This is to certify that the

thesis entitled

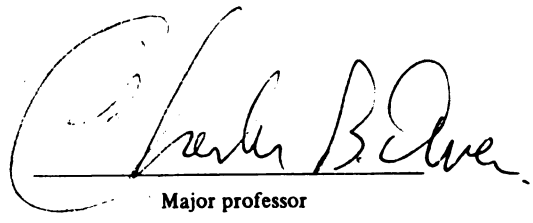
VISION-BASED TRACKING OF FIDUCIALS
FOR AUGMENTED REALITY

presented by

PAUL W. MIDDLIN

has been accepted towards fulfillment
of the requirements for

M.S. degree in COMPUTER SCIENCE



Major professor

Date 12/13/02

VISION-BASED TRACKING OF FIDUCIALS FOR AUGMENTED REALITY

By

Paul W. Middlin

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTERS OF SCIENCE

Computer Science

2002

ABSTRACT

VISION-BASED TRACKING OF FIDUCIALS FOR AUGMENTED REALITY

By

Paul W. Middlin

Visible fiducial images are a common method for supporting vision-based tracking in augmented reality systems. This thesis describes algorithmic improvements in fiducial-based tracking including an improved fiducial design, better fiducial location, and improved pose computation. A set of criteria that are desirable in an optically-tracked fiducial are presented and a new fiducial image set is designed that meets these criteria. The images in this set utilize a square black-border pattern with a 15% border width and an interior image that supports orientation determination and unique identification. The interior image is constructed from orthogonal Discreet Cosine Transform basis images chosen to minimize the probability of misidentification and to be robust to noise and occlusion. This image could be integrated into an Augmented Reality software system such as the well-known and widely used ARToolKit to improve accuracy in identification of fiducials.

Fiducial tracking involves more than simply creating a good fiducial image. The tracking includes methods to accurately locate the fiducial in the image, then use this information to calculate the location and orientation of the fiducial in relation to the camera. This location and orientation is known as *pose*. The ability of this system to track and calculate the pose of fiducials has been evaluated and compared to the ARToolKit as well. The system has proved to be generally better than the ARToolKit in terms of locating, identifying, and calculating the pose of a fiducial.

Dedicated to Stuart Griffin, whose ambition inspires us all.

ACKNOWLEDGMENTS

I would like to thank Dr. Charles Owen for his extensive contributions in the implementation of this project, as well as for his advice and direction.

I would also like to thank the students of CSE891, Augmented Reality, whose ideas and discussion led to the adoption of the fiducial criteria used, and the eventual DCT method itself.

Thanks go to Tony Lambert for helping set up the testing environment, and for the use of his laptop and truck.

Michael Malinak was helpful when doing the background research necessary for this thesis.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 Introduction

1.1 Background

Augmented reality (AR) is the blending of computer-generated virtual elements with reality [1]. A common example AR application is rendering computer graphics onto existing imagery such that the graphics appear to be seamless additions to or augmentations of the real image, registered in space, matching in scale. One of the most difficult challenges in this application is aligning the real and virtual worlds so as to achieve this seamless registration. The parameters of the rendering environment must exactly match those of the camera system that captured the image. Vision-based tracking uses images of the world to support this computation, either through tracking of natural image features [2, 3] or through the use of markers or fiducials placed in the scene. This thesis proposes a set of criteria to use when designing a fiducial and a vision-based tracking system for the fiducial design, making arguments for a specific type of fiducial that was created with optimization of these criteria in mind. Further, a system has been designed that uses these fiducials for tracking, which has been optimized for performance in a way that is consistent with the fiducial criteria.

Existing fiducial tracking systems use ad-hoc fiducial images based on either comparison to a library of template images or simple bar-code-based mechanisms. The designs are typically based on human, not machine, identification and ease of identification at high resolutions. The images tend to be highly correlated and often are misidentified.

This thesis recognizes the need for a set of fiducial images that can be systematically produced, has a small chance of being misidentified, and can be easily and accurately tracked. These images are two-dimensional forms of the Discrete Cosine Transform (DCT) basis set. The shape, border width, color, and method of locating the fiducial have been chosen after analysis of the criteria set forth in this thesis. The choices were made in an attempt to satisfy the general majority of fiducial tracking needs, though they will not be ideal for all situations. The fiducials will utilize a square shape with a black border that is 15% of the width of the fiducial. The interior images are monochrome and are based on the DCT basis set, with an orientation component built in.

The choices made in the design of the tracking system are shown to be theoretically superior choices. However, theoretical superiority is not a guarantee of performance in a real implementation. To verify the performance of this system, it has been compared to the ARToolKit [4] with tests in pose calculation accuracy and fiducial identification. The ARToolKit was used as a benchmark since it is one of the most popular and widely used fiducial tracking systems for Augmented Reality. The ARToolKit will in fact be mentioned numerous times throughout this thesis as a basis for performance comparison.

The testing has shown that the system created for this thesis was more capable both in terms of fiducial identification and pose estimation. Additionally, this system executes much more quickly than the ARToolKit system, allowing more time to do the three-dimensional rendering required in most AR applications.

The need for such an improved system stems from the wide variety of AR applications that use such technology. For instance, Fjeld and Voegtli [5] have created a

system that uses fiducials to allow a user to view chemical models in a more interactive way. A series of fiducials are used to identify different chemical compounds, and a graphical overlay of a model for these compounds is placed over the fiducial. This addition of graphics is done on a viewscreen. The user can interact with the models by moving the fiducials, or by using another cube that has fiducials on it. This cube can be rotated with a person's hand, and will cause the chemical model to rotate synchronously.

Using a viewscreen is not the only option for displaying Augmented Reality. Some systems use a Head-Mounted Display (HMD), which is like having two small monitors in front of the user (one for each eye). HMDs come in two major forms: video see-through and optical see-through. A video see through HMD uses a camera to record video, then passes this video to the eye with small LCD displays. In this case, the user is seeing the video as the camera(s) see it. For fiducial tracking, this means that the fiducials could be replaced with a virtual element before the video is seen by the user, thereby augmenting the reality that the user sees.

An optical see-through display is similar, but the user can see the world directly through the HMD. Here, graphics are overlaid using a half-silvered mirror and LCDs to combine the computer display's light with light coming from the actual objects. Again, fiducials could be used to calculate where the user's head is in relation to the objects he or she is viewing so that the virtual elements can be registered with the real objects in the user's line of sight. This is also an example of a time when the fiducials being tracked do not need to be in the same space as the virtual elements. That is, a separate camera can be used strictly to track the user's HMD, so the fiducial on the HMD would never be seen by the user; the fiducial is never shown on the HMD display.

Fiducial tracking can be extended to many other media, such as video monitors, handheld devices, or systems that do not use visual representations at all. The purpose behind using the fiducials is for tracking, which implies that any application in which the location and orientation of an object needs to be known can benefit from vision-based tracking.

1.2 Contributions

Contributions of this thesis are as follows:

- A set of criteria that define the qualities of a good fiducial tracking system
- A set of fiducial design choices that optimize those criteria
- A specific set of fiducial images based on the DCT that perform well with respect to the given criteria
- A system implemented using these fiducials, optimized for performance
- Testing of the system and comparison to a well-established fiducial tracking system (ARToolKit)
- Evaluation results that demonstrate improved accuracy, stability, and reliability.

1.3 Outline of Chapters

Chapter 2 outlines a representative set of existing vision-based fiducial tracking systems. Chapter 3 describes a set of criteria created based on the needs of such systems as those described in Chapter 2. Chapter 4 utilizes these criteria to derive a new type of fiducial that performs well relative to those criteria. Chapter 5 describes this system in detail and Chapter 6 presents evaluation of the system performance. The fiducial system

created is still not necessarily ideal, and ideas for the improvement of this system are described in Chapter 7.

2 Related Work

Fiducial-based tracking is a key enabling technology for a wide variety of applications. The motion-picture industry uses fiducials to track camera movement in support of augmented imagery. Manufacturing applications track fiducial images on circuit boards and other components so as to support accurate assembly alignment. Because of this general utility, there are many fiducial systems that have been proposed in both commercial and research areas. Some systems exist only to support location of a single point in an image. Others support only two axis of alignment for parts placement. Only a limited number of systems support full pose computation as described in this thesis. This chapter describes a set of the major systems described in the literature.

2.1 ARToolKit

One of the most well known and widely used fiducial tracking systems is the ARToolKit. It was created by H. Kato and M. Billinghurst at the University of Washington in the Human Interface Technology Lab [4, 6] and supports full pose calculation in addition to identification of a set of fiducials. The ARToolKit is widely distributed as open source for a variety of target platforms. Between its free distribution, documentation, and ease of use it has become the center of a wide variety of AR applications that depend on vision-based tracking. It is used both for research and commercial use, and for development of other systems in the form of its compiled tracking libraries.

ARToolKit markers are square fiducial images with a fixed, black band exterior surrounding a unique image interior. Figure 2-1 is an example ARToolKit fiducial. The

outer black band contrasts against a light background and is used to locate a candidate fiducial in a captured image. The interior image enables the identification of the candidate from a set of expected images and determination of the four possible orientations. The four corners of the located fiducial are used to unambiguously determine the position and orientation of the fiducial relative to a calibrated camera.

Figure 2-1 - Example ARToolKit Fiducial

Design of the distinquising interior image is completely up to the user. This content is ad hoc, in that there is no systematic process to generate it or to choose good alternatives. Frequently, single letters or numbers are used.

The ARToolKit requires several steps to find and match a fiducial image. The image is thresholded against a constant value and all connected components are labeled. The edges of the connected regions are located using contour following. These contours are then fitted to lines to form a quadrilateral. If a quadrilateral is found, then the pixels

in this quadrilateral is resampled into a 16x16 upright square image that is to be compared with the fiducial patterns registered with the system.

The comparison is done by calculating the correlation coefficient between the captured candidate image and a stored template pattern. In the following equations, I(x,y) is the candidate image and P(x,y) is the pattern.

First, the mean and standard deviations for the image and pattern are computed (clearly the pattern data can be pre-computed). The following equations show how to compute the standard deviation for the candidate image (σ_I) and for each pattern (σ_I). μ_I and μ_P from equation (2-1) and (2-2) are just substitutions into the standard deviation equations in (2-3).

$$(2\text{-}1)\quad \mu_I = \frac{1}{xy}\sum_x\sum_y I(x,y)$$

$$(2\text{-}2)\quad \mu_P = \frac{1}{xy}\sum_x\sum_y P(x,y)$$

$$(2\text{-}3)\quad \sigma_I = \left(\sum_x\sum_y (I(x,y)-\mu_I)^2\right)^{1/2} \qquad \sigma_P = \left(\sum_x\sum_y (P(x,y)-\mu_P)^2\right)^{1/2}$$

Then, the correlation coefficient (ρ) is computed as:

$$(2\text{-}4)\quad \rho = \frac{\sum_x\sum_y (I(x,y)-\mu_I)(P(x,y)-\mu_P)}{\sigma_I\sigma_P}$$

The correlation coefficient is a non-negative value such that larger values indicate similarity of the image based on an L_2 norm. If the coefficient for one image is maximal for the image set and exceeds a fixed threshold (0.5), then the image is accepted.

Obviously, this process is a complex calculation. More importantly, using this process means that to find a best match the system must calculate a coefficient between the candidate image and each of the expected patterns, an O(N) operation. The more patterns in the system, the longer it will take to perform this calculation. The ARToolKit actually has a hard limit as to the number of fiducials that can be registered with the system. This limit helps to prevent it from taking too long to match the fiducial, but limits the flexibility of the system because of the small number of fiducials that can be used.

2.2 CyberCode

The CyberCode system was created at Sony Computer Science Laboratories [7]. CyberCode is based on a two dimensional bar code fiducial. Here, the interest was more in producing a large number of unique fiducials. A CyberCode fiducial consists of a square area for the patterned code, with a black bar alongside the square region to help determine orientation. There is no surrounding border as with the ARToolKit fiducials.

Figure 2-2(a) shows an example of a CyberCode fiducial. The guide bar is pointed out in (b). The four corners of the square area are always black (c), so the code pattern is the cross-shaped area inside of this (d).

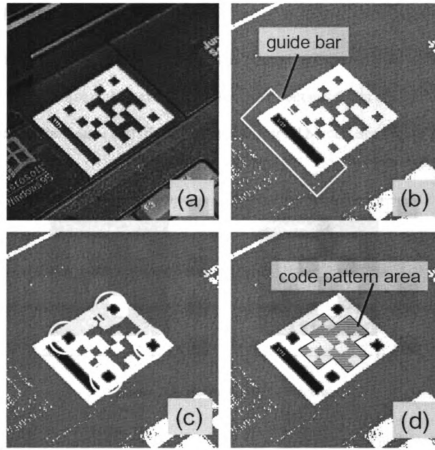


Figure 2-2 - CyberCode recognition steps

The tags are found by adaptive thresholding the image, then applying a connected components algorithm. The connected regions are then searched a specific second order moment, indicating the guide bar. From there, the algorithm locates the four corners, and uses these locations to account for distortion from tilt/angle. The last step is of course to decode the bitmap inside the four corners.

Sony claims to be able to use 24 bits to encode the identification, meaning that there are over 16 million possible CyberCode markers. This is a very wide space. Sony published little about the performance of this system in terms of adaptability to different lighting conditions, low resolution images, or 3D location accuracy.

2.3 HOM System

Similar to CyberCode, the HOM system created by Siemens uses a 2D code with a side bar [8]. In this case, however, the sidebar also contains 6 bits of additional coding information and the square part of the fiducial has a solid border. See Figure 2-3 as an example.

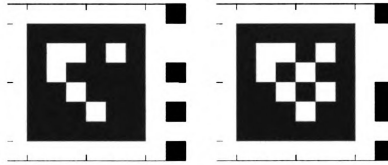


Figure 2-3 - Example HOM Fiducials

2.4 IGD System

The IGD system is another coded fiducial system using a black border and a bitmap in the middle [9]. The IDG marker system was implemented at the Institute for Computer Graphics (Institut Graphische Datenverarbeitung) in Darmstadt, which is an ARVIKA partner. ARVIKA is the German government supported research project to develop AR-related applications in industry. Many ARVIKA-related applications are developed using the IGD marker system. An IGD marker is a square divided into 6x6 square tiles of equal size. The inner 4x4 tiles are used to determine the orientation and the code of the marker. Figure 2-4 shows an example of this fiducial. The precompiled libraries of the IGD marker system are available to ARVIKA participants [10].

Figure 2-4 - Example IGD Fiducials

2.5 SCR System

The SCR marker system was developed by Siemens Research Corporation for AR applications [11]. It also uses a coded matrix to identify the fiducial, as seen in Figure 2-5. Additionally, it locates 8 feature points instead of the usual 4 found in most square fiducial systems. The additional points might help to increase the accuracy of the location of the fiducial, which in turn can help make 3D translations more accurate.

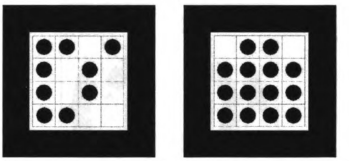


Figure 2-5 - Example SCR Fiducials

2.6 TRIP System

The TRIP (Target Recognition using Image Processing) system is a circle-based system. It was developed at Cambridge University in the Laboratory for Communications Engineering [12]. It uses a sector-based circular system of bar coding.

The innermost part of the target is a "bull's-eye". The bull's-eye is used to locate the fiducial. The TRIP algorithm thresholds the image, does edge detection, and then edge following. The connected edges are examined and only those that are circular (or ovular) are kept. Finally, the bull's-eye is identified when two concentric circles are found.

