Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker

Leonid Naimark & Eric Foxlin
InterSense Inc.
{leonidn/ericf}@isense.com

Abstract

A wearable low-power hybrid vision-inertial tracker has been demonstrated based on a flexible sensor fusion core architecture, which allows easy reconfiguration by plugging-in different kinds of sensors. A particular prototype implementation consists of one inertial measurement unit and one outward-looking wide-angle Smart Camera, with a built-in DSP to run all required image-processing tasks. The Smart Camera operates on newly designed 2-D bar-coded fiducials printed on a standard black-and-white printer. The fiducial design allows having thousands of different codes, thus enabling uninterrupted tracking throughout a large building or even a campus at very reasonable cost. The system operates in various real-world lighting conditions without any user intervention due to homomorphic image processing algorithms for extracting fiducials in the presence of very non-uniform lighting
.

1. Introduction

One of the most pressing unsolved problems in AR is to develop a tracking system that can provide high accuracy, very low latency, and a very wide tracking area. For many AR applications, including large-scale manufacturing, and outdoor applications, the area of operation is so large that a wearable computer must be used to drive the AR head-mounted display. In order to keep the mobility of the wearable computer, a wearable self-tracking system is needed, with as little environmental infrastructure as possible. Researchers have set to work diligently trying to develop a vision-based self-tracking solution that can meet these difficult challenges, but so far a robust, practical and general wide-area tracker in a low-power wearable form factor is still lacking.

To simplify the considerable computer vision challenges, many researchers have used artificial landmarks or fiducials (e.g. [11], [12], [14], [16], [17]), but even these have not yet yielded robust trackers, because when the camera moves quickly, the image processor has to search the whole image to find candidate fiducial marks, and this is too slow and unreliable. The system easily becomes disoriented and takes a long time to recover. In order to build a robust and fast self-tracker, there is growing consensus that you need to combine inertial and computer vision technologies (e.g. [2], [12], [18], [19], [20]). We have implemented a hybrid vision-inertial self-tracker (hereafter "VIS-Tracker"), which was demonstrated in an early form at ISAR last year, and in a much more developed form at the International Conference on Robotics and Automation (ICRA2002) in May. To our knowledge, this is the first system which is in a convenient belt-mounted form factor, can run several hours on a battery, and can track robustly throughout a large building. The VIS-Tracker is described in general terms in [8], and [7] explains the unique modular sensor fusion architecture on which it is based.

The current paper adds an in-depth description of the vision subsystem and the inter-related issues of fiducial design and image processing algorithms that are critical to the performance of the tracker. The primary contributions are:

- A new fiducial design which offers much higher information density for a given area than previous designs, in which the coding is essentially 1-D, yet provides the sub-pixel accuracy of centroid location which makes circular fiducials popular.
- A strategy for detecting, reading and tracking the fiducials based on homomorphic image processing which is fast and simple, yet robust in the presence of varied real-world lighting conditions.
- An implementation running on a low-cost and low-power 16-bit DSP, which we have tested successfully in a variety of environments.

These issues are important, because even with a hybrid tracker we have found that standard fiducial designs such as concentric circles do not offer enough unique ID codes to install the system in a truly wide-are setting such as a factory without ambiguity, and standard image processing routines used to detect the fiducials are prone to failure in real-world situations where lighting cannot be controlled. A few works have tried to address these problems, but usually at too much computational cost for today's wearable computers. We will review some of these works in Section 3 on fiducial design

and Section 4 on image processing strategy after first giving a brief overview of the hybrid vision-inertial tracker hardware and operating modes in Section 2.

2. System overview

Figure 1 shows the basic architecture of the system. Each self-tracker consists of an inertial measurement unit (IMU), a sensor fusion core (SFC), and one or more smart camera (SC) devices.

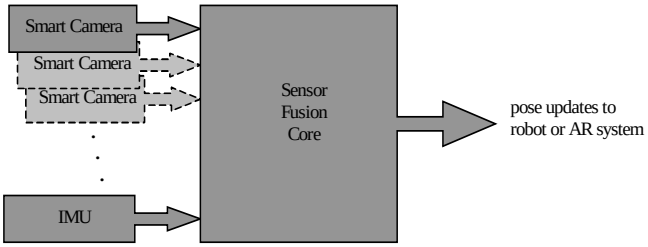

Figure 1: Block diagram of basic self-tracking system architecture

In our current implementation the SC is a CCD with an attached DSP to detect artificial fiducials in the image, read their codes during initialization, and extract their centroids during tracking. However, at the architectural level, the SCs are very general sensors, and may even include lasers, sonars or radar units.

