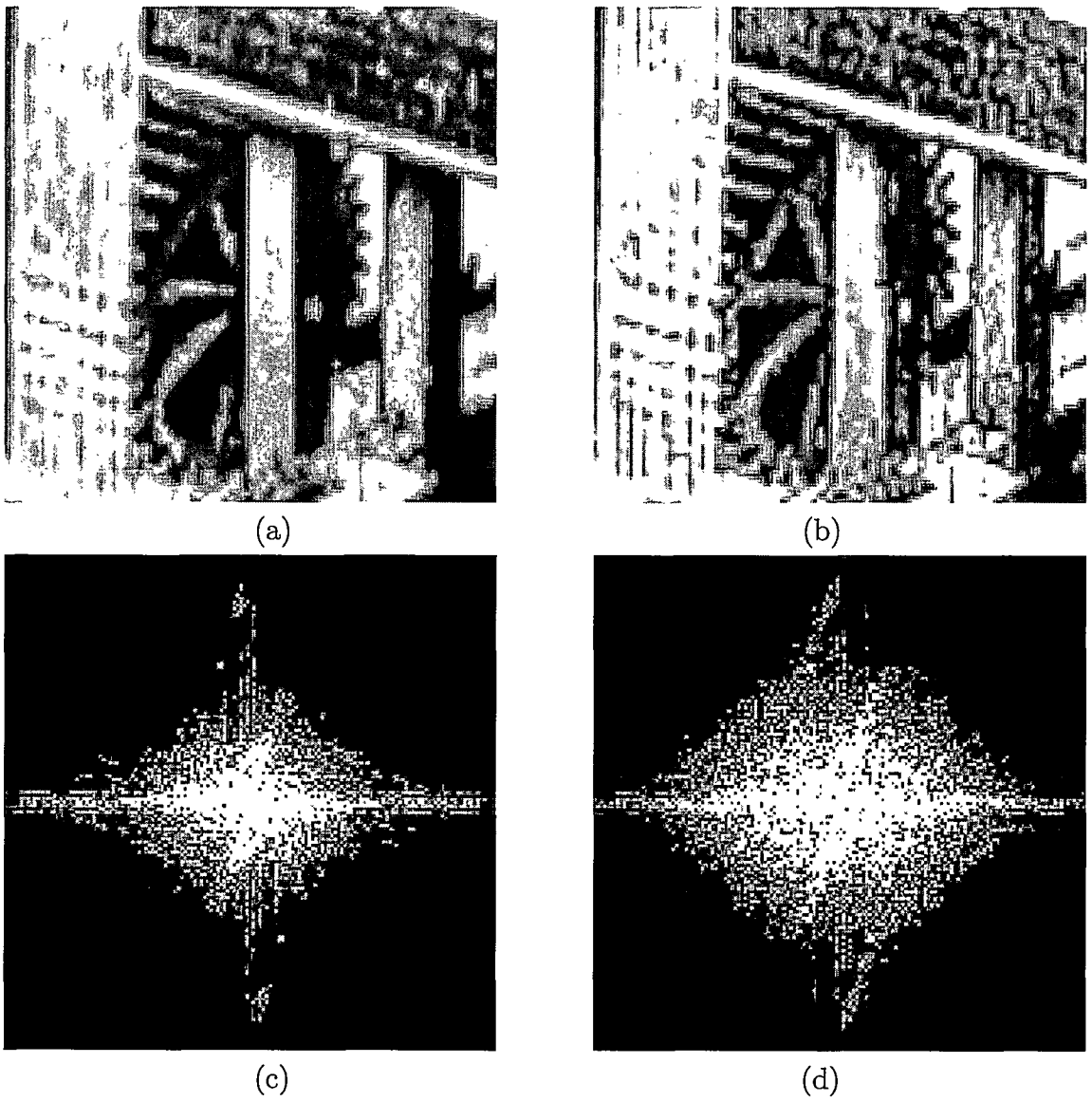


Figure 4.35: **Crops and FFT analysis of bilinearly interpolated and POCS restored Escher image set** *: Crops of bilinearly interpolated and POCS restored Escher image [(a) and (b) respectively]. FFT analysis on the cropped segments [(c) and (d) respectively].*