After finding the fiducial, the two concentric rings around the bull's-eye are examined. They are broken into 16 sectors, as shown in Figure 2-6. One of these is used as a synchronization sector; two others are used for even-parity. The remaining 13 sectors are used as a ternary code. There are therefore $3^{13} = 1{,}594{,}323 \approx 2^{20}$ possible codes.

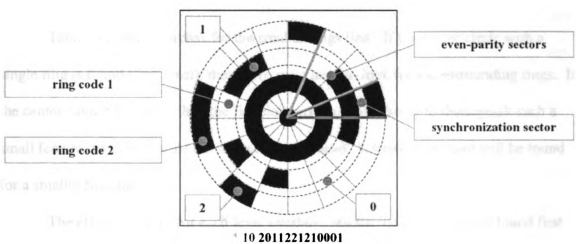


Figure 2-6 - TRIP Target representing 1160407

Despite providing only one real location point, the TRIP system does indeed calculate the 3D position of the target in relation to the camera. It does this using the POSE_FROM_CIRCLE algorithm described by Forsyth et al [13]. The synchronization sector is used to find the orientation of the circle.

2.7 Multi-resolution Colored Rings

Cho, Lee, and Neumann at the University of Southern California have created a system that uses nested colored rings [14]. The purpose is to make fiducials that can be found over a wide viewing range. Each fiducial consists of a center circle, then three rings of increasing width surround the center (Figure 2-7).

Figure 2-7 - Multi-size color fiducials

Their algorithm searches for the smaller rings first. If the center circle with a single ring is found (first level), then there is no need to look for the surrounding rings. If the center cannot be found, then the fiducial must be too far away to distinguish such a small feature, so it will locate the second level instead. Likewise the third will be found for a smaller fiducial.

The effective range for each level overlaps, but the smaller should be found first in the case of an overlap because this requires less processing time. The range of sizes for identifiable fiducials is about 24 to 56 pixels in diameter.

It should also be noted that each fiducial returns only a single point, so any calculation of 3D location would require 3 or more fiducials in the scene. In fact, using strictly the three points, there are often up to four solutions [15]. This implies that this system employs some extra processing between frames to rule out other solutions, or that it is sometimes inaccurate because of the lack of a fourth point for correspondence.

2.8 Other Systems

Many simple approaches using fixed color squares, circles, or cross patterns have been demonstrated. Most projects approach the problem either from the standpoint of selecting a set of images (as in ARToolKit) or choosing a way to encode data into images (as in CyberCode). There are a plethora of other systems that do fiducial based tracking. See the following references: [14, 16-19].

2.9 Pose Calculation Methods

Pose is the location and orientation of an object. The location implies three degrees of freedom – a point on an 'X', 'Y', and 'Z' axis. This alone does not reveal the way the object is situated at that location, so the orientation component is needed as well. Orientation is three degrees of freedom as well – rotations about the X, Y, and Z axes. Therefore, pose involves six degrees of freedom.

There are many methods for calculating the 3D location of points in relation to a calibrated camera given the screen coordinates of these points and the model of the object. It is assumed that a single fiducial (or a set of fiducials in some systems) represents a coordinate system. It is typical that a single corner of a fiducial image will be declared to be the origin of the system and all points are considered to be in the (x,y) plane. Any

fiducial tracking system used for AR must use some form of pose calculation to estimate the 3D location of the fiducial. Three particular methods are examined in this thesis (see Section 5.9), but there are many methods in existence. The methods described here relate mainly to those presented by Shapiro and Stockman [20] and that used by the ARToolKit [4].

It seems valid to mention the work in this area by Ji et al [21] which describes methods for doing pose calculation from a variety of geometric shapes. Also important is the work of Quan and Lan [22], who have developed a linear method for pose calculation (instead of an iterative approach as is described in Section 5.9). This method solves the systems of equations using the classical Sylvester resultant [23] and quaternions. This solution is not a perfect least-squared solution, it is an estimate. See also [24-29] for examples of other methods and applications in the subject of pose calculation.

3 Criteria for a Good Fiducial

Clearly there are tradeoffs among the criteria for a good fiducial image. Existing designs for AR fiducials have been ad hoc and have not started with specific design criteria other than support for some level of tracking (planer, pose, etc.). This thesis approaches the problem by asking question and proposing answers consistent with many applications in augmented reality and commonly available hardware. The questions addressed in this section are:

- What is a good fiducial shape?
- What colors should be utilized in a fiducial image?
- How should a specific fiducial be located in an image?
- How should a specific fiducial be identified?
- Over what range of sizes should the fiducial be identified?
- Should a human be able to decode/identify a fiducial?

The answers to these questions can vary depending on the application or domain that the fiducials will be used in. Some applications may require fiducials with anthropomorphic characteristics; others may be optimized for computer tracking only. Care will be taken, however, to try to make the answers to these questions as generally applicable as possible. Additionally, points that may influence one's decision on the best choice to meet a given criteria will be presented to help make this decision.

3.1 Fiducial Shape

The purpose of a fiducial image is to provide automatic correspondences between points in a camera frame and points in a captured image. Clearly, any visual feature can

be used as a fiducial if its location is known (or can be computed) and it can be automatically identified. Indeed, tracking systems designed for use in unprepared environments have been proposed that use regions, lines, and other natural environmental features [30, 31]. Most applications for fiducial images, however, assume a prepared space with specific images placed in the environment, with the assumption that the relative transformation between a camera frame and frames indicated by the fiducials needs to be determined. In tracking terminology, the position and orientation (six degrees of freedom) of the frame marked by fiducials needs to be identified relative to the camera. This problem is also commonly referred to as *pose estimation.*

Determination of position and orientation of a physical object relative to a camera frame requires the correspondence of at least four non-linear points. As an example, estimating the pose of a camera relative to a physical environment will require the identification of four 2D points in the camera image and knowledge of their 3D coordinates in the world coordinate system. It is possible to compute pose from only three points. However, the result is ambiguous, generally having two, and often three or four, solutions [15]. Hence, any ideal fiducial solution supporting 6DOF pose estimation should always emit a minimum of four located points, no three of which are colinear. Additional points can be used to compute least-square solutions that can average out errors and increase the estimate's accuracy. Many fiducial methods utilize a single, typically very simple, fiducial image such as a ring or disk with the requirement that multiple fiducials must be simultaneously tracked [14].

Since the location of fiducials in camera images will always be permuted by noise and quantization error, there is a clear advantage to tracking additional points, so

fiducials that emit multiple tracking points seem advantageous. Also, many applications require tracking of styli, independent marked locations, or multiple users, where placement of a large number of fiducial images is prohibitive.

An assertion of this thesis is that an ideal fiducial image should emit at least four points. Beyond that, it is clear that the points should approximate a square. The size of the fiducial equates to resolution in the capture image. Four points not in the form of a square will result in some elements of the image presenting a lesser resolution to the camera than others, thereby decreasing tracking accuracy in corresponding orientations.

This requirement does not necessarily imply that the fiducial image itself must be square. Any image that can emit four points would suffice. However, there are clear computational advantages to simplicity, and a square fiducial image is the simplest possible fiducial emitting four points. The straight edges of a square can be used to compute best-fit lines allowing corners to be computed with greater, potentially sub-pixel accuracy. Indeed, the ARToolKit standard fiducial image is a square image.

It should be noted that a circular marker can be used to determine pose if a point on the circle can be determined. The POSE_FROM_CIRCLE algorithm provides a robust solution given circle edge points [13]. However, an interior image for identification is more difficult to implement and cannot be represented in a rectangular array. Most implementations based on pose estimation from circles are based on barcodes (or, more precisely, ringcodes) [12].

3.2 Fiducial Color

The question of fiducial color is much more difficult to address. Clearly, choosing a color fiducial as opposed to monochrome increases the possible set of fiducial images. Indeed, both color and monochrome images have been utilized in existing systems. However, there are several technical reasons to favor a monochrome fiducial:

- Varying chroma resolution in camera systems
- Decreased image representation
- Higher-performance localization algorithms

The spatial frequency sensitivity of the human visual system for luminance components is much greater than for chrominance components [32]. Unfortunately, many imaging systems designed for computers mimic this characteristic, transmitting chrominance information in lower bandwidth channels or representing chrominance information with lower resolution. This necessarily decreases the detection resolution for color fiducials. Use of inexpensive web-cams has become very popular for fiducial-based tracking. These cameras clearly exhibit decreased color resolution. Hence, for the most accurate results using a wide range of cameras, a monochrome fiducial image is the best choice. When high-quality cameras are available, color fiducial images can increase the information available in the fiducial image.

Even if an RGB color presentation is captured at full resolution, the resulting color image will increase the memory usage and, consequently, the analysis time, by a factor of three (or four). This is a consequence of the increased memory bandwidth requirements.

An additional element in the choice of color or monochrome is the choice of localization algorithms. High-performance algorithms have been developed for color fiducials, but assume very simple shapes that can be identified by cross-sectional lines [14]. One advantage of color fiducials is the use of color to identify the specific fiducial, as in the multi-ring approach. However, the number of colors that can be uniquely identified varies greatly depending on lighting conditions, and is likely to be small. Specular reflection will not only affect the luminance of an image, but can also modify the hue of imaged colors. Additionally, the colors must contrast with colors naturally occurring in the scene.

One option for color is to utilize retro-reflective fiducials and infrared illumination [33] or direct imaging of infrared emitters [34]. This option is a very different technological approach from visible-image tracking, requiring special camera, illumination, and reflective technologies. In addition, IR fiducials based on retroreflective materials do not lend themselves well to patterned individual fiducials other than simple binary patterns. As the focus of this thesis is visible image fiducials, IR approaches are beyond the scope of this discussion.

3.3 Locating the Fiducial

The shape and color of a fiducial is directly related to the algorithm utilized to locate it in the camera image. As mentioned previously, the ARToolKit contains a fiducial tracking system using a square image with a black border as illustrated in Figure 2-1. An interior image contained within the border provides identification for the particular fiducial image. It is assumed that the marker will contrast with a surrounding region when converted to a binary image. Typically, this contrast can be achieved by

simply ensuring that the fiducial is mounted on a white surface or is printed on a larger white sheet of paper. More details of the ARToolKit approach will be included in later sections. Kato and Billinghurst [4] allow for the fiducial corners to be rapidly and accurately located in a camera image. The approach assumes a monochrome fiducial image.

Is this the best fiducial design for localization, the location of the fiducial in an image? There are several distinct advantages to this design. The shape is a square design and yields four corner points for tracking purposes. The edges are straight between the corner points. This allows the corners to be determined by line fitting to the edges, yielding measurements that are less sensitive to noise in the vicinity of the corner and quantization errors. The black border also yields a maximum contrast relative to the background, particularly a white background. Once the corners have been located, the interior can be warped to a common frame of reference (16 by 16 in the ARToolKit approach) for comparison to a database of marker images.

This fiducial approach does not emit an orientation other than through analysis of the interior image; hence, the offset of the interior text in the marker image in Figure 2-1. Would it be better to design the outline to emit orientation independent of the interior text? This design could be accomplished in a variety of ways, including offsetting the interior image, adding an orientation image in addition to the interior image, or using varying colors on the edge. Varying colors is not considered a good choice for the reasons mentioned in the previous section and because it would eliminate the homogeneity of the design. Detection performance would be determined by the least common denominator of detection of the two types of borders. Offsetting the image or adding an image

component for orientation is equivalent to using a larger interior image and determining orientation from the interior image alone. Figure 3-1 illustrates this equivalence. When either the interior image is offset or a special orientation pattern is added, the fiducial can be considered equivalent to a simple border with a larger interior image, as indicated by the dotted lines.

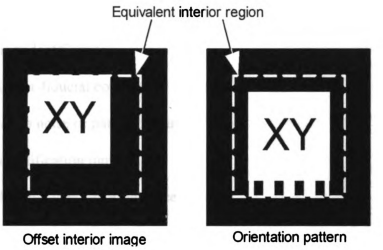


Figure 3-1 - Equivalence of interior images for orientation determination

Given these criteria, the square ARToolKit fiducial outline seems to be a "good" approach. The border width and the interior image will be adjusted in this research, though.

3.4 Fiducial Identification

Once an individual fiducial image is located, it must be identified. The identification of the interior image is simplified if a border has been located. The interior image can then be warped to a square image with a fixed scale.

Clearly, marking a space with identical fiducials would require the analysis of relative placement for identification, so it is advantageous if fiducials are unique. Uniqueness can be accomplished in a variety of ways, including color combinations, bar codes, or patterns. The pattern must be unique and accurately identifiable at a variety of resolutions. Several desirable characteristics for fiducial identification have been collected:

- Orientation identification
- Minimal inter-fiducial correlation.
- Resistance to noise or partial obscuring.
- A large identification range.
- A large fiducial identification space.

As discussed, using a fixed monochrome square image, as in the ARToolKit fiducials, is a preferred method. The identification image is then set inside this box. It is also preferred that the orientation, and thereby the correspondence of detected image corners with physical coordinates, is determined by an interior image. Consequently, the image must support determination of a unique orientation. In ARToolKit, fiducials are commonly designed with offset text or blocks that make the orientation unique. Then a candidate image is compared to the known images in each of the four possible orientations. This method of comparison necessarily limits what can be selected as a fiducial, particularly if users desire fiducial images with visually perceptible meaning.

A key characteristic of fiducial images is that there is minimal inter-fiducial correlation in all orientations. A variety of methods are possible for comparing images.

Mean squared error (MSE) is a common measure of image similarity, particularly when measuring image degradation:

$$(3\text{-}1) \quad c(I,P) = \left(\sum_x \sum_y (I(x,y) - P(x,y))^2 \right)^{1/2}$$

In this equation, I(x,y) is the candidate image, P(x,y) is the pattern, and c(I,P) is a measure of the dissimilarity between the two. For an MSE measure, small values indicate similarity. This approach is not luminance invariant, however. A better approach is the correlation coefficient. This is the approach that the ARToolKit uses, and was explained in detail in Section 2.1.

Clearly, no guarantees can be made about inter-fiducial correlations when images are chosen ad-hoc. As an example, consider the fiducial set in Figure 3-2. The Hiro and Kanji images (first two images in the first row) are standard fiducials included with ARToolKit. The remaining images illustrate an obvious idea of using alphabetic characters as interior images.

Clean, computer-generated images were compared using the correlation coefficient. The Hiro pattern had a worst case correlation to the A pattern of 0.163. The Kanji pattern has a worst case correlation to the A pattern of 0.498, just below the standard threshold. The G pattern correlates to B with 0.637 and C with 0.820, both far above the identification threshold. Obviously, the letters have too little difference to be good choices, but even the Hiro and Kanji fiducials have correlations of 0.204. In a test by Zhang et al [10], the ARToolKit system identified a fiducial with the pattern "3" with an 85% confidence as being a "2", while at the same time giving only a 69% confidence that the actual "2" pattern was "2". Clearly, the images that identify a fiducial must be

carefully chosen or identification errors will result due to correlations among the candidate set. If two possible candidate fiducials high correlation, it becomes difficult to distinguish between them in a real application.

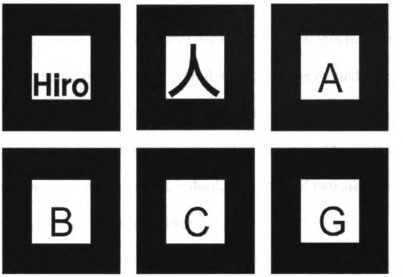


Figure 3-2 - Example images for correlation tests

This problem is accentuated in the presence of noise. Fiducial images need to be robust in the presence of noise and partial occlusion. This need implies a potential drawback of the ad hoc choices in Figure 3-2. Were a small part of the G obscured, it would be indistinguishable from the C. This is even more of an issue when bar-codes are applied to fiducials [7, 12]. The TRIP system, for example, requires 15 unique regions in the cross-section of the image center. If reduced to a size of 25 pixels across, most regions are one or two pixels and would be difficult to detect with edge detection algorithms. Small errors will change the code, violating the ringcode parity and rejecting the marker or falsely identifying it.