Figure 2 shows a photograph of our complete prototype system, including the head-mounted sensor assembly of camera and InertiaCube, and the belt-mounted electronics unit which houses the CPU board, a battery, and the DSP image processor board (which is hidden beneath the CPU board).

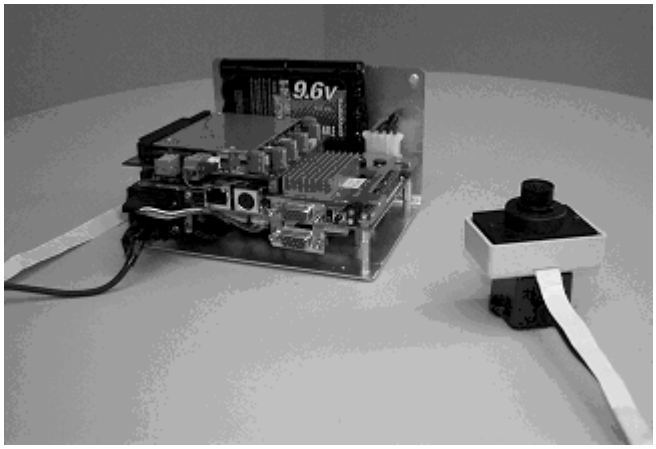

Figure 2: Photograph of prototype self-tracking system.

The belt-mounted electronics unit is about 100 mm X 150 mm X 80 mm, and weighs 600 g without batteries. It consumes 13 W of power, which adds about 100 g of Lithium-ion batteries per hour of desired battery life. The sensor head is about 35 mm deep, 50 mm tall, 30 mm wide at the base and 57 mm wide at the camera head, and weighs 70 g. Figure 3 shows a person wearing the VIS-Tracker on an HMD and walking around a lobby test area with fiducials installed. As can be seen, we typically tend to place most of the fiducials on the ceiling when the goal is to allow a user to walk around throughout a large facility with continuous tracking, but they could also be placed exclusively on the walls, with the head-mounted camera facing forwards, or for the ultimate performance, one could have a mixture of wall- and ceiling-mounted fiducials, and a system which incorporates two SCs facing up and forwards.

Figure 3: Person wearing self-tracker on an HMD walking around the lobby test area.

For the SC, we use an image processing board based on the ADSP 2185 chip, and a 640x480 progressive-scan black & white CCD with a micro-video lens having 108° diagonal field of view. Such wide-angle lenses suffer from huge distortion, especially in the corners of the image. We implemented a lens calibration procedure and applied compensation algorithms using the techniques in [10] and [21]. Although the SC can theoretically process up to 30 frames/sec, the actual number of frames is smaller and depends on shutter speed. Image transfer from the CCD takes 33 ms, so for 8 ms shutter speed we have 1/(.033+.008) = 24.4 frames/sec. We programmed the camera to perform three main functions in response to serial port commands from the SFC:

1. **Acquisition**: When the SFC is first trying to acquire its pose after the system is turned on, or when trying to re-acquire after the system has become lost due to a prolonged occlusion, it sends the acquisition command to the SC, which searches the entire image to find any and all fiducials in view, and returns the u,v coordinates and ID codes of the best 4 or 5 of them. The SFC looks up the pre-stored 3-D x,y,z locations of these fiducials in its map, and then solves a 4-point pose recovery algorithm.
2. **Tracking**: Once an initial pose estimate has been found, the SFC enters tracking mode in which the

measurements from the SC are only used to make small corrections to the pose calculated by the inertial sensor, through a complementary Kalman filter. Using the predicted pose of the inertial tracker, the SFC sends a tracking command requesting the SC to only search in a small rectangular area about the location where it expects to see a certain fiducial in the image, and return the precise u,v coordinates of the centroid of the blob found. Because the search box is normally only large enough to contain one fiducial, there is no need to read the barcode, and image processing is performed very fast.