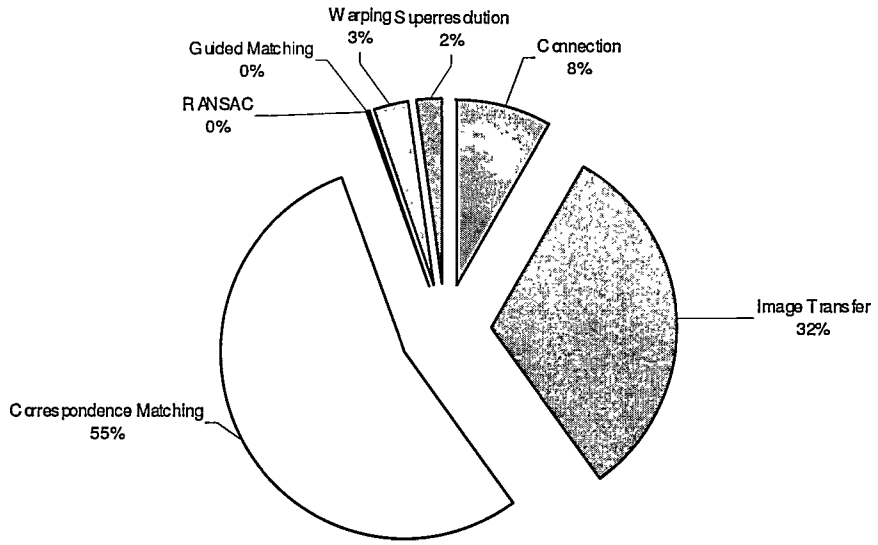


Figure 4.36: Timing Analysis for Escher image set

Table 4.8: Timing for All Image Sets

Procedure	**Res.Chart(s)**	**Castle(s)**	**Escher(s)**
Connection	44.44	32.65	31.23
Image Transfer	112.99	116.76	115.92
Correspondence Matching	114.39	149.47	200.70
RANSAC	2.61	1.03	0.78
Guided Matching	0.34	0.34	0.36
Warping	9.74	8.53	10.99
Superresolution	8.73	8.78	8.64
Total	**293.24**	**317.57**	**368.62**

Chapter 5

Discussion and Future Work

5.1 Superresolution

It was demonstrated that SR techniques could be used to generate images of higher resolution than those of individual images being used. One restriction is the requirement that images taken by the camera nodes be related by a 2D projective transform or homography. This limits the system to viewing a scene with a single camera in a panning motion, or to multiple cameras viewing a planar surface. The case of a panning camera does not lend itself to the methodology of the system, and therefore only planar surfaces can be viewed. One advantage however, is that specific camera parameters need not be known to find the 2D projective transformation that relates them. This alleviates the need for cumbersome camera calibration and only requires the raw image pixels in order to proceed.

5.2 System Performance

The systems performance was largely dominated by the image transmission and correspondence matching times. The 100 Byte packet length is an experimental limitation and with faster transmission speeds could be virtually eliminated. Further improvements can be attained from optimizing the correspondence matching algorithm. Currently it uses a brute force proximity and neighbourhood intensity metric which results in long search times.

5.3 Future Work

5.3.1 Implementation

The system is inherently modular from a hardware viewpoint as well as a software viewpoint. Choices were made based upon cost and readily available parts on the hardware side, and simplicity and 'ease-of-coding' on the software side. Some components can naturally be interchanged with competing or improved hardware or algorithms.

The cameras chosen use a proprietary compression scheme to transmit images over a USB connection. This compression could not be examined and therefore possible compression artifacts could have been introduced into the system. Using cameras capable of RAW output may lend to better performance of the superresolution algorithm being used.

Another aspect of the system that can be readily changed is the wireless communication standard. Having an abstracted API that is optimized for object exchange could enhance the transmission process of large objects across the Bluetooth link. Bluetooth transmission rates are currently restricted to 1 Mbps. Other wireless technologies are capable of higher transmission rates with increased range capabilities.

This would allow for the base low-resolution images to be of modest size, with super-high-resolution output at the vision server.

Various superresolution techniques exist that could be interchanged with the POCS implementation. Regularized SR reconstruction techniques are among those, outlined in section 1.2, that can also superresolve an image.