One way to describe this problem is that the features that make a fiducial unique from the set are often highly decorrelated in the image. This places a high percentage of

the information content into a minimum set of pixels, making the system much more sensitive to perturbations of those pixels.

3.5 Fiducial Identification Range

The identification range of a fiducial depends on the camera resolution and camera parameters. Some systems have been designed to have redundant identifiers at multiple scales, so that larger images become available as the camera moves beyond the range of smaller images [14]. Clearly, the same effect can be had by creating multiple fiducials in the space of varying size and, indeed, size ratios of two are shown by Cho, Lee, and Newmann to be an effective choice.

For the purposes of this research, the concern is with how small the image of a fiducial may become and still be reliably recognized. This size determination is primarily dependent on the native size of the identification image. To be consistent with current systems like ARToolKit and with the goal of having a small fiducial, a 16x16 identification image size has been chosen. Therefore, the minimum dimension of the identified image in any axis must be 16 pixels. This is not the size of the actual fiducial, but rather the minimum size of the interior image.

Given this criteria, the question arises: how wide should the border be? To ensure reliable outline location, the border must be wide enough to ensure the point spread function of some pixel on the border will cover the region at every point along the border. If the border is too narrow, the border may fall between pixels, leaving the pixels a shade of gray too indistinct to allow edge following. So, the edge must be wider than twice the distance between any two pixels. This distance is actually 2.83, because the worst case distance between pixels is 1.41 for diagonal lines. Hence, the image must be

at least 16 + 2.83 + 2.83 = 21.66 pixels wide in the recognized image. For design purposes, this implies that the border must be at least 13% of the fiducial width. To be conservative, a 15% border width was selected. Note that the border width should be kept minimal in order to increase the size of the interior image and allow for a larger recognition range.

3.6 A Large Fiducial Identification Space

The size of a marked space and the number of marked implements in that space is limited by the number of unique fiducials that can be applied to the space. Marking each two foot square ceiling tile in a twenty foot square room will require one hundred unique fiducials. Clearly, a desirable characteristic of fiducials is a large space of identifiers. While some of the bar code solutions claim ranges in the millions, this range is dependent on recognition of a high-resolution code in camera images from varying distances. Consequently, the images must be relatively large.

A 16x16 image can have up to 256 patterns that are orthogonal to each other, if the minimal correlation criterion is desired, though the set is easily expanded to 512 if maximum negative correlation is also allowed. Treating those 16 by 16 images as a 256 binary value would significantly increase the number of possible fiducial images at the expense of highly correlated images.

3.7 Human Identification

It is sometimes advantageous to have fiducials that are anthropomorphic; can be easily identified by a human. For instance, if a set of fiducials corresponded to a set of

objects, then it would be easy for a person to pick up the object that they wanted because the fiducial could be something that implies that object. A fiducial with an "A" on it, could mean the "Antelope" object.

Identification by a computer, however, is much different. It may be the case that a set of fiducial images is more easily recognized by the machine than by a person. Since one of the main criteria is to create fiducials that are not easily confused by the machine, this research will not consider human identification to be an important capability for the fiducial system to have. Human identification can easily be added to any fiducial with a simple label near the fiducial. Additionally, fiducials are often replaced in an AR application by graphics, or could have this capability added to account for the human identification factor. The prevalence of bar-code systems clearly indicates that anthropomorphic characteristics are not important in many applications.

3.8 Summary of Desirable Characteristics

This is a summary of the chosen criteria as proposed in this thesis: An ideal fiducial image should support the unambiguous determination of position and orientation relative to a calibrated camera. The image should not favor some orientations over others. The image must be a member of a set of images that are unlikely to be confused such that a large space or set of objects can be uniquely marked. The image must be easy to locate and identify using fast and simple algorithms. Images must function over a wide camera capture range.

Given these criteria, arguments in this chapter have supported the design of a square fiducial (Section 3.1) with a black border (Section 3.3) 15% of the width of the image (Section 3.5) and some internal image suitable for identification of the fiducial

(Section 3.4). The image will be monochrome (Section 3.2) and will be designed without respect to human identification (Section 3.7). The next section will detail the design of a suitable interior image.

4 A "Good" Fiducial Interior Image

A "good" fiducial interior image set will have a large set of images from which to choose, a means for accurate orientation determination, and a fast algorithm for identification. A major contribution of this thesis is the design of a new fiducial interior image that supports these requirements. The main goal is to select images such that the correlation coefficient of any two images is minimal. The optimum selection, then, would be a set of images wherein correlation coefficients among any two non-equivalent images are null. Other obvious criteria are that the image can be represented using real-valued images (no negative or complex pixel values), and that the intensity be maximal (as bright as possible). This section describes the derivation of a new fiducial interior image based on DCT basis functions and an associated algorithm for efficient identification of instances of this fiducial design.

4.1 Deriving the Image

A common method for comparing two images and producing a measure of similarity is the use of the correlation coefficient. Of the alternatives, the correlation coefficient is least sensitive to noise and provides a single measure of image similarity. To get pattern images that are as different from one another as possible, the correlation coefficient between any two images should be 0. Recall equation (2-4) shows how to calculate the correlation coefficient. Setting the equation for the correlation coefficient to zero for two images I_1 and I_2:

$$(4\text{-}1)\quad \frac{\sum_x \sum_y (I_1(x,y)-\mu_{I_1})(I_2(x,y)-\mu_{I_2})}{\sigma_{I_1}\sigma_{I_2}} = 0$$

implies:

$$(4\text{-}2)\quad \sum_x \sum_y (I_1(x,y)-\mu_{I_1})(I_2(x,y)-\mu_{I_2}) = 0$$

This equation (4-2) will be satisfied if I_1 and I_2 each is the sum of a DC offset and a member of a set of functions such that the dot product of any two non-equivalent basis functions is zero. In other words, a good choice for fiducial interior images is a set of orthogonal basis functions scaled to a peak-to-peak range equal to the pixel intensity range and added to a DC offset sufficient to make the image non-negative.

There are a wide variety of basis function sets available. Most existing 2D linear transforms, including Fourier, Hadamard, Haar, and many others, emit sets of real-valued basis images. Among the 2D sets with real values, we have chosen to use the basis functions for the Discrete Cosine Transform (DCT), specifically DCT-II [35]. This N by N 2D basis function set is defined by:

$$(4\text{-}3)\quad B_{u,v}(x,y) = \cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

In this application, N=16. Alternative sizes could be utilized, though larger fiducial images would be required and the recognition range would be decreased.

One approach would be to construct a fiducial interior image as:

$$(4\text{-}4)\quad I_{u,v}(x,y) = \frac{B_{u,v}(x,y)+1}{2}$$

For simplicity, assume normalized pixel intensities in the range [0,1]. This interior image set supports 256 combinations of (u,v) as fiducial interior images, all of which have inter-correlation values of zero for non-equal interior images.

However, this solution does not satisfy one of the specified requirements: it does not directly include orientation information. In fact, the images with even (u,v) values are invariant under 180 degree rotation. The odd values could be utilized and do indicate orientation, but that would reduce the set of fiducial images by 75%.

A solution to this problem is to consider the image as the sum of three parts: the DC offset value (required to make a non-negative image), an orientation image, and an identification image. The orientation image is the (1,0) basis image:

$$(4\text{-}5) \quad B_{1,0}(x,y) = \cos\left(\frac{(2x+1)\pi}{2N}\right)$$

This basis image is a bit less than a half cycle of a cosine wave ($\pi/32$ to $31\pi/32$).

With proper scaling, the fiducial interior image is defined as:

$$(4\text{-}6) \quad I_{u,v}(x,y) = \frac{B_{u,v}(x,y) + B_{1,0}(x,y) + 2}{4}$$

When the orientation component and the mean of the image (the DC component) are subtracted, all images are orthogonal to each other, reducing the likelihood of false fiducial identification. An advantage of a DCT basis function as a fiducial image is that the pixels within the fiducial are highly correlated. This high correlation makes any correlation-based detection less sensitive to partial occlusions and noise. Whereas some fiducial systems store the information in edge data, particularly barcode-based systems,

or within bounded regions as in CyberCode, the DCT basis approach embeds the identification information in the entire interior image gradient.

The interior fiducial image equation assumes the creation of a 16 by 16 image. However, the images used in a room are much larger than 16 by 16. In practice, a fiducial image will be created at a high resolution for printing, sampled by the image capture system, resampled by the warping, and compared to the basis set. In this application, 3.5 inch square images printed by a 600dpi laser printer are commonly utilized. So, the analysis fiducial image (16 by 16) must be resampled to the printer resolution. The equation for creating a resampled fiducial image of arbitrary size is:

$$(4\text{-}7) \quad \hat{I}_{u,v}(x,y) = I_{u,v}\left(\frac{xN}{W} - \frac{1}{2}, \frac{yN}{H} - \frac{1}{2}\right)$$

In this equation, (x,y) are coordinates in a W by H image. This equation is used to create the fiducial image at high resolution. The one half pixel offset ensures that the high resolution image will properly resample if divided into 16 by 16 square regions and sampled in the center of the region. This is important to ensure the fiducial image is not offset.

Figure 4-1 illustrates several example fiducial images based on this system. The lighter characteristic on the left side is due to the orientation image component. The sinusoidal patterns of the basis functions are clearly visible in the images.

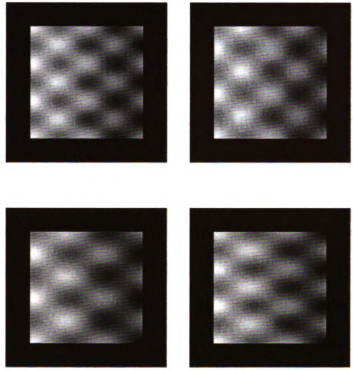


Figure 4-1 - Example DCT fiducial Images

4.2 Detection

The choice of DCT basis functions as components of a fiducial image allows for fast identification using the Discrete Cosine Transform. Fast algorithms exist for the DCT, especially for the 16 by 16 size utilized in the MPEG video compression standard [35]. Computing the DCT performs a simultaneous correlation with all 256 possible basis images.

The 2D DCT-II N by N unnormalized transform is:

$$\text{(4-8)} \quad F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

The DCT result is linearly dependent on the amplitude of the input signal. This amplitude (effectively with a fixed scaling from the correlation coefficient equation) can be directly determined by examining the F(0,0) (DC) term of the DCT result. Dividing all other values by F(0,0) normalizes for intensity. (A threshold is used to reject images less than a minimum intensity).

An interesting characteristic of the DCT-II is its behavior under rotation. Let I be an original image and I' the image obtained by rotating I though 90 degrees counter clockwise. Then, letting $\hat{I} = DCT(I)$ and $\hat{I}' = DCT(I')$:

$$\hat{I}(u,v) = (-1)^v I(v,u) \quad (4\text{-}9)$$

Applied recursively, it can be seen that the DCT of any orientation can be easily derived from the DCT of any other orientation. The orientation is indicated by the presence of the following DCT terms:

- F(1,0) positive: No rotation.
- F(0,1) negative: 90° rotation.
- F(1,0) negative: 180° rotation.
- F(0,1) positive: 270° rotation.

Once the orientation is determined, identification of the fiducial consists of determining the cell with the maximum absolute value (other than 0,0, 1,0 and 0,1). The cell can then be trivially corrected for the image rotation by exchange of terms and/or negating the correlation result.

Note that the DCT is only performed once. The orientation is determined and then the index of the cell with the maximum absolute value in the current orientation can be translated to the index for the cell in the normal orientation.

The major contribution of this chapter is the use of DCT basis functions to produce fiducial images with built-in orientation identification and excellent cross-correlation characteristics and an associated algorithm that allows for rapid identification of the fiducial. This is the first fiducial system to make effective use of the image gradient to convey information. This is a significant contrast with existing systems that utilize binary images, thereby not taking advantage of the range of pixel values other than for antialiasing purposes.

5 A Functioning System

The following is a description of a system created for this thesis by the author (Paul Middlin) and Dr. Charles Owen [36] to use the method derived by examining the important criteria for making a good fiducial. This DCT method was implemented as the identification method for the fiducials, has a square fiducial with a black border, and can calculate pose from the fiducial locations. During development, it has been the goal to produce an accurate and reliable fiducial tracking system. Hence, many steps in the process have been redesigned relative to existing systems or optimized for best processor and evaluation performance. These are the major steps in the program, which will be described in detail:

1. Search for the beginning of a border, using a threshold.
2. Trace the border to get an outline of the quadrilateral.
3. Account for camera distortion in the outline.
4. Locate the 4 corners of the quadrilateral.
5. Test to make sure it is a quadrilateral.
6. Fit lines between these corners to get sub-pixel accuracy on the actual corners.
7. Warp the square to a 16x16 candidate image
8. Do a DCT transform on the candidate image for identification.
9. Use the corners of the fiducial to calculate pose.

5.1 Finding the Border

To find where a fiducial might start, the system uses scan line techniques to search for a point where the pixel values move from a background color to the fiducial edge color. Rather than thresholding the entire image and then scanning each line for a black pixel, the system utilizes a faster approach. The image is not thresholded ahead of time, which will save time and memory by not having to create an intermediate image. The grayscale value is calculated as the pixel is examined. If this value is above a threshold, it is white, otherwise it is considered black.

The value of the threshold is chosen by taking a quick sample of the image (every 5^{th} column in every 5^{th} row). This provides an estimate of the average intensity in the image. This is used as the threshold value.

The system also scans only every 5^{th} line because any fiducial that we consider large enough to be found will have to cross at least one of these scans. Remember that we require the inner part of the fiducial to be 16x16 pixels at least, plus the 15% border width for a total of a 25x25 pixel fiducial. It might in fact be possible to skip even more lines to save time.

The threshold is one of the touchiest parts of the system in varied lighting conditions. It is often possible to choose a good constant threshold for a particular image, but this value might be completely ineffective for the next. In an effort to find a good technique, other possibilities were examined and even implemented for comparison.

One technique tried was to look for a sudden change in pixel intensity while scanning through the image. At the point where the intensity changed, that pixel value was chosen as the base black value. Any pixel, then, within a smaller threshold range

was taken as black as well. For instance, a pixel was found by searching for change of 100 in pixel intensity. That pixel's intensity (say, 50) will be used, plus an additional threshold amount equal to 1/4 of the initial threshold. So, for an initial threshold of 100, and a pixel found with value 50, any pixel with an intensity of 75 or less would be considered black.

This differencing technique tended to be more adept at finding the outlines in varied lighting conditions; however it tended to jitter quite a bit. This is because the first pixel found and used as the base black value could vary greatly in intensity from frame to frame, causing the outline of the fiducial to grow or shrink depending on how black the initial pixel was. The method was reliable in isolation, but did not produce effective results in a system.

Other techniques, such as trying to trace using only differences in pixels were tried, with little success. An image-wide edge detection could be done by using something like a Canny filter. This technique would probably be much slower, and would not necessarily produce connected outline regions the way the thresholding technique does.

5.2 Tracing the Border

The pixels for the border are found by starting from the first pixel identified as a border pixel (from step 1) and following the edge in a counter clockwise fashion. This is a complex process, because the first direction that is tried for the current pixel being tested depends on the direction of the last marked pixel. That is, if the algorithm got to the current pixel from the right, it would try a different direction first than if it had arrived at the pixel from the left.