3. "**Tracquisition**": This SC command is used during automapping mode to build a map of the fiducial constellation after installation. The user must first acquire using 4 "seed" fiducials whose x,y,z locations are measured manually and downloaded to the tracker. Once the tracker enters tracking mode, the user may then switch it into automapping mode, which uses a simultaneous localization and map-building (SLAM) filter based on an augmented state vector containing estimates of all the initialized fiducial positions in addition to the tracker states [7]. During automapping mode, the SFC schedules mostly measurements of already observed fiducials using the tracking command, supplemented with about 10% exploratory measurements using the "tracquisition" command to ask the SC to try to find new fiducials to add to the map. The task of the SC for "tracquisition" is similar to tracking in that only a portion of the image needs to be searched, but similar to acquistion in that any objects found in that area must have their barcodes read to determine that they are bona fide fiducials. When the SC returns the barcode, center, and size of a newly found fiducial, the SFC initializes a new point in the growing map with the appropriate ID code and an initial position estimate deduced from the returned u,v position and approximate distance, then proceeds to refine the position estimate using the SLAM filter as subsequent tracking measurements come in.

3. Fiducial design

Many researchers have investigated different aspects of fiducial design. Perhaps the most popular type of fiducial is the contrasting concentric circle (CCC) [9] with a sequence of black and white (or color [5] or retroreflective [4]) rings. In [15], the accuracy of locating circular features using machine vision is investigated. Circular fiducials have the advantage that the center is the centroid, is invariant to viewing direction and angle (except for a small perspective distortion offset which can be compensated exactly if the viewing angle is known), and can be determined with sub-pixel accuracy which increases with the size of the fiducial because the larger the number of pixels going into the centroid calculation, the more the individual pixel noise averages out. The CCC approach inherits these advantages of the ordinary circle, but adds a reliable check that an object is a fiducial because the centroids of all the rings must coincide, and has an advantage that fiducials can be recognized from a wide range of distances. However the ability to generate significant numbers of different codes is poor, making this approach applicable with at most a few hundred fiducials installed.

Another widely used type is a square box fiducial with some codes inside (e.g. [3],[17],[20],[22]). For example in ARToolKit [3] a set of Japanese characters has been used. Square fiducials have an advantage that lines are invariant under perspective transformation, so if the lens distortion is small then the fiducial boundary can be identified as 4 intersecting lines, which can be found quickly by testing colinearity of several test points [17]. Inside the square box one may place some barcode-type patterns or any arbitrary designs for template matching. The latter approach is limited though, because for any large database, template matching becomes a computationally expensive procedure with high-rate of false alarms. Worse yet, if lens distortion is significant, then image processing would need to undistort the whole image in real-time in order to apply accurate template matching. Since the centroid of a square under perspective transformation shifts depending on viewing direction, one must instead find the center using the corners, for example by intersecting the two diagonal lines which connect opposite corners. However, to find the corners with the same degree of noise-reducing sub-pixel averaging as the centroid of a circle would require first fitting lines to a large number of sample points found along the boundary edges, which would be much more expensive than the simple centroid operation used for circles.

3.1. Goals and assumptions

All the previous fiducial designs proposed in AR papers have been unable to generate sufficient number of different codes, and thus cannot be used for really wide-area tracking applications. Our goal was to design of fiducials with the following properties:

- Fiducials can be robustly extracted in real time from the scene including barcode reading with almost zero false alarm rate, using a simple low-power DSP
- A uniquely defined point on the fiducial can be found with great accuracy and speed
- Thousands of different codes can be generated
- Black and white fiducials can be printed on a standard laser or ink-jet printer on 8.5x11 inch paper for office-high ceiling and on 11x17 inch paper for high-ceiling buildings (>3.5m)

Our fiducial and image processing design were done in parallel, within the context of some constraints we set for ourselves in order to make sure to develop a practical and affordable product. To insure that the algorithms we

developed would run on low-power low-cost hardware, we first surveyed "smart camera" products and dedicated single-chip image processors or imaging sensors with built-in image processing capabilities that are available today. For the most promising hardware candidates, we looked in detail at the image processing operations that were provided and their timings. We found that simple 8-bit grayscale processing such as convolutional filtering with kernels up to 5 x 5 pixels, simple image area statistics and histograms, look-up tables (LUTs), thresholding, and basic binary morphology operations such as erosion and dilation were usually available and fast. Also, the better chips or libraries were able to provide segmentation and primitive feature extraction (area, bounding box, etc) of binary images in a few milliseconds.

As a result of this product search, and also of our suspicion that a 640 X 480 color camera would have less effective resolution for determining the exact geometric center of a blob than would a 640 x 480 black-and-white camera, we decided to stick to black-and-white fiducials and image processing, even though this almost certainly makes the task of first locating candidate fiducials much harder. Given our hybrid inertial system design, most of the time (i.e. during tracking) we would be just finding the exact center of a fiducial which we already could predict within a few pixels, so speed and precision in tracking seemed more important than speed in acquisition which was only an occasional task. Likewise, for the sake of speed and precision in tracking mode, we chose to pursue a circular fiducial design. Because of the built-in "blob analysis" functions in our candidate hardware, we wanted as well to design a fiducial that would read as a single fully-connected blob, so that by the time the segmentation operation was done we could get the centroid result with no further processing.