5.3.2 Towards an Embedded Smart Camera Platform

The camera node workstations in the vision system consist of an entire workstation and their sole use is for image capture purposes. This setup provide experimental convenience and in practice, tethering the camera and communication device to a workstation would be impractical. An embedded platform to house the camera would be used. This camera platform would house the imaging device, and be capable of wireless communication. Additionally, image processing algorithms such as feature point extraction, could be implemented directly on board to alleviate the load on the vision server. Currently, all computations occur at the vision server which introduces a single global bottleneck. Offloading the computations that can occur in parallel would make the system more efficient. The superresolution algorithm requires an image set to be registered onto one base image. To produce the homography which registers one of the lr_i images, only the base image is required. With knowledge of the base image, the homographies for the other images could also be computed at the respective camera nodes and in parallel.

Bibliography

[1] A.J. Patti; Y. Altunbasak. Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants. *IEEE Transactions on Image Processing*, 10(1):179–186, January 2001.

[2] R.C. Hardie; K.J. Barnard; E.E. Armstrong. Joint map registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6:1621–1633, December 1997.

[3] Brent A. Miller; Chatschik Bisdikian. *Bluetooth Revealed.* Prentice Hall PTR, 2001.

[4] M.A. Fischler; R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.*, 24(6):381–395, 1981.

[5] M.C. Escher. *Waterfall.* ©.

[6] M. Elad; A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–1658, December 1997.

[7] M. Elad; A. Feuer. Superresolution restoration of an image sequence : adaptive filtering approach. *IEEE Transactions on Image Processing*, 8(3):387–395, March 1999.

[8] United States Air Force. *1951 USAF Resolution Test Pattern.* ©United States Air Force, c/o http://medfmt.8k.com/mf/resolution.html, 1951.

[9] P. Cheeseman; B. Kanefsky; R. Kraft; J. Stutz; R. Hanson. Super-resolved surface reconstruction from multiple images. Technical Report FIA-94-12, NASA Ames Research Centre, Moffett Field, CA, December 1994.

[10] R.Y. Tsai; T.S. Huang. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing*, 1:317–339, 1984.

[11] S.C. Park; M.K. Park; M.G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, May 2003.

[12] A.K. Katsaggelos, editor. *Digital Image Restoration*, volume 23. Heidelberg, Germany : Springer-Verlag. Springer., 1991.

[13] R. Kirsch. *Schloss Neuschwanstein Photo.* ©Wizard & Genius Idealdecor - Zurich Switzerland, 1997.

[14] B.K. Gunturk; Y. Altunbasak; R. Mersereau. Gray-scale resolution enhancement. *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on.*

[15] A. Noble. *Descriptions of Image Surfaces.* PhD thesis, Department of Engineering Science, Oxford University, 1989.

[16] H. Stark; P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal Optical Society of America A*, 6:1715–1726, 1989.

[17] M. Irani; S. Peleg. Improving resolution by image registration. *CVGIP:Graphical Models and Image Processing*, 53:231–239, May 1991.

[18] A.M. Tekalp; M.K. Ozkan; M.I. Sezan. High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 3, pages 169–172, 1992.

[19] M.K. Ozkan; A.M. Tekalp; M.I. Sezan. Pocs-based restoration of space-varying blurred images. *IEEE Transactions on Image Processing*, 3(4):450–454, July 1994.

[20] C. Harris; M. Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[21] T.K. Moon; W.C. Stirling. *Mathematical Methods and Algorithms for Signal Processing.* Prentice Hall, 1999.

[22] S.P. Kim; W.Y. Su. Recursive high-resolution reconstruction of blurred multiframe images. *IEEE Transactions on Image Processing*, 2:534–539, October 1993.

[23] A.J. Patti; M.I. Sezan; A.M. Tekalp. High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur. In *IEEE International Conference Image Processing, Proceedings.*, volume 1, pages 343–347, November 1994.

[24] A.J. Patti; M.I. Sezan; A.M. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Transactions on Image Processing*, 6(8):1064–1076, August 1997.

[25] A.M. Tekalp. *Digital video processing.* Prentice-Hall signal processing series. Prentice-Hall, 1995.

[26] S.P. Kim; N.K. Bose; H.M. Valenzuela. Recursive reconstruction of high-resolution image from noisy undersampled frames. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1013–1027, June 1990.

[27] D. Capel; A. Zisserman. Computer vision applied to super resolution. *IEEE Signal Processing Magazine*, 20(3), 2003.

[28] R. Hartley; A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.