Table 5-1 shows which direction the algorithm tries first, given the direction that was successful from the last pixel. The first column shows the direction that was tried from the last pixel to find the current one. The second column is the first direction that will be tried from the current pixel:

From	First
↗	↘
↑	↗
↖	↗
←	↖
↙	↖
↓	↙
↘	↙
→	↘

Table 5-1 - Starting Directions for Tracing

The first starting direction will be the ↖ direction (despite the fact that the very first pixel was found coming from the left).

The algorithm will continue moving around the edge of the potential quadrilateral until it reaches the starting point, in which case a loop has been created. The pixels are marked as "visited" as they are added to the list of pixels in the outline. This is not so that they won't be visited again while tracing, but rather so that they won't be retraced later on during step 1 (scanning for starting edges). In fact, it may be necessary to revisit a pixel while outlining to make the loop complete, such as in the following example:

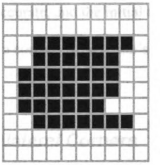


Figure 5-1- Region with extraneous pixels

In this example, there is no where to go from the bottom right pixel except for back to the previous pixel. While this pixel is probably an extraneous error, it must of course be accounted for and included in the outline.

There are a few final optimizations. Small regions are thrown out – if the outline is less than 8 pixels in length it is discarded. Note also that not every pixel inside the fiducial has to be processed. In fact, only the pixels along the border and the pixels immediately surrounding those border pixels are examined. Region growing techniques such as those in ARToolKit would require visiting every pixel in the image twice: once for thresholding and once again for labeling of the region. Additionally, the edge pixels in the region would be visited to do the outlining.

5.3 Accounting for Camera Distortion

The camera can add quite a bit of distortion to the pixels in the image, making lines bend or stretching objects. If the camera being used has been calibrated (see Section 11), then this can be accounted for in the outline pixels before trying to analyze them. This is done at this stage, to help find better corners. It straightens the lines, so

that the line fitting step is more accurate, and the initial corner finding is less likely to find a bowed out edge instead of a corner. Roger Tsai's method was used [37].

5.4 Locating the Quadrilateral Corners

The corners of the square are found by finding two vertices that are far apart, drawing a line between them, and finding the vertices furthest away from this line on either side.

The first step is to estimate where the first vertex is by taking the pixel in the outline with the smallest X value. In the event of a tie, the one with the smallest Y value will be used. This is not the actual first vertex, but is merely a means by which to find the third vertex.

The third vertex is the pixel from the outline that is the furthest from the estimated first vertex using the Euclidean distance (Figure 5-2(a)). Once found, the algorithm repeats the process of finding the furthest outline pixel from the third vertex, which will be the actual first vertex (Figure 5-2(b)).

Next, draw a line from the first vertex to the third. There will be a one pixel on each side of the line that is furthest away from the line, as seen in Figure 5-2(b).

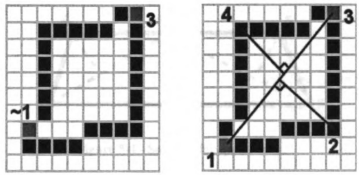


Figure 5-2 - (a) Estimated first, actual third (b) Finding 2 and 4

This covers the majority of the cases; however there is one special case that must be addressed separately. If the fiducial is viewed from a straight perspective angle, a rhombus-like shape is produced as in Figure 5-3. In this case, there is no point found to the right of the line drawn from vertex 1 to vertex 3. Therefore, the point found as vertex 4 is not necessarily a vertex at all. Also, vertex 3 from the previous steps must actually be vertex 2.

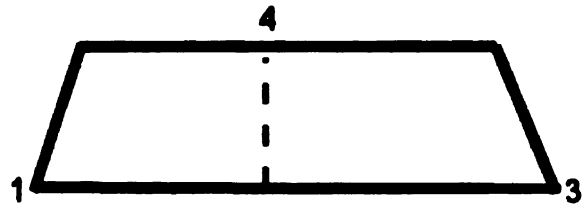


Figure 5-3 - Special Case

This situation can be detected when a point is found more than a certain distance away from the line on one side but not the other. In that event, a new line is drawn from the point found (call it pivot) to the first vertex, and a second line is drawn from the third vertex to the pivot. If a point is found to the right of each of these lines, then these are the real points 3 and 4. If one of the two lines has no points to either side of the line, then the pivot can be identified as being a vertex. See Figure 5-4:

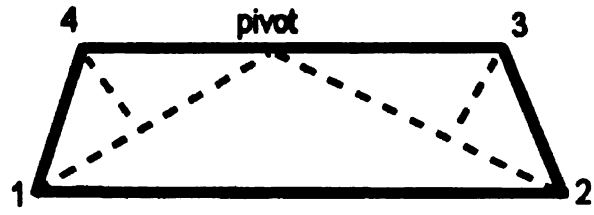


Figure 5-4 - Solution for special case

If it is found that the quadrilateral is less than 25 pixels across in any dimension, it is discarded. Fiducials of smaller size than this cannot be reliably identified because the resolution prevents proper location of the lines. Additionally, the inner part of the fiducial would be less than 16x16, the size of the base images.

5.5 Quadrilateral Test

After finding the four vertices, each of the four edges is tested for straightness. This is done by creating an approximate line edge from the first vertex to the second, then comparing each of the pixels along the edge to the line. If a pixel is above a certain tolerance level of distance, then the line is not considered straight, and this is not a quadrilateral.

5.6 Line Fitting

To find the true corners of the quadrilateral, a line is fitted between each of the vertices using the pixels between each vertex from the outline. This produces four lines that may actually intersect somewhere other than at the center of the supposed vertex (Figure 5-5). This allows for sub-pixel accuracy of the corners of the quadrilateral. This would also fix problems like the extraneous pixels in Figure 5-1. The corner points themselves are actually excluded from the line fitting, because the corners jitter between frames more than the rest of the line.

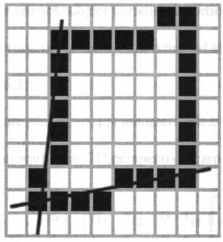


Figure 5-5 - Line fitting and intersection

5.7 Warping

Now that the four points of the quadrilateral have been found, the image inside the quadrilateral must be warped to a 16x16 square so that it can be identified. The points on a fiducial as it appears in a captured image are subject to perspective projection using a camera calibration matrix:

$$(5\text{-}1)\qquad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Here, x, y, and z are coordinates in the fiducial's coordinate system. u and v are coordinates on the screen (coordinates in the image that is to be warped). The matrix T transforms the points from the fiducial coordinate system to the screen coordinates. s is a scaling factor.

This assumes that u,v are not subject to radial distortion. In this application it is assumed that the radial distortion of u,v have been removed at an earlier step, so u,v are undistorted values (Section 5.3).

The fiducial images are planer, so it is assumed that z=0 for all points. Hence, the problem can be reduced to an equivalent 2D perspective warp:

$$(5\text{-}2)\qquad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & 0 & t_{14} \\ t_{21} & t_{22} & 0 & t_{24} \\ t_{31} & t_{32} & 0 & t_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

This can be rewritten as:

$$(5\text{-}3)\qquad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

This equation (5-3) describes the relationship between pixel locations in the fiducial image coordinate system (omitting the z axis) and image coordinates. The resulting pixel values are arbitrarily scaled and must be divided on a pixel-by-pixel basis by the scale factor s to determine the results. Because the matrix P can be arbitrarily scaled, a unique P is determined by setting $p_{33}=1$:

$$(5\text{-}4)\qquad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Expanding this gives:

$$
\begin{aligned}
su &= p_{11}x + p_{12}y + p_{13} \\
sv &= p_{21}x + p_{22}y + p_{23} \\
s &= p_{31}x + p_{32}y + 1
\end{aligned}
\tag{5-5}
$$

Substituting s into su and sv:

$$
\begin{aligned}
(p_{31}x + p_{32}y + 1)u &= p_{11}x + p_{12}y + p_{13} \\
(p_{31}x + p_{32}y + 1)v &= p_{21}x + p_{22}y + p_{23}
\end{aligned}
\tag{5-6}
$$

Multiplying this out and solving for u and v on one side:

$$
\begin{aligned}
u &= p_{11}x + p_{12}y + p_{13} - p_{31}xu - p_{32}yu \\
v &= p_{21}x + p_{22}y + p_{23} - p_{31}xv - p_{32}yv
\end{aligned}
\tag{5-7}
$$

Therefore, there are 8 unknown variables. There are four pairs of known (u,v) and (x,y) coordinates, creating 8 equations. Therefore, there is an exact solution for P.

The first step in the warping process is to compute P given the corners of the fiducial in the fiducial coordinate system and the corresponding points in the image. The image is subject to an arbitrary rotation of some multiple of 90 degrees which will be ignored at this point in the process. That rotation is removed in the later orientation determination set. The points on the fiducial image are determined directly by the size of the image: (0,0), (d,0), (d,d), (0,d), where d is the width (and height) of the fiducial image in world coordinates. However, the actual scaling of d is arbitrary in this step, so we may use 1 for d, simplifying the calculations. The points in the image are determined by the corner finding process described in the previous section.

Using the equations from (5-7) and substituting (x, y, u, v) with the coordinates of each corner point (x_1, y_1, u_1, v_1 through x_4, y_4, u_4, v_4), we form a system of equations represented as a matrix as follows:

$$\textbf{(5-8)} \quad Ax = b$$

where:

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -x_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -x_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_2 & -y_3u_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_2 & -x_3v_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_2 & -y_4u_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_2 & -x_4v_4 \end{bmatrix}, x = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \end{bmatrix}, b = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix}$$

Here, the arbitrarily scaled (x,y) coordinates will be: (0,0), (1,0), (1,1), (0,1). This makes the A matrix simply:

$$\textbf{(5-9)} \quad A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & -u_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -v_2 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & -u_3 & -u_3 \\ 0 & 0 & 0 & 1 & 1 & 1 & -v_3 & -v_3 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -u_4 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -v_4 \end{bmatrix}$$

Solving for x will reveal the values in the P matrix of equation (5-3). Now, the P matrix can be used to warp the image.

Let the 16x16 warped image be W. W(x,y) is a pixel within this image, where x and y are scaled from 0 to 1 (instead of 0 to 15). Let I be the original image from the

camera, and let I(u,v) be a pixel in this image. To fill in the warped image W, use equation (5-3) to get the corresponding (u,v) coordinate in I for each (x,y) in W.

Because the actual interior image is surrounded by a black border, the algorithm must compensate by scaling and adding an appropriate amount to find (x,y).

That is:

```
for each (i,j) where (0<i<15) and (0<j<15)
{
     x = (i+.5)/16 * .7 + .15
     y = (j+.5)/16 * .7 + .15
     u = p11x + p12y + p13 - p31xu - p32yu
     v = p21x + p22y + p23 - p31xv - p32yv
     W(i, j) = I(u, v)
}
```

Figure 5-6 - Pseudo-code for Finding Warped Image

The last statement in Figure 5-6 is pseudo-code. In implementation, simple bilinear interpolation is used to get a sampled value from the pixels surrounding (u,v).

5.8 Identifying with the DCT

The 16x16 image W will now be examined by taking the Discrete Cosine Transform of these values as described in Section 4.2. Recall that both the orientation and identification of the fiducial is done in this one step.

5.9 Calculating Pose

Calculation of pose from the points on the fiducial can be done in many ways. The general problem of calculating pose from four points is called the P4P problem. As

mentioned previously, pose *can* be calculated from only three points, but can have multiple solutions [15]. In fact, it almost always has two solutions and can have up to four [38]. This is called the three-point perspective problem or P3P. One available option in this system is to calculate pose from just three of the points using an iterative method described in [20] and adapted from code originally written by Dr. George Stockman. This method runs in a reasonable amount of time but finds only one of the two possible solutions, and therefore tends to find the wrong solution approximately 50% of the time.

The ARToolKit uses a non-iterative vector based method for calculating pose. This algorithm uses all four points to determine a rotation, then a translation that fits the points. This method is sensitive to noise and tends to produce graphics with a large amount of jitter. An imaginary triangle is created between the camera origin and each of the corners to make 4 planes. The normals to these planes are then calculated. The cross product of plane 1's normal with that of plane 3 determines the X direction. Likewise, the cross product of the second and fourth plane normals determines the Y direction vector.

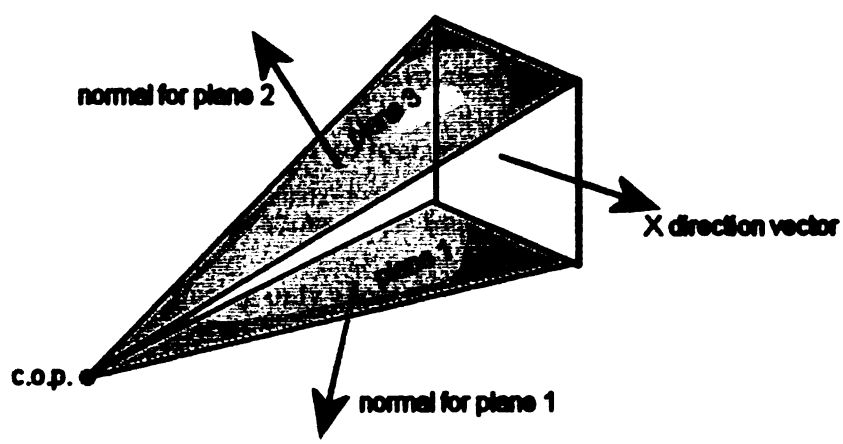


Figure 5-7 - Finding the X axis

Because the positions of the pixels are not perfectly accurate, the X axis and Y axis found may not be perfectly perpendicular. To fix this, the half vector between the two is found and is used to find a true X and Y axis equidistant from the half vector. The Z direction is then simply the cross product of the X and Y vectors.

Once the rotation has been established, a least squared solution for the translation is found. It is important to note that this solution for a transformation is least squared with respect to the translation only. It may have been possible to create a better rotation that would give less error in the least squared sense.

This vector solution tends to be very jittery and noisy. More specifically, the rotation tends to be noisy since it is calculated analytically and not in a least-squared sense before taking translation into account. Small changes in corner locations between frames create large rotational changes. If the only goal was to lay a flat polygonal model onto the fiducial surface, this technique would suffice. If, however, a model with some depth in the Z direction is necessary (such as a simple cube) then this method causes far too much visible "dancing".

Given the problems exhibited by a traditional P3P solution and the ARToolKit P4P solution, an alternative solution was developed for this application based on extension of the inverse Jacobian P3P to four points. This method utilizes all four points to find the most probable 3D locations of the four points in relation to the camera. It tends to converge quickly, using only about 5 iterations to get to an error less than 1E-6 per point.

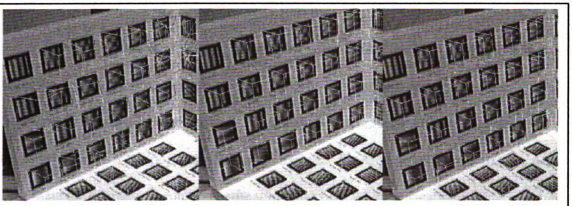
This P4P method finds the 3D location of the fiducial in camera coordinates by assuming that the 3D coordinates lie along four rays that point from the camera's center

of projection to the screen coordinates of the four corners of the fiducial on the view reference plane. So, what this algorithm really does is find distances along these rays where the fiducial corners should be in the camera's coordinates. These distances are a least-squared error fit of the known distances between the fiducial corners and the apparent screen coordinates of the corners.

Knowing the 3D location of these corners is not enough to know the pose of the fiducial. The next step is to calculate a transformation between the fiducial's coordinate system and the camera coordinate system. This computation is done using a Singular Value Decomposition (SVD) method to find a least-squared error transformation (in both the rotational and translational sense). This was adapted from the algorithm presented by K.S. Arun et al [39].