3.2. Evolution of the fiducial design

Our first generation fiducial is shown in Figure 4(a). Here we tried to increase the number of codes by dividing the fiducial into 4 sectors and reading codes in two orthogonal directions. Compared to an ordinary multi-ring barcode in which the code is read only along one scan line through the center, this design could potentially allow almost twice as many bits of information for a given size. However, in order to guarantee that stray objects were never mistaken for fiducials, we restricted the design to have one ring of white in each direction (so we could count 6 B/W transitions in each direction for a true fiducial), and in order to keep it as a singly-connected black blob, we disallowed codes in which the stripe in one direction was at the same or adjacent position to the stripe in the other direction. As a result, a fiducial with 7 active rings could only produce a few tens of different codes.

Figure 4: (a) First generation fiducial (b) Second generation fiducial (c) 6 x 6 data matrix 2-D barcode

We modified this design and demonstrated a system using second-generation fiducials (Figure 4(b)) in the Emerging Technologies gallery at SIGGRAPH 2001 [6]. This design guaranteed connectivity through the black cut-outs, and we determined that we could adequately test for "fiducialness" by testing the symmetry of the color-change points on the left and right of the center, so we allowed all 2^7 codes, equivalent to an ordinary CCC fiducial with 7 active code rings.

These first two designs were fast and easy to process. During tracking, the symmetrical disposition of the white holes caused their effects on the centroid of the black circle to cancel out, so we could just find the single blob and report its centroid position as if it were a solid black circle. However, they were totally unsatisfactory in terms of information density. If we wanted thousands of codes we would need a dozen rings or more, which would never fit on an 8.5x11 inch paper and be viewable from a good distance.

Clearly we needed a 2D barcode scheme similar to the DataMatrix codes which are now widely used for product identification labels because they take so much less space than traditional 1-D barcodes with comparable capacity [1]. As Figure 4(c) illustrates, even an area just 6 times as wide as the basic feature size can encode 14 bits of data (as compared to 7 bits for a CCC with diameter 17 times the feature size).

The perceived obstacle was that the nonsymmetrical distribution of white areas inside a circular data matrix fiducial would make it impossible to find the center using the simple centroid operation we depend on, and trying to fit an ellipse to the outside edge would be much too slow and inaccurate. It occurred to us that one solution would be to find the fiducial, read the barcode if necessary (acquisition or tracquisition mode), and then flood fill all the interior white areas with black to make a solid circle and find its centroid. However, we found that this filling operation was too slow on our hardware.

The breakthrough occurred when we realized there was no need to actually fill the fiducial, and invented a "virtual filling" algorithm that allowed us to find the exact centroid as if it had been filled, without performing any additional image processing operations. The key to the virtual filling algorithm is that the segmentation operation that finds the centroid and area of the hole-riddled black object also finds the centroid and area of each of the white objects that are its holes. It is easy to identify all the white objects that are

completely inside the black object, and then the centroid of the filled object is just the sum of the centroid times the area of all the objects which make it up.

3.3. New 2D bar-coded fiducials

Armed with a virtual filling algorithm that allows us to have asymmetrical white areas in the interior, we combined the DataMatrix concept with the benefits of the CCC to produce our third and final fiducial design, which was demonstrated at ISAR last year. Three of them are shown in Figure 5. This particular sub-design has 2^{15}=32768 possible codes and will be described below in detail.

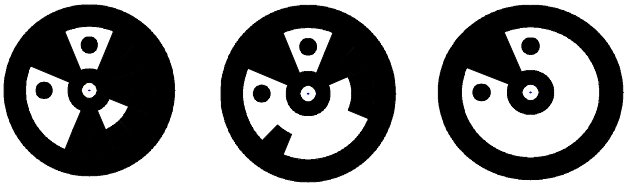

Figure 5: Fiducials with barcodes 101, 1967, 32767.

Suppose that the fiducial diameter $D = 8u$, where u is one length unit. If a fiducial is 6-inch diameter then $u = 6/8 = 0.75$ inches. Every fiducial has an outer black ring of 1u width, two data rings, each of which is 1u wide, and an inner black ring with 1u width.

1. **Structure of outer ring:** This ring is always black.
2. **Structure of inner ring:** The inner ring consists of a black circle of diameter 2u with a white "eye" in the center with 3/8u diameter. This eye is used in acquisition and tracquisition.
3. **Structure of data rings**: Both data rings are divided into 8 equal sectors with 45^o degrees for each sector. 3 sectors have specific structure. Two of them are used to determine fiducial orientation for identifying the barcode reading grid points. These sectors are white with a 3/8u black eye inside. They are located 17/8u from the center. The three eyes form a 45-45-90 triangle. Between these two sectors there is a special black sector, which guarantees connectivity of all the black color (except the two black eyes) into a single blob. The other 5 sectors contain actual data cells, 3 per sector, totaling 15 quasi-trapezoidal data cells, which can be colored either black or white.

If there is a need to further increase this number, then several options can be considered. We mention here just 5 of them. The first two keep the size and structure of the shape the same, but make barcode construction slightly more complicated. In this case the totally black sector is used for coding as well. A straightforward extension is to allow 3 combinations of its outer data ring cells, being white-black, black-white or black-black. Then the total number of codes is 3×2^{15}. We can increase this number even further almost to 2^{18} if we allow arbitrary colorings in this sector, but then we have to exclude codes where there is no connectivity between the inner and outer black rings.

Two other ways to increase the number of codes are to increase the number of data sectors or rings. Having 7 data sectors with data instead of 5 increases the number of codes to 2^{21}. A shortcoming of this approach is that data cells are smaller and there are more false alarms, or a bigger size of fiducials is required. Another extension that requires bigger size but gives an extremely large number of barcodes is to have 3 (or more) data rings instead. Fortunately, increasing the size here by 20% increases the number of possible codes to 2^{40} or more. With a few more rings, this design might even be useful in traditional 2-D barcoding applications. A fifth way to increase the number of codes would be to use color. With 6 colors, the number of codes jumps from 2^{15} to 6^{15}. However, as already mentioned, the use of a color camera would make it more difficult to find the true centroid of the overall circle, since the centroids of differently colored regions are shifted in different directions according to the pattern of pixel color assignments in the Bayer filter of the camera.

In the sequel we will consider only the case with 5 data sectors and 2^{15} barcodes. Even this number basically allows tracking an area of 10,000-50,000m^2 (depending on ceiling height), which is enough to cover almost any building.

4. Image processing algorithms

Even in a hybrid system, robust performance is hard to achieve when the system is required to track over an extended area. Most vision-based systems can operate only in specific lighting conditions or must be manually tuned-up for specific lighting scenarios. In wide-area tracking, the user will move from room to room with different light conditions, so the image-processing algorithms should be robust enough to handle various conditions including daylight through the windows, fluorescent light from the ceiling in offices, and showroom lights with strong point sources and weak background illumination. Any combination of these is possible, including situations where the ceiling-mounted fiducials are very dimly lit and subject to strong gradients in illumination, and the system should adapt without user intervention.

In most of the cited fiducial-tracking literature, little attention was paid to the problem of variable lighting, and fiducial objects were extracted through color segmentation or simple thresholding. In [5], a more complicated scheme is presented using a rule-based approach that looks for groups of samples of similar color that are likely to all belong to the same fiducial. However, this approach depends completely on use of a color camera and image processing, and also requires a specifically designed CCC fiducial with even lower information density than a generic multi-ring fiducial. A more general attempt to overcome lighting difficulties in AR vision-based tracking has been done in a sequence of papers starting from [22] where the watershed algorithm has

been used for segmentation. However, the watershed is computationally costly and difficult to run for real time applications using reasonable portable CPUs.

4.1. Homomorphic image processing motivation

Our goal is simply to distinguish black and white areas of paper, which sounds like an easy task for a simple threshold operation. The problem is that the grayscale values of both black and white regions vary tremendously according to the illumination. Under normal daylight conditions, white paper can vary from 70 to 255 in 8-bit gray scale values, while black is typically 20-50. However, a fiducial located near a window may have a black level of 80, while in the opposite corner of the same room, the level of white would be 55. Under just incandescent lighting, the white level can be as low as 5-10, while black level is 0-3.

The situation may become even worse when there is a strong light toward the camera from the ceiling and the fiducial to be detected is placed near this light. The light causes a blooming effect in the image sensor, in which the brightness is elevated in the area surrounding the light, falling off with distance from the light. The black level of the part of the fiducial near the light is higher than the white level of the same fiducial on its further side. For example, the white level can decrease from 120 to 55 along the sheet of paper, while on the same fiducial black is going down from 70 to 30.