This P4P solution is extremely stable in comparison with the other two solutions. There is some instability when the normal of the fiducial is pointing very close to the camera. The iterative method may be finding locally ideal solutions instead of the best solution, and in such a case will be sensitive to small changes in corner location.

Figure 5-8 shows a comparison of these three methods. The cubes in the P4P vector solution are often times rotated poorly. Note that in the P3P method the fiducials in the bottom and left corners are facing in entirely the wrong direction – these are the extra solutions that come from using only three points. The P4P iterative method yields straight polygons.



P4P Vector P3P Non-linear P4P Non-linear

Figure 5-8 - Pose Finding Method Comparison

6 Evaluation

The system was evaluated in a variety of ways as this research sought to improve the performance of fiducial-based tracking. If fact, it is not this system so much as the methods that it uses that need testing. To give the results more meaning, they are compared to tests of the popular ARToolKit system. Three particular types of tests were done: distance estimation, rotation estimation, and fiducial identification.

6.1 Distance and Rotation Test Setup

A series of pictures were taken using the QuickCam Pro 3000 USB web-cam. The idea was to take pictures at a variety of known distances and angles. This was done by placing the fiducial such that it was in the center of the picture, at regular distance intervals. Additionally, at each distance the fiducial was tipped away at regular angles.

A long, flat board was used to mark off the locations of the fiducial. The fiducial was attached to the front of a rigid box. The box was placed every 1 foot along the board. The box was then rotated about the left edge of the fiducial, such that the left edge remained the same distance away from the camera. The right edge would be moving away from the camera as this angle increases. A picture was taken every 15° from 0° to 75°. Figure 6-1 shows the picture taken when the fiducial was 4 feet away and at an angle of 45°.

The distance to the camera is not technically at 1 feet, 2 feet, etc. This is the distance from the end of the board. The camera is then an additional 1¼" from the edge

of the board. This still, however, is not the true distance, because the center of projection (COP) is not at the front edge of the camera. For this camera, the COP is about 7/8" from the front edge of the camera as determined by camera calibration.

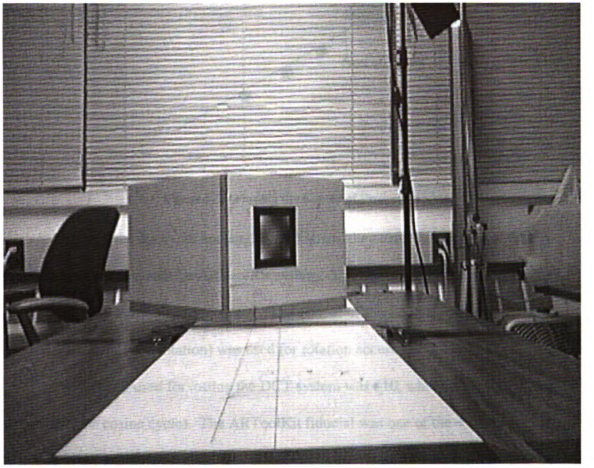


Figure 6-1 - Test Setup

This measurement was determined by using the edge of the field of view as a guide. To find the COP, objects (soup cans) were placed in a line just inside the field of view of the camera. Drawing a line along these objects on both sides of the camera reveals the COP where the two lines intersect. See Figure 6-2 for a diagram.

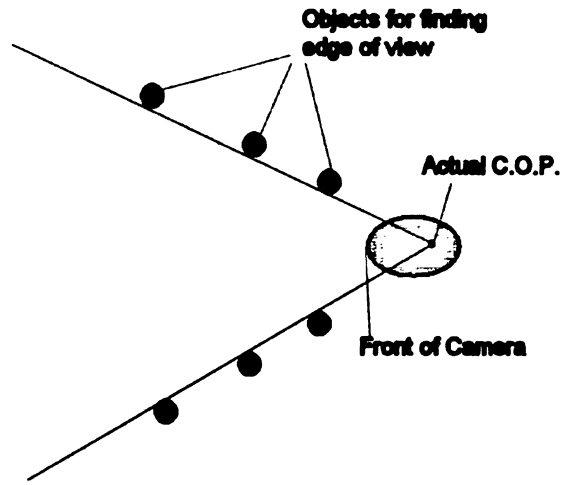


Figure 6-2 - Finding the Center of Projection

Now that the distances and angles are known, they can be compared to results from the DCT system and the ARToolKit system. The bottom-left corner of the fiducial was used to calculate distance accuracy. The normal of the fiducial (compared with the correct normal for each rotation) was used for rotation accuracy.

The fiducial used for testing the DCT system was #10, which is DCT #2 in both directions (one cosine cycle). The ARToolKit fiducial was one of the 4 fiducials included in the example program they distribute. Both fiducials were 3.5" in both dimensions. Both systems were set up to expect fiducials of this size for accurate pose calculation.

Figure 6-3 - **DCT Test Fiducial**

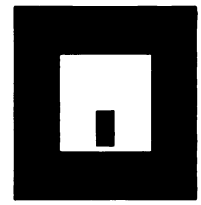


Figure 6-4 - **ARToolKit Test Fiducial**

6.2 Distance Results

When running the tests, fiducials could not always be found in the picture. For instance, when the fiducial is far away and is at a sharp angle, the fiducial will not be 25 pixels across, and will be immediately rejected by the DCT system without further testing. The ARToolKit will make an attempt to identify the fiducial at any size, but will often do poorly in such situations, as would be expected. The following graph shows the error for both systems as a percentage in relation to the correct distance. That is, the percentage error is:

$$(6\text{-}1) \quad error = ABS\left(\frac{MeasuredDist - ActualDist}{ActualDist} \times 100\%\right)$$

Gaps in the DCT line are where there was no fiducial found. The numbering on the bottom indicates the picture number that was being analyzed. Remember that 6 different angles were taken at each distance, so pictures 1 through 6 would be the 6 angles at the first distance. Also remember that the lower-left corner of the fiducial will not change locations as the fiducial rotates.

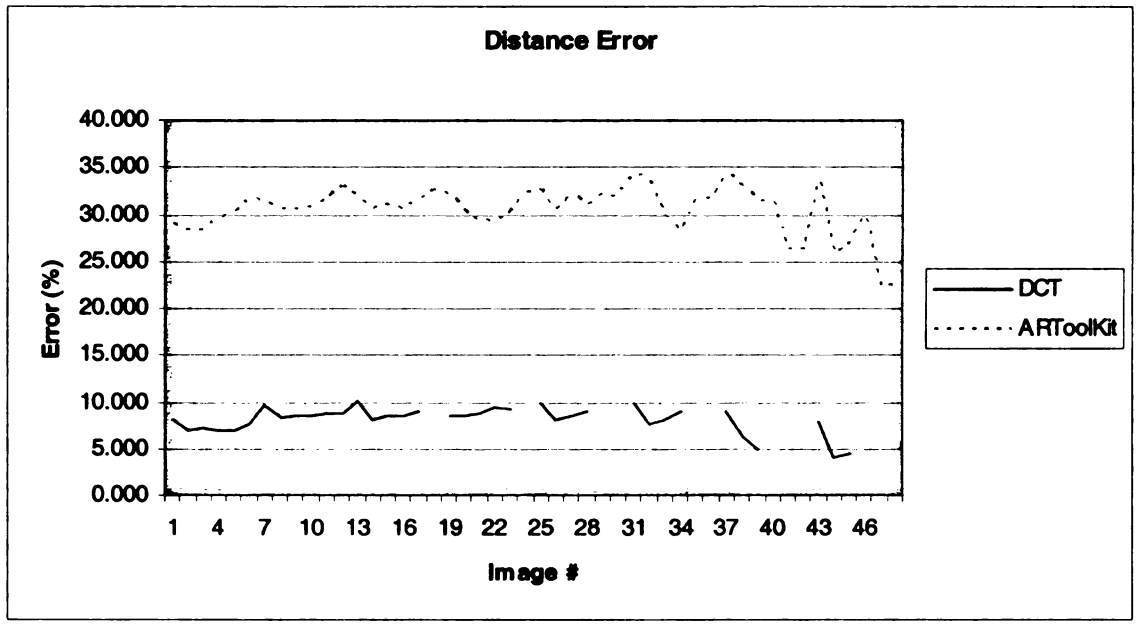


Figure 6-5 - Distance Error Comparison

The percentages shown could technically be shown as negative, because for both systems the error was always on the short side. That is, both systems always underestimated the distance to the lower left corner of the fiducial. Clearly, both system exhibit a measurement bias that could be factored out in future work.

6.3 Rotation results

The same pictures were used for testing rotation as for distance. To test rotation, the vector (0,0,1) was used as the normal to the fiducial in "world" coordinates. World coordinates in this case are the coordinates according to the model, which is the fiducial. The normal vector was multiplied by the transformation matrix generated from the pose estimation done per fiducial. This results in a vector that is normal to the fiducial in the camera's coordinate system. At a 0° rotation, this normal should be pointing directly

back at the camera. The normal would then be rotated 15° about the Y axis to get the next normal.

The calculated normal of the fiducial was compared to the correct normal by taking the dot product of the two normals, then taking the arcos() of this value to get the number of degrees of error.

There are multiple graphs that follow to illustrate the angular errors. The first graph, Figure 6-6, shows the results for all 48 pictures. Remember that the pictures were taken in order by distance, so the angles kept changing from picture to picture (0, 15, 30, 45, 60, 75, 0, 15, 30, ...). To help make the comparison more clear, a graph is shown for the 8 distances at a constant angle, for the 6 different angles in Figure 6-7 through Figure 6-12.

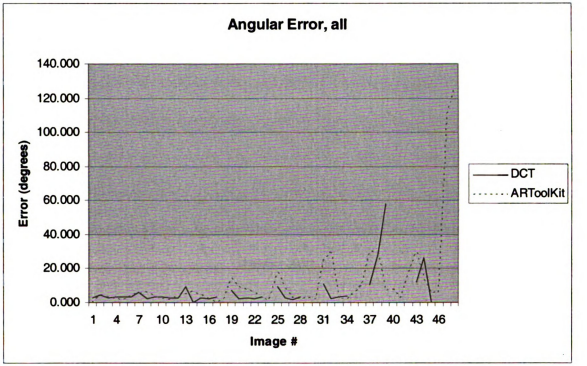


Figure 6-6 - Angular Error, All Images

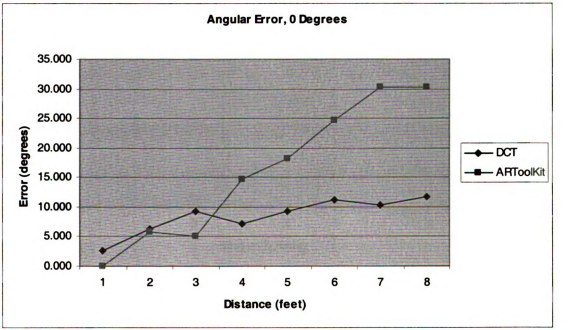


Figure 6-7 - Angular Error, 0 Degrees

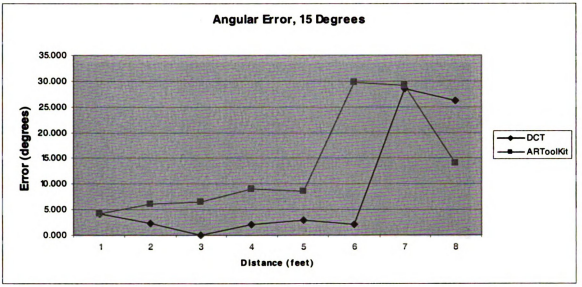


Figure 6-8 - Angular Error, 15 Degrees

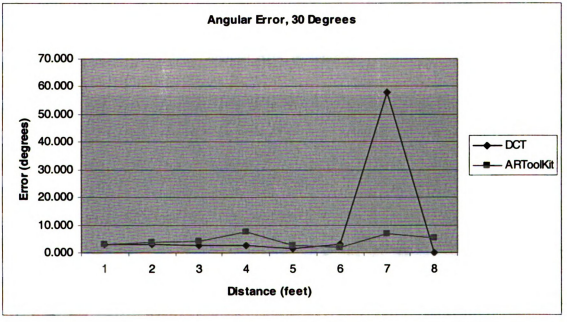


Figure 6-9 - Angular Error, 30 Degrees

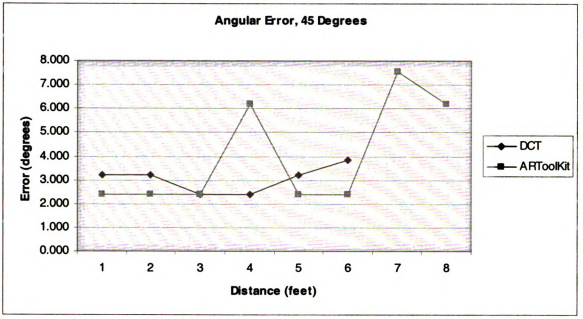


Figure 6-10 - Angular Error, 45 Degrees

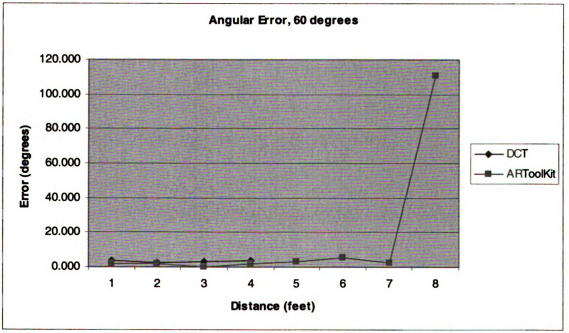


Figure 6-11 - Angular Error, 60 Degrees

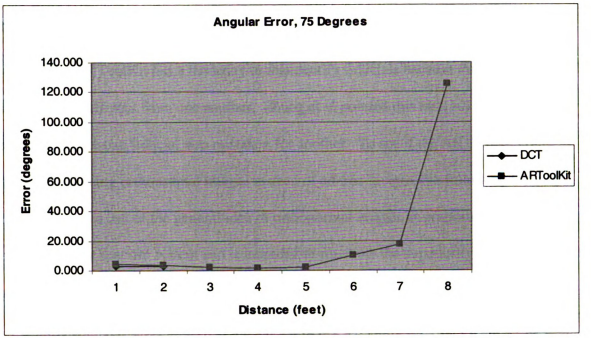


Figure 6-12 - Angular Error, 75 Degrees

6.4 Identification Results

The ARToolKit has a tendency to misidentify fiducials because they are not significantly different from one another. Zhang et al pointed this out [10] and showed an example, where one fiducial was mistaken for another. To see if the ARToolKit could benefit from using orthonormal images instead of ad-hoc images, the following test was set up. This is a test of the general idea of orthonormal images as proposed in this thesis, rather than of a specific system and is intended to illustrate the superiority of this design over ad hoc fiducial images even in an existing system.

Six fiducials were chosen and loaded into the ARToolKit. These were ad-hoc images. The images chosen had the numbers 2,3, and 8 inscribed in the squares. The letters C, G, and B were also chosen. A drawing shape was associated with each fiducial (cone, cube, sphere, and torus). The associations were as follows:

Fiducial	Associated Shape
2	Cube
3	Cone
8	Torus
B	Sphere
G	Cone
C	Cube

Table 6-1 - ARToolKit Fiducial Shape Associations

A few pictures were taken from about 3 feet. Figure 6-13, Figure 6-14, and Figure 6-15 show the misidentification errors common with the ARToolKit. In the first picture, it is apparent that the system has confused fiducials "2" and "3". In the second picture the

"8" has a specularity from the light, causing it not to be found as a square. With "8" out of the way, "3" is now identified as "8". "2" is still identified as "3".