At first, we tried to use an adaptive algorithm to find an optimal threshold value that could separate black and white in a given frame, but even with the best possible threshold value we could rarely separate even half of the fiducials in an image from the background with one threshold. We tried thresholding on multiple values, so that lower thresholds would separate the fiducials on the darker side of the room, while higher thresholds would find the ones in the light. We found that to detect 90% of the fiducials would require at least 10 different thresholds to be applied, if it could happen at all. Unfortunately, this approach is way too expensive, because for each threshold value one must again binarize the image and perform segmentation and labeling, which takes significant time. And of course those fiducials affected by blooming could never be detected using any threshold.

At this point, we decided to apply a modified form of homomorphic image processing [13, p.463], which is designed to eliminate the effect of non-uniform lighting in images. The concept is to model the grayscale image as a product of illumination and reflectance values:

$$f(n_1,n_2) = i(n_1,n_2)r(n_1,n_2) \qquad (1.1)$$

and assume that the illumination factor $i(n_1,n_2)$ varies relatively slowly compared to the reflectance $r(n_1,n_2)$. In ordinary homomorphic processing, you take the logarithm of the image in order to separate i and r into additive terms:

$$\log f(n_1,n_2) = \log i(n_1,n_2) + \log r(n_1,n_2) \qquad (1.2)$$

then apply a high-pass filter designed to attenuate the slowly-varying illumination term, then exponentiate to get back a good looking image. Normally, the high-pass filter is chosen with a cutoff frequency just above 1 cycle/image, in order to kill the illumination gradient without affecting the reflectance image too much. This requires a large and expensive filter. Realizing that we were not concerned with restoring a good-looking image at the end, we decided to use a much higher cut-off frequency that could be implemented with a small convolutional kernel. Taking this to the extreme we settled on a 3 x 3 Sobel edge detector. This allowed us to find the edges of all the fiducials in the image, with complete immunity to any lighting variations gradual relative to the 3-pixel width of our filter. Even the relatively steep gradients produced by blooming near the lights are gradual on this 3-pixel scale, and we are now able to reliably extract fiducials that are quite close to bright lights.

As a second modification, we determined that it was advantageous to skip the final exponentiation that is normally used to restore the image to human-viewable brightness levels. This not only saves time, but allows us to use a constant fixed threshold to find all the black/white edges in every image, regardless of lighting conditions. To see this, consider that the difference in reflectance between white paper and black ink is always at least a factor of 2.5, which we determined experimentally by taking measurements of nearby pixels on black and white parts of fiducials from a wide variety of test images.

When we take the log of an image consisting of grayscale values $f \in [0,...,255]$, we create a new image

$$p(n_1,n_2) = 105.89 * \log(f(n_1,n_2)+1) \qquad (1.3)$$

where scaling has been applied so that p is also in the range [0,…,255]. Consider the grayscale difference between processed pixels on the black side and the white side of an edge:

$$\Delta p \triangleq p_w - p_b = 105.89 * \log\frac{f_w+1}{f_b+1} \geq 105.89 * \log 2.5 = 42$$

So the logarithm (which we implemented with a fast LUT) serves as a contrast enhancement operation that guarantees that the contrast between neighboring black and white pixels will always be at least 40 gray levels, whether their original intensities had been 1 and 3, 5 and 13, or 100 and 250.

Following contrast enhancement, we apply a Sobel edge detector, which is good for extracting circular edges, and runs fast on simple hardware. The edge-detected image will have values of at least 20 along strong black/white boundaries with contrast of at least 40, and much lower values elsewhere. Remarkably, for any original image one can now use a fixed value of threshold to find all the true black/white edges. We found that a threshold of 11 (after a blur filter to attenuate small objects and soft edges) guarantees that at least 90% of the original fiducials are usually detected. We use this basic sequence of contrast enhancement and edge detection in both acquisition and tracking routines, which are described in detail in the following sections.

4.2. Acquisition: finding candidate objects

Figure 6 illustrates the sequence of operations used to identify potential candidate objects that might be fiducials. The candidates are then tested one at a time to see if they have barcodes that can be read, as described in Section 4.3.

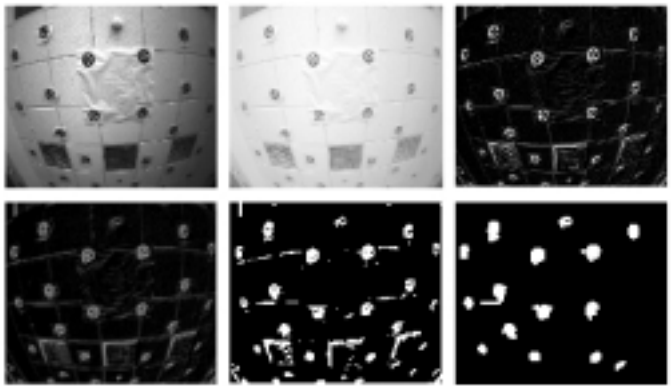

Figure 6: Selected steps in finding fiducial candidates: (a) downsample original image to 320 x 240 for speed, (b) contrast enhancement, (c) edge detection, (d) blur filter, (e) binarization with threshold value of 11, (f) final candidates selection

The final candidate selection involves several operations:

Image erosion: There are still too many small objects in the image. Due to the nature of our hardware it is very fast to implement an erosion operation along the horizontal direction only, which removes objects 1 or 2 pixels thick.

Feature extraction. At this stage we apply a standard feature extraction procedure, where all remaining objects (typically 10-80) are labeled and their properties are computed (color, area in pixels and min and max extent in both *u* and *v* directions).

Size, color and area tests:

1. **Maximal size:** Objects greater than 80 pixels in either direction are thrown out.
2. **Minimal size**: Objects smaller than 16 pixels in either direction are thrown out, since with feature size less than 2 pixels it might not be possible to correctly read their barcode in the next stage. We intend to improve the barcode reading algorithm soon so that objects as small as 10 x 10 pixels can be read.
3. **Color**: Only white objects are considered.
4. **Area ratio test**: Since fiducials are circles, they should appear approximately elliptical in images, which implies

$$A \approx \frac{\pi}{4}(u_{\max} - u_{\min})(v_{\max} - v_{\min}).$$

Applying this formula with some margins allows omitting narrow and L-shaped objects.

The resulting image after all weed-out tests is shown in Figure 6(f). Typically at this stage we have 90% of the good fiducials plus a few extraneous objects selected as candidates for barcode reading.

4.3. Acquisition: barcode reading

When candidate selection is finished, then each successful candidate is checked for a barcode as illustrated in Figure 7. We return to the original grayscale image and compute a histogram for each sub-area surrounding a fiducial candidate. The minimum of the histogram between the peaks corresponding to black and white (in this case 71) is used as a threshold for binarization. If the binarized image contains exactly three black objects, two of which are small black "eyes" inside of the large object, and there is one small white "eye" in the center of the large object, then it is considered a fiducial. The white eye is used as an origin, and the vectors to the black eyes are used as u and v-axis basis vectors for determining the 15 test points where we read a binary value to decode the fiducial. For the current fiducial design, there are 8 pie-shaped sectors, each of which contains 3 test points (along the lines in the 4th figure).

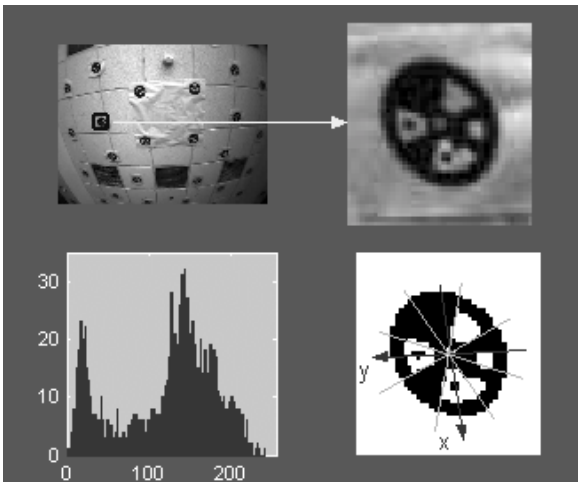

Figure 7: Barcode reading sequence (original picture, area with 1 fiducial selected, histogram for this fiducial, binarized fiducial at level 71 with axes and sectors for barcode reading).

Finally, if there are 4 or more fiducials detected, then the best 4 of them are sent to the SFC for initial camera pose determination. When more than 4 fiducials have been read, the camera picks the four which form a quadrilateral with greatest area, in order to minimize geometric sensitivity of the pose recovery algorithm.

4.4. Tracking

The acquisition process takes about 250ms to complete. Tracking cannot have such a luxury, because high update rate and low latency are very important. Although tracking

operates on a much smaller window (typically 30x40 versus 320x240) it is still impossible to implement the complete acquisition sequence. Fortunately, since the SFC predicts the fiducial location within several pixels, there is no need to identify candidates or check barcodes.