Figure 6-13, Figure 6-14 were taken at a distance of about 3 feet. Figure 6-15 was taken at about 6 feet from the fiducials.

The ARToolKit was then tested using a different set of fiducials. Fiducials generated with DCT patterns like those in the proposed system were used. The border width was kept the same, as this is what the ARToolKit expects. The idea was to see if the ARToolKit would fare better with orthonormal images instead of the ad-hoc images. Technically, the DCT images are not completely orthonormal because they still have the orientation component which will be common amongst all of the images.

Figure 6-16 shows a picture taken from 3 feet away. All 8 of the fiducials are correctly identified, despite the fact that many of the DCT images are "close together". DCT images 10, 11, 12, 13, 14, 15, 100, and 105 were used. The lower-left portion of each picture shows the original image, without overlays. The lower-right portion shows the binarized image with identified fiducials shown in red outlines.

In the second picture, which was taken at 6 feet (Figure 6-17), there are 2 fiducials that were not identified (#11 and #14). The rest of the fiducials, however, are identified correctly. While it is not good that 2 fiducials were not identified, it is good that they were not mistaken for the wrong fiducials.

Table 6-2 summarizes the identification problems of these systems for each of the pictures taken. The misidentified fiducials are highlighted.

ARToolKit				DCT System		
Fiducial	**Figure 6-13**	**Figure 6-14**	**Figure 6-15**	**Fiducial**	**Figure 6-16**	**Figure 6-17**
B	B	B	B	**10**	10	10
C	C	C	-	**11**	11	-
G	G	G	C	**12**	12	12
2	3	3	-	**13**	13	13
3	2	8	3	**14**	14	-
8	8	-	8	**15**	15	15
				100	100	100
				105	105	105

Table 6-2 - Identification test results

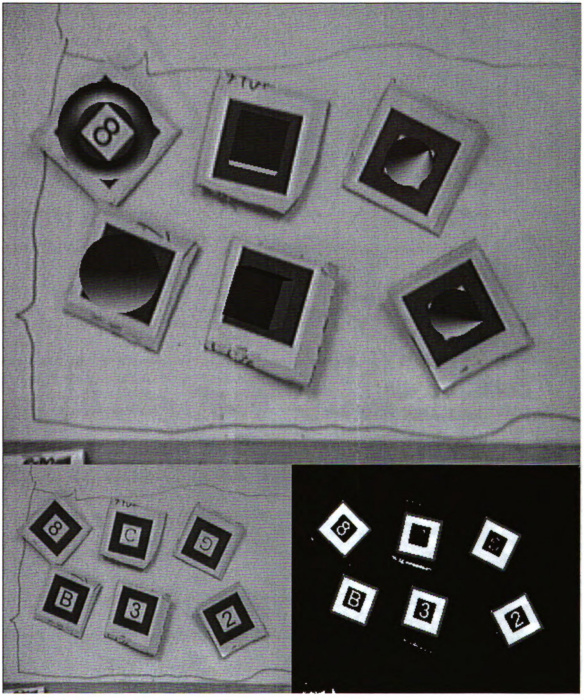


Figure 6-13 - ARToolKit Misidentification, 3 foot, #1

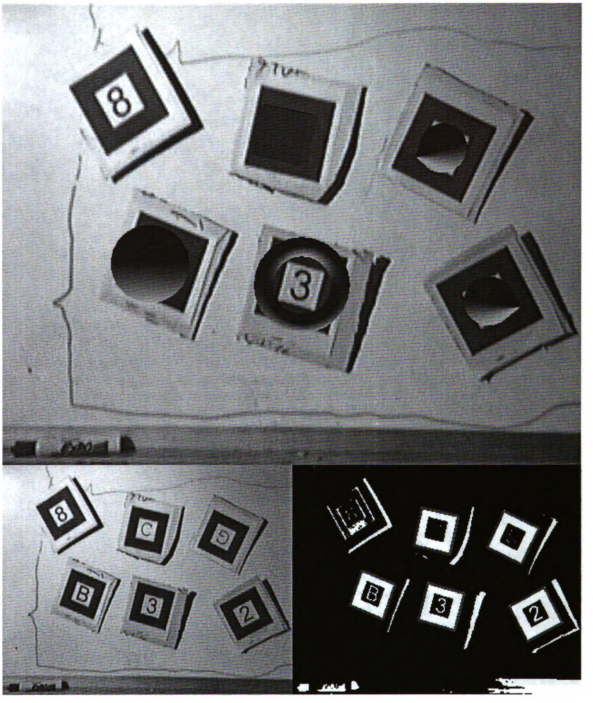


Figure 6-14 - ARToolKit Misidentification, 3 foot, #2

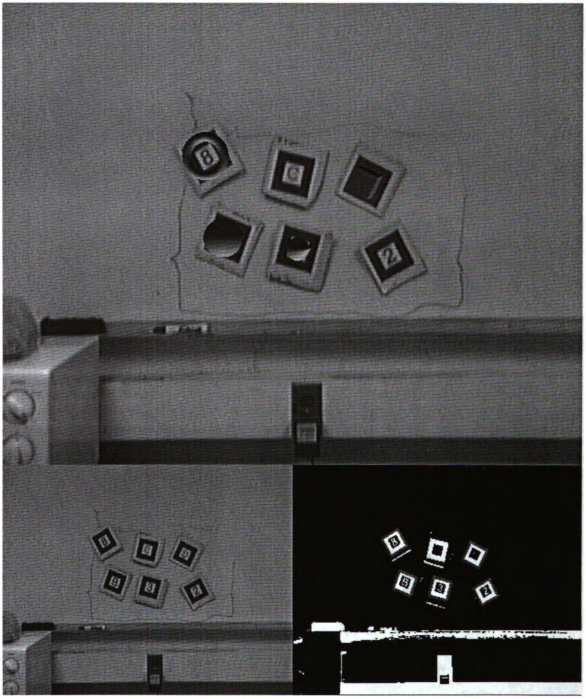


Figure 6-15 - ARToolKit Misidentification, 6 foot

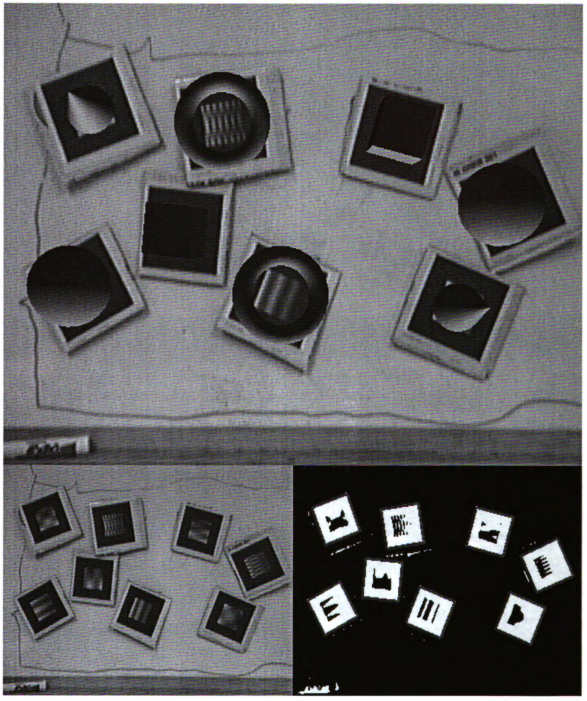


Figure 6-16 - ARToolKit Correct Identification Using DCT, 3 feet

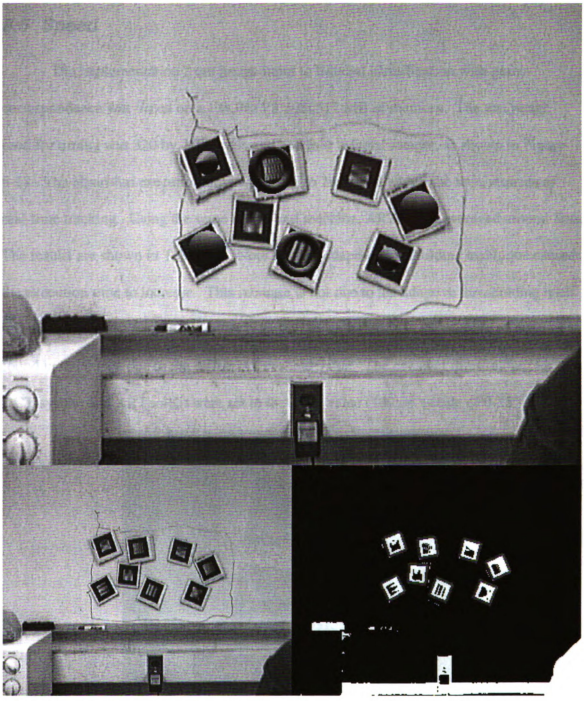


Figure 6-17 - ARToolKit Correct Identification Using DCT, 6 feet

6.5 Speed

The implementation from image input to fiducial identification with point correspondence was timed on a 1.0GHz P3 with 512 MB of memory. The test image used for timing was 320 by 240 pixels and had four fiducial images, as shown in Figure 6-18. The algorithm execution time was close to 2ms, well within the requirements of real-time tracking. Using the same image and machine, ARToolKit averaged around 8ms. The results are shown in Table 6-3. Note that the adaptive thresholding technique caused the execution time to increase. This increase is not due to the adaptive thresholding itself taking significantly longer, but rather is caused by an increase in the number of outlines found in the image due to the different threshold value. The constant thresholding for the DCT-system and ARToolKit were set to the same value (100 on a scale of 0-255).

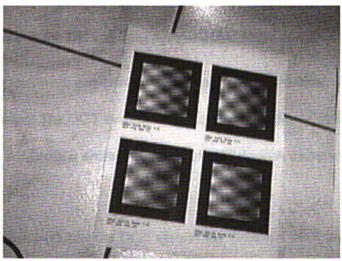


Figure 6-18 - Example Test Image

Method	Average Time (ms)
ATK	8.142
DCT (constant threshold)	1.811
DCT (adaptive threshold)	2.368

Table 6-3 - Speed Test Results

6.6 Discussion of Results

It is important to note that the ARToolKit uses inter-frame processing to smooth out errors or lost fiducials. When running these tests, the ARToolKit would use old results for a frame if it could not find a fiducial, which would of course cause large errors. When examining the graphs above, it would probably be fair to ignore ARToolKit results where the DCT system did not find a fiducial. In these pictures, the size of the fiducial should be considered too small for accurate processing. The ARToolKit could easily ignore these fiducials as the DCT system does to avoid some of the larger errors.

The translations were surprisingly inaccurate in both systems. Distances were always underestimated. The DCT system ranged from about 5% to 10%, which is still a sizable error. The ARToolKit seemed to do much worse, with an average error around 30%.

One of the instances that the DCT system has trouble with is when the fiducial is pointed directly at the camera. While the translation is still relatively accurate, the rotation can be very erroneous. The ARToolKit had a similar problem (see Figure 6-7). The ARToolKit was more accurate in this case at close distances, but worse at large distances. Both systems, in fact, got worse as distances increased.

The identification results seemed to be better when using more appropriate images. The numbers "2" and "3" are not very similar, yet the ARToolKit seems to confuse these two fiducials. Even the high-frequency fiducials seemed to do well, at least at close ranges. It should be noted that the DCT system discussed in this thesis does not suffer from this misidentification problem. While it is of course possible for this to happen, it has been the experience of those who worked with the system that it has never once incorrectly identified a fiducial. Part of this stems from the fact that the DCT system never tries to identify fiducials that are too small, but the orthonormal property of the images is clearly the main reason for the prevention of misidentification.

7 Future Work

The system created was done so with the ideal fiducial criteria in mind. This does not mean, however, that all of the courses of action taken are the best way to meet those criteria. Further, these criteria might be met in many ways, or the criteria might change depending on the needs of a specific application.

7.1 Basis Set

The DCT was an obvious first choice for an orthonormal basis set, but it is not the only possibility. The higher frequency components are more sensitive to errors in the outline detection process and image warping. Cameras have decreased high frequency content as shown in Appendix A. In addition, image blurring impacts high frequency content more than low frequency content.

In preliminary experiments, several custom basis image sets have been constructed that may exhibit better high frequency and frequency spreading characteristics. The trade-off is the lack of a fast transform for identification. However, as the transform is separable, it may be possible to construct a custom transform that runs fast enough for this application. Other known transforms, such as Fourier, Hadamard, or Haar might have better suited basis vectors.

There is also ample opportunity to work in more advanced combinations of basis vectors. Color could be used, for instance, to expand this set dramatically. A blue image could be combined with a red image, immediately squaring the number of combinations possible.

It has been assumed that an orthogonal basis set is the best choice for this application. However, the set can be supplemented using the negatives of the basis functions. Negative correlation is, in fact, as good as zero correlation in an identification system. The only correlation in the augmented set is between basis functions and their negatives. This negative correlation is easily identified in the DCT result. Adding the negative basis functions would not in any way decrease performance and effectively doubles the set size for free.

The proportion of the orientation part of the image to the coded part of the image might also be changed. Using 1/4 orientation, 1/4 DC offset, and 1/2 DCT code may not be the ideal combination. It may be possible to decrease the contribution of the orientation in order to get a better dynamic range with the coded part, and therefore a better resolution. This might allow us to use more of the high-frequency fiducials, or do a better job of finding fiducials in noisy images.

7.2 Pose Estimation

The P4P non-iterative method is clearly better than the others implemented for most situations, but is not nearly perfect. The noisy results when the fiducial normal is pointing close to the camera could be extremely detrimental in applications where this happens often. It may be possible to improve this method by finding better starting values for the iterations. This might be done by using the linear P3P technique, which would be estimation but could start the P4P technique closer to the right results. This might prevent the P4P process from finding locally ideal solutions and move more quickly to the overall ideal solution [40].

The P3P method by Stockman [20] actually only provides one of the solutions, instead of finding all of the possible solutions. Huttenlocher and Ullman [41] devised a method that will produce two solutions for computing the pose of a rigid configuration of three points from a single weak perspective projection. It may be possible to choose between these two solutions to get the right one, particularly if inter-frame information or other such factors are used.

The pose could also benefit greatly from using multiple fiducials together [42]. Using all of the points for all of the fiducials at once would allow for a much less noisy least squared fitting. Each individual fiducial may not line up as well with the corresponding virtual parts, but there would be little noise and more overall accuracy.

7.3 Finding Potential Fiducials

The thresholding technique (and consequently, the edge finding) could be improved to work under a wider range of lighting conditions. There are a variety of techniques in existence, such as the modified homomorphic image processing method used by InterSense Inc. in their circular fiducial system [16]. Most of these more effective methods are, however, much more expensive operations. The method used in this system is fast, since that was one of the goals set forth initially. Nevertheless, as PCs get faster and custom hardware is created, more advanced algorithms become possible.

8 Conclusions

This thesis has set forth a set of criteria by asking questions about what makes up a good fiducial. Though the answers to these questions and the degree to which a fiducial tracking system meets these criteria can differ, the answers to these questions are as general as possible. This set of criteria sets forth a standard by which to judge the quality of a fiducial tracking system, but more importantly creates guidelines which can be used create a good system.

Given these criteria, a new type of fiducial image was created using the Discrete Cosine Transform basis images which has proven to be an effective choice. Additionally, a system using this type of interior image was created with fast algorithms taking advantage of the choices made when examining the criteria for a good fiducial. The speed increase is apparent in all areas: locating the fiducial, identifying the fiducial, and calculating the pose of the fiducial.

Further, to prove the validity of such criteria and the decisions made using those criteria, the system was tested and compared to a popular and effective system. The results have shown significant performance increases in speed, identification accuracy, and pose estimation.

Examining the real needs and purposes behind fiducial tracking has shed a great deal of light on what systems work and why. More specifically, the individual parts of the many systems can be evaluated for the effectiveness of their purpose. The focus of this research is obviously on the science behind tracking technologies, not a specific

implementation. The system discussed here was created to demonstrate the effectiveness of the choices made when examining the identified criteria.

This system helps to validate how fast and reliable the fiducial tracking can be, yet leaves the door open for improvements in many areas. If other technologies and systems are reevaluated with the goals in mind they might be improved. It is the author's hope that this more analytical approach will lead to advancements in the theory behind fiducial tracking rather than mild improvements in specific implementations.

9 Appendix A – Camera Frequency Response

Camera imaging systems are not perfect. High frequency scenes cannot be captured perfectly in any camera, and some cameras are worse than others. To better understand what types of fiducials can be identified, the camera that is capturing the images must be examined. This section focuses on testing the response of cameras to images of varying pixel intensity.

9.1 Background

Cameras use a CCD to convert the visual signal into an electronic pattern. The CCD is an array of receptors (640x480 in many cases) that each capture some part of the scene that the camera is taking a picture of. Unfortunately, these receptors are somewhat interdependent. If a black piece of paper with a single white dot was held in front of a camera, the dot would not register on only one receptor. Even if perfectly aligned, there would be one sensor that receives most of the stimulation while the sensors surrounding it have some smaller stimulation as well. The amount of spillover from pixel to pixel is referred to as the point spread function.

Cameras with a wide point spread function will have trouble viewing high frequency images. That is, images with pixels that change drastically in intensity are difficult to identify. An image that goes back and forth between black and white tends to be blurred and grayed out, as in Figure 9-1. An image that should be alternating pixels of pure black then white becomes shades of gray.

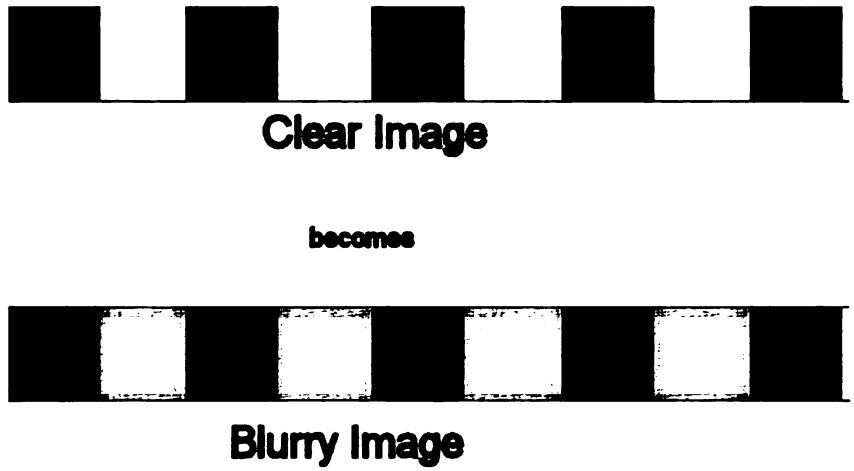


Figure 9-1 - Effect of point spread function

9.2 Test Setup

The purpose of this test is to see how well cameras will respond to changing frequency images. To quantify the frequency, images were printed in bands ½" wide as a cosine function. The cosine function of course was scaled to range from black to white instead of -1 to 1. Figure 9-2 is an example image, which was printed with fifteen ½" bands on an 8½" by 11" paper with a laser printer. The images printed were very high resolution (6000x4500 pixels). The first band has 0 cycles, the second has 1, the third has 2, and so on.

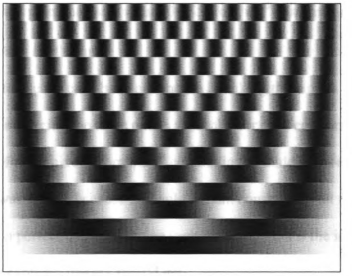


Figure 9-2 - Testing pattern

A total of 22 images were printed in order to print out 320 different bands. The 320th band has 320 complete cosine cycles, which means that on a 640x480 image the values would be exactly 255,0,255,0,... through the whole band. This is the Nyquist frequency for the 640 width image; doing bands with more than 320 cycles would be pointless.

Each printed image was photographed with a camera, and the camera was moved (or zoomed) so that the edges of the image were right at the edges of the camera's picture. Care was taken to get a straight alignment. These images were then processed for frequency response.

The frequency response was measured using the Root-Mean-Square (RMS) method. RMS works by taking the sum of the squared difference of each pixel from the average, divided by the number of pixels squared:

$$\text{(9-1)} \quad RMS = \sum_{i=1}^{N} \frac{(p_i - \bar{p})^2}{N^2}$$

where p_i is the pixel intensity value (0-255) for the i^{th} pixel in a row, and $\bar{p}$ is the average pixel intensity for the row.

The pixels to use for each band are chosen by taking three rows from the middle of each band, then averaging the three pixels vertically for each column. That is, p_i is really an average of three pixels from the i^{th} column in the middle three rows of that band.

The RMS was chosen because it should remain constant for all frequency bands. The RMS value of a perfect 640x480 image is about 12.6. This is the ideal value that each of the camera captured images should get.

9.3 Results

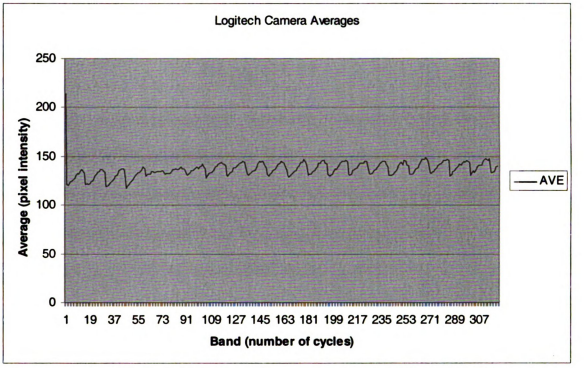
Four cameras were tested. The first is a Logitech QuickCam Pro 3000, which is a medium priced web-cam and is also the camera used during development of this system. This camera has very little distortion, and is capable of providing 30 non-interlaced frames per second. To do this, however, the frames are compressed before being sent to the PC from the camera.

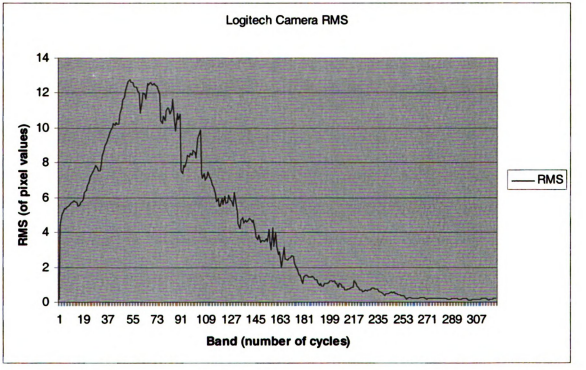
The second camera was a Sony DCR-VX1000 Digital Video camera. This camera is normally used for video capture, but is non-interlaced as is the standard for DV cameras. Also standard is the 720x480 resolution, so the top of these pictures was slightly cropped off. This is a somewhat expensive video camera.

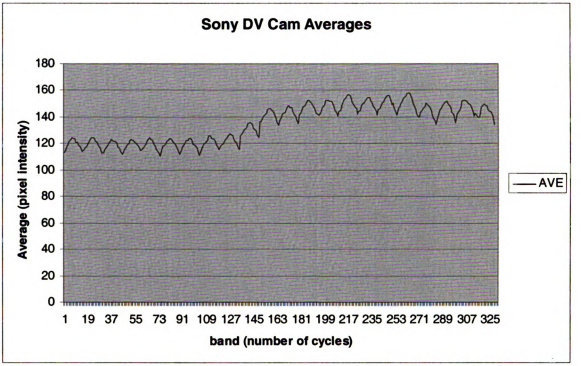
The third camera was a Sony DXC-LS1 lipstick camera. This is also a more expensive camera. While the picture quality is high, it is interlaced and has a great deal of radial distortion in comparison with the other cameras.

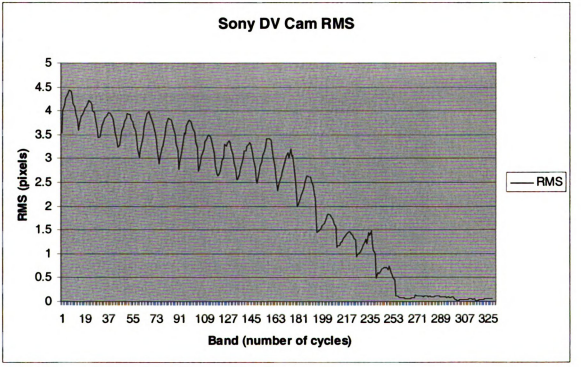
The fourth camera was an older IBM web-cam, purchased for about $20 over 2 years before these tests. This camera is very low quality, and can only provide up to 320x240 images.

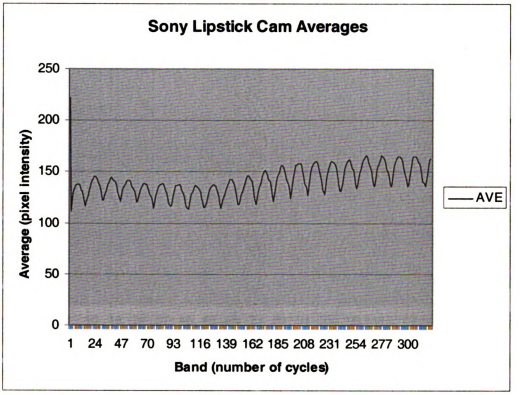
The following are graphs of the average pixel intensities and RMS values of the images taken by each of the four cameras. Remember that the ideal RMS value is 12.6, and the idea average pixel intensity should be about 128 out of 255.

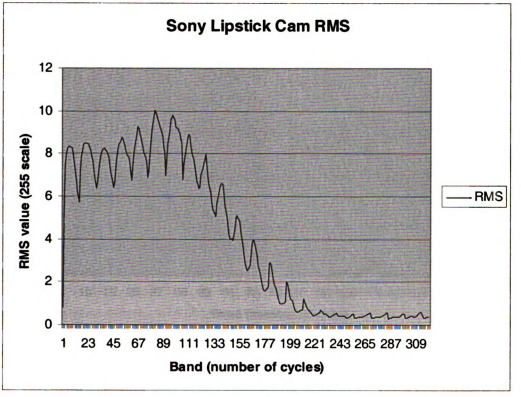

Logitech Camera Averages
250
200
150
100
50
0
Average (pixel intensity)
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307
Band (number of cycles)
AVE

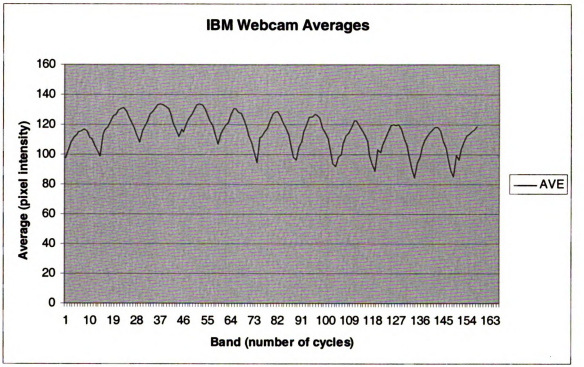

Logitech Camera RMS
14
12
10
8
6
4
2
0
RMS (of pixel values)
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307
Band (number of cycles)
RMS


Sony DV Cam Averages
Average (pixel intensity)
180
160
140
120
100
80
60
40
20
0
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307 325
band (number of cycles)
AVE


Sony DV Cam RMS
RMS (pixels)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307 325
Band (number of cycles)
RMS


Sony Lipstick Cam Averages
250
200
150
100
50
0
Average (pixel intensity)
1 24 47 70 93 116 139 162 185 208 231 254 277 300
Band (number of cycles)
AVE


Sony Lipstick Cam RMS
12
10
8
6
4
2
0
RMS value (255 scale)
1 23 45 67 89 111 133 155 177 199 221 243 265 287 309
Band (number of cycles)
RMS


IBM Webcam Averages
Average (pixel intensity)
160
140
120
100
80
60
40
20
0
1 10 19 28 37 46 55 64 73 82 91 100 109 118 127 136 145 154 163
Band (number of cycles)
AVE

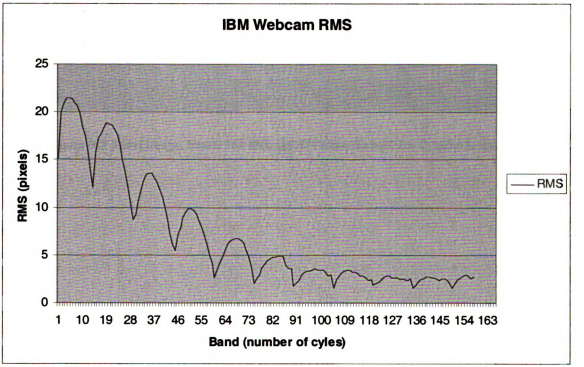

IBM Webcam RMS
RMS (pixels)
25
20
15
10
5
0
1 10 19 28 37 46 55 64 73 82 91 100 109 118 127 136 145 154 163
Band (number of cyles)
RMS

Image comparison:

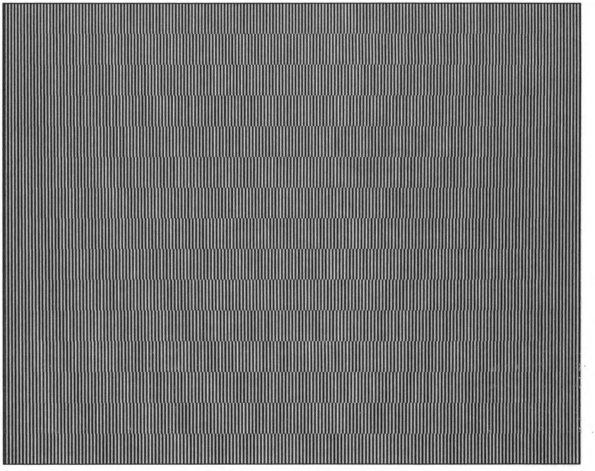


Figure 9-3 -Ideal image, bands 165 through 179 (close to half the Nyquist frequency)

Logitech image:

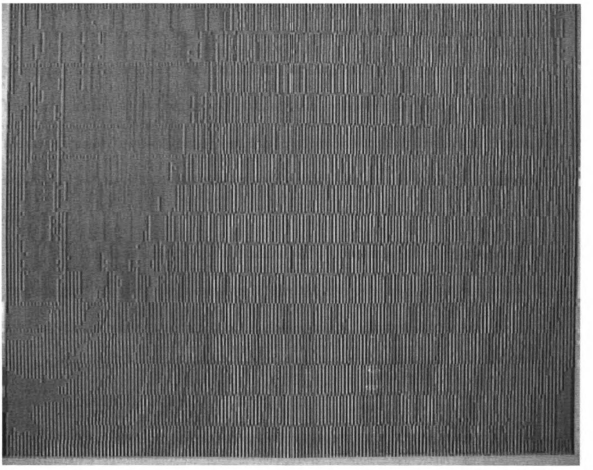


Figure 9-4 - Logitech Image

9.4 Discussion

The most obvious trend is that for each camera, the RMS value decreases as the number of bands increases. This confirms that cameras do not react equally to high frequency noise. As the frequency increases, the spillover from pixel to pixel (result of the point spread function) has a more drastic affect on the pixel intensities. Figure 9-3 and Figure 9-4 show a comparison between an ideal image and that of the Logitech camera.

The Logitech images have the additional disadvantage of being compressed into jpeg images. All of the other cameras offered the ability to take an uncompressed frame. Jpeg was actually designed to remove high-frequency information in favor of compression. In most photographic applications this is not noticeable, but in this case it made a significant difference.

Another interesting trend is that both the averages and RMS values tend to have "bumps". These bumps occur every 15 bands – which is how many bands are on each page. This implies that both the average image intensity and the frequency response are higher across the center of the image than they are towards the edges. So, the camera is more capable towards the center of the image.

The cameras tend to have a high average pixel intensity – the original image would have an average of 128, whereas the pictures taken by the cameras would often be above 140. This varied throughout the picture, despite the ample and constant lighting when the images were taken. Most of these cameras provide some type of automatic gain control, exposure setting, or white balance. The automatic adjustment of these features

was turned off when possible, though it seemed that often times there was still some processing out of the user's control.

These results make it clear that high-frequency fiducials, while orthogonal during creation, will not remain orthogonal when processed by the camera. The cameras' built in smoothing from their point-spread functions will smooth out the image as a sort of low-pass filter. This makes high-frequency fiducials a bad choice as they will become correlated from the camera noise. This reduces the set of usable DCT basis vectors.

10 Appendix B – Testing Data.

Real Data			DCT-based System Estimates			ARToolKit Estimates		
IMG	Angle	RealDist	Dist	Angular Err	%DistErr	Dist	Angular Err	%DistErr
1	0	14.125	12.990	2.563	8.035	10.040	0.000	28.920
2	15	14.125	13.140	4.283	6.973	10.130	4.237	28.283
3	30	14.125	13.100	2.929	7.257	10.110	3.144	28.425
4	45	14.125	13.130	3.206	7.044	9.980	2.374	29.345
5	60	14.125	13.140	3.413	6.973	9.850	1.616	30.265
6	75	14.125	13.040	3.055	7.681	9.660	4.291	31.611
7	0	26.125	23.620	6.280	9.589	17.910	5.732	31.445
8	15	26.125	23.960	2.403	8.287	18.130	6.141	30.603
9	30	26.125	23.920	3.076	8.440	18.150	3.661	30.526
10	45	26.125	23.900	3.206	8.517	18.080	2.374	30.794
11	60	26.125	23.840	2.735	8.746	17.890	1.596	31.522
12	75	26.125	23.830	3.000	8.785	17.470	3.755	33.129
13	0	38.125	34.290	9.249	10.059	25.970	5.126	31.882
14	15	38.125	35.000	0.000	8.197	26.440	6.483	30.649
15	30	38.125	34.870	2.775	8.538	26.300	4.107	31.016
16	45	38.125	34.830	2.374	8.643	26.470	2.374	30.570
17	60	38.125	34.680	3.281	9.036	26.150	0.000	31.410
18	75	38.125				25.720	2.435	32.538
19	0	50.125	45.870	7.252	8.489	33.990	14.760	32.190
20	15	50.125	45.800	2.100	8.628	34.870	9.107	30.434
21	30	50.125	45.700	2.817	8.828	35.370	7.487	29.436
22	45	50.125	45.430	2.374	9.367	35.470	6.178	29.237
23	60	50.125	45.490	3.413	9.247	34.950	2.060	30.274
24	75	50.125				33.850	1.693	32.469
25	0	62.125	56.040	9.249	9.795	41.910	18.195	32.539
26	15	62.125	57.090	2.859	8.105	43.170	8.559	30.511
27	30	62.125	56.850	1.576	8.491	42.220	2.668	32.040
28	45	62.125	56.590	3.206	8.909	42.860	2.374	31.010
29	60	62.125				42.140	2.892	32.169
30	75	62.125				42.280	2.435	31.944
31	0	74.125	66.750	11.187	9.949	48.560	24.632	34.489
32	15	74.125	68.530	2.108	7.548	48.980	29.724	33.922
33	30	74.125	68.130	3.216	8.088	51.990	1.959	29.862
34	45	74.125	67.510	3.863	8.924	53.120	2.374	28.337
35	60	74.125				50.710	5.282	31.589
36	75	74.125				50.710	10.295	31.589
37	0	86.125	78.390	10.263	8.981	56.520	30.231	34.374
38	15	86.125	80.690	28.498	6.311	57.760	29.118	32.935
39	30	86.125	81.930	57.855	4.871	59.060	6.997	31.425
40	45	86.125				59.040	7.536	31.448
41	60	86.125				63.570	2.485	26.189
42	75	86.125				63.570	17.537	26.189

43	0	98.125	90.390	11.763	7.883	64.960	30.345	33.799
44	15	98.125	94.130	26.101	4.071	72.760	14.038	25.850
45	30	98.125	93.700	0.000	4.510	71.770	5.299	26.859
46	45	98.125				68.550	6.178	30.140
47	60	98.125				76.050	110.553	22.497
48	75	98.125				76.050	125.545	22.497
AVERAGE:				**6.876**	**8.105**		**12.458**	**30.441**
STANDARD DEV:				**10.666**	**1.401**		**23.801**	**2.611**

11 Appendix C - Camera Calibration

In order to get accurate pose calculations, it is helpful to have a calibrated camera model. Using the assumed pin-hole camera model is not accurate with many cameras, and will result in creating inaccurate pose estimations. Since camera calibration need only be done once for a camera, a complicated computation is not daunting.

This system uses Roger Tsai's method [37] for camera calibration, adapted from a book by Shapiro and Stockman [20] (with some corrections). This method calculates both the extrinsic camera parameters (translation, rotation) and the intrinsic parameters. Intrinsic parameters include finding the principal point, scale factors, aspect distortion factor, focal length, and lens distortion factor for the camera.

The calibration is done using a jig (see Figure 11-1) onto which was printed 72 fiducials of known location. With four key points (corners) per fiducial, this allowed up to 288 points from which to do the calibration. The photograph seen in Figure 11-1 was taken with the Sony DXC-LS1 lipstick camera. This camera has a significant radial distortion factor. Figure 11-2 shows a blown up version of the upper-right portion of the picture, with two straight red lines drawn along the edges of two rows of fiducials. It is clear that the image bows out, away from the straight lines.

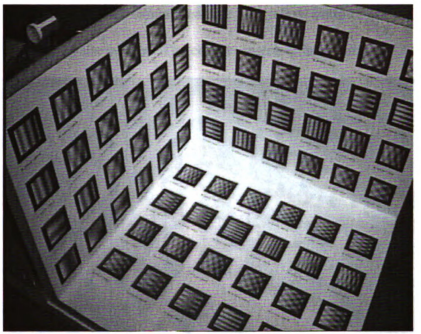


Figure 11-1 - Jig used for calibration

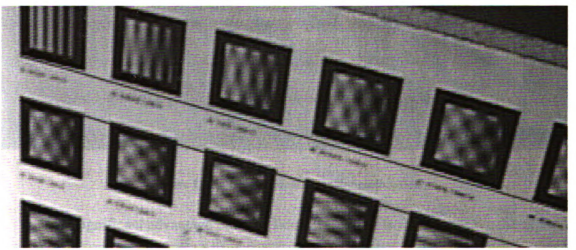


Figure 11-2 - Radial Distortion

After using Tsai's calibration technique, 3D translation is much more accurate. Figure 11-3 is the same jig as above, with the corners marked with a red +. These locations came from translating the known 3D location of the corners into 2D screen

coordinates, using the calibrated camera matrix. The yellow outlines are the outlines of the fiducials as found by the system. The corners match quite accurately.

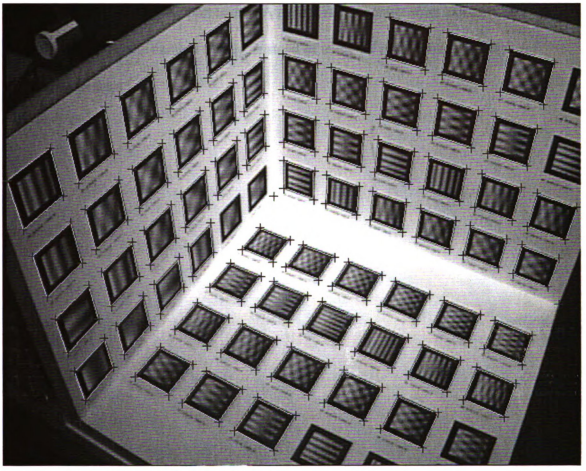


Figure 11-3 - Corner locations after calibration

One significant advantage of using fiducial tracking is that a camera calibration such as the one described here can be done automatically. In most systems, camera calibration has to be done by forcing the user to associate points in the image with points in the real world. This can be a painstaking process, involving much wasted time on the part of the user. Additionally, users rarely will take the time to input as many associations as are used in the example above.

With fiducial identification and tracking, the corner points are found automatically, and the identification of the fiducials allows the system to know what the camera coordinates of those points are. A robust camera calibration can now be done simply by pointing the camera at a jig, instead of using an error-prone, heavily user-interactive process.

References

1. Milgram, P. and F. Kishino, *A Taxonomy of Mixed Reality Visual Displays.* IEICE Transactions on Information Systems, 1994. **E77-D**(12).

2. Ferrari, V., T. Tuytelaars, and L.V. Gool. *Markerless augmented reality with a real-time affine region tracker.* in *IEEE and ACM International Symposium on Augmented Reality (ISAR'01).* 2001. New York, NY.

3. Simon, G., A.W. Fitzgibbon, and A. Zisserman. *Markerless Tracking using Planar Structures in the Scene.* in *International Symposium on Augmented Reality.* 2001. New York, NY.

4. *ARToolKit*, http://www.hitl.washington.edu/research/shared_space/.
5. Fjeld, M. and B.M. Voegtli. *Augmented Chemistry: An Interactive Educational Workbench.* in *The First IEEE International Augmented Reality Toolkit Workshop.* 2002. Darmstadt, Germany.

6. Kato, H. and M. Billinghurst. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System.* in *2nd IEEE and ACM international Workshop on Augmented REALITY (IWAR 99).* 1999. San Fransisco, CA.

7. Rekimoto, J. and Y. Avatsuka. *CyberCode: Designing Augmented Reality Environments with Visual Tags.* in *DARE 2000.* 2000.

8. Appel, M. and N. Navab. *Registration of technical drawings and calibrated images for industrial augmented reality.* in *IEEE Workshop on Applications of Computer Vision.* 2000.

9. *ARVIKA*, http://www.arvika.de/www/index.htm.

10. Zhang, X., S. Fronz, and N. Navab. *Visual Marker Detection and Decoding in AR Systems: A Comparative Study.* in *IEEE and ACM International Symposium on Mixed and Augmented Reality.* 2002. Darmstadt, Germany.

11. Zhang, X. and N. Navab. *Taking AR into large scale industrial environments: Navigation and information access with mobile computers.* in *IEEE International Symposium on Augmeted Reality.* 2001.

12. Ipiña, L.d. *TRIP: a low-Cost Visuion-Based Location System for Ubiquitous Computing.* in *2001 Workshop on Perceptive Computer Interfaces.* 2001.

13. Forsyth, D. and J.L. Mundy, *Invariant Descriptors for 3-D Object Recognition and Pose.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991. **13**(10): p. 971-991.

14. Cho, Y., J. Lee, and U. Neumann. *Multi-ring Color Fiducial Systems and An Intensity-Invariant Detection Method for Scalable Fiducial Tracking Augmented Reality.* in *IEEE International Workshop on Augmented Reality.* 1998.

15. Wolfe, W.J. and D. Mathis, *The Perspective View of Three Points.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991. **13**(1).

16. Naimark, L. and E. Foxlin. *Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker.* in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2002).* 2002. Darmstadt, Germany.

17. Auer, T., A. Pinz, and M. Gervautz. *Tracking in a Multi-user Augmented Reality System.* in *Proceedings of the IASTED Inter91.* 1998.

18. Efrat, A. and C. Gotsman, *Subpixel Image Registration Using Circular Fiducials.* International Journal of Computational Geometry and Applications, 1994. **4**(4): p. 403-422.

19. Yoshimi, B.H. and P. Allen, *Closed-Loop Visual Grasping and Manipulation*: Department of Computer Science, Columbia University.

20. Shapiro, G. and G.C. Stockman, *Computer Vision.* 2001: Prentice Hall.

21. Ji, Q., et al. *An integrated technique for pose estimation from different geometric features.* in *Vision Interface '98.* 1998. Vancouver.

22. Quan, L. and A. Lan, *Linear N-Point Camera Pose Determination.* IEEEIEEE Transactions on Pattern Analysis and Machine Intelligence, 1999. **21**(7).

23. Wearden, B.L.V.D., *Modern Algebra.* 1950, New York: F. Ungar.

24. Liu, M.L. and K.H. Wong, *Pose Estimation using Four Corresponding Points.* 1998: Chinese University of Hong Kong.

25. Park, J., B. Jiang, and U. Neumann. *Vision-Based Pose Computation: Robust and Accurate Augmented Reality Tracking.* in *2nd IEEE and ACM International Workshop on Augmented Reality.* 1999.

26. Sharma, R. and J. Molineros, *Computer vision-based augmented reality.* Presence: Teleoperators and Virtual Environments, 1997. **6**(3): p. 292-317.

27. Lowe, D.G., *Fitting Parameterized Three-Dimensional Models to Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991. **13**: p. 441-450.

28. Yuan, J.S.C., *A General Photogrammetric Methods for Determining Object Position and Orientation.* IEEE Transactions on Robotics & Automation, 89. **15**: p. 129-142.

29. Dementhon, D. and L. Davis, *Model Based Object Pose in 25 Lines of Code.* International Journal of Computer Vision, 1995. **15**: p. 123-141.

30. Strickler, D. and G. Klinker. *A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications.* in *International Workshop on Augmented Reality.* 1998.

31. Jiang, B. and U. Neumann. *Extendible tracking by line auto calibration.* in *IEEE and ACM International Symposium on Augmented Reality.* 2001. New York, NY.

32. Wandell, B.A., A.E. Gamal, and B. Girod, *Common Principles of Image Acquisition Systems and Biological Vision.* Proceedings of the IEEE, 2002. **90**(1).

33. Ribo, M., A. Pinz, and A. Fuhrmann. *A new Optical Tracking System for Augmented Reality Applications.* in *IEEE Instrumentation and Measurement Technology Conference IMTC.* 2001.

34. Welch, G. and e. al. *The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments.* in *Symposium on Virtual Reality Software and Technology.* 1999.

35. Yip, P. and K.R. Rao, *Discrete Cosine Transform: Algorithms, Advantages, and Applications.* 1997: Academic Press.

36. Owen, C.B., F. Xiao, and P. Middlin. *What is the best fiducial.* in *The First IEEE International Augmented Reality Toolkit Workshop.* 2002. Darmstadt, Germany.

37. Tsai, R.Y., *A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses.* IEEE Journal of Robotics and Automation, 1987. **RA-3**(4): p. 323-344.

38. Fischler, M. and R. Bolles, *Random concensus: a paradigm for model fitting with applications in image analysis and automated cartography.* Communications of the ACM, 1981. **24**: p. 381-395.

39. Arun, K.S., T.S. Huang, and S.D. Blostein, *Least-Squares Fitting of Two 3-D Point Sets.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987. **9**(5).

40. Oliensis, J., *A Critique of Structure from Motion Algorithms*, in *NECI Technical Report.* 1997.

41. Hettenlocher, D. and S. Ullman. *Recognizing solid objects by alignment.* in *Proceedings on DARPA Spring Meeting.* 1988.

42. Baratoff, G., A. Neubeck, and H. Regenbrecht. *Interactive Multi-Marker Calibration for Augmented Reality Applications.* in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2002).* 2002. Darmstadt, Germany.