The complete tracking sequence is illustrated in Figure 8. Contrast enhancement is followed by edge detection, and then local binarization is applied. The threshold value that we used is again constant for all lighting scenarios, but different from the one used for acquisition, because the blur filter is not used for tracking. At this stage (4th image) we have a big mass of black pixels related to the inspected fiducial, and some smaller objects related to paper edges and noise. They are removed by size and area tests similar to those used for acquisition. Finally, the fiducial is virtually filled as described in Section 3.2 and its center of gravity is computed (shown in 5th image).

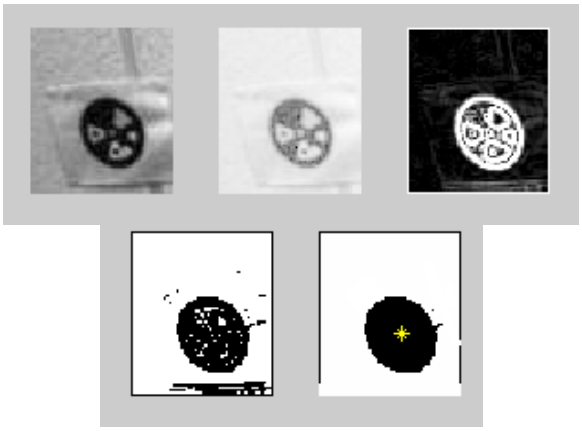

Figure 8: Stages of tracking (original image, contrast enhancement, edge detection, binarization, virtual filling)

Before sending results back to the SFC, one additional test is applied to check symmetry. If the difference between the center of the bounding box and the center of gravity is too big, then the camera reports that the fiducial failed the symmetry test and the Kalman filter does not use the measurement. This test is very effective at rejecting fiducials that are partially washed out or have been altered in appearance by objects between the camera and the fiducial paper.

The companion video clip illustrates the performance of the SC during several different tracking sequences. The image processing board has a VGA output that displays the grayscale images captured by the CCD, with overlays that we have added for debugging and demonstration purposes. The blue boxes show the size and position of the small search areas in which image processing is performed during tracking. They are provided by predictions sent from the SFC to the SC. If a blob is found inside this area which passes all size and symmetry tests, a blue plus sign is also added showing the centroid location and size of the found circle. As the video shows, the camera is able to accurately find the fiducial inside the search box about 90% of the time even in difficult lighting conditions. There is also a sequence where we intentionally occlude the camera with a hand so that most of the fiducials cannot be found. This demonstrates that 1) the camera is able to reject false positives through use of it's size and symmetry tests, and 2) even an occasional accurate measurement return is sufficient to keep the tracking going in this hybrid vision-inertial system.

5. Results and comparisons

5.1. Acquisition for difficult lighting

We have touted the robustness of the system to various lighting conditions. To illustrate this point, Figure 9 shows a scenario with a strong light source facing toward the camera lens. This light source creates a blooming effect in the CCD, and a light gradient across the image. Also, the image has really dark areas where markers are almost invisible to the human eye. The original image is shown on Figure 9 (a).

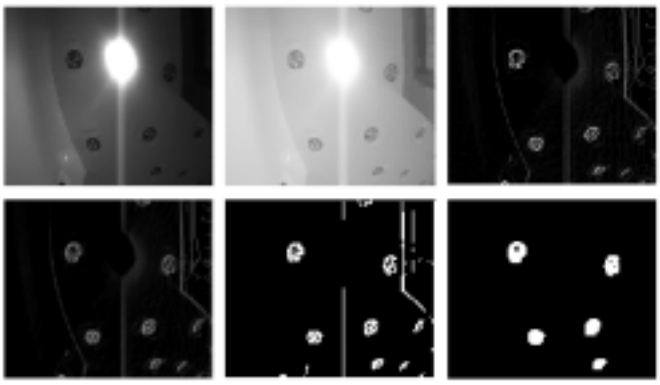

Figure 9: Steps in finding fiducial candidates when light conditions are extreme.

After contrast enhancement, all the fiducials can be seen with equal contrast. Edge detection removes the light source because the blooming effect has soft edges. Finally, stray lines and the two smallest fiducials are removed by size, area and shape considerations. The resulting 5 fiducials can be read and used for successful acquisition.

5.2. Comparison with global threshold methods

Although most of the steps used for image processing are known in the computer vision literature, they have not been used in AR vision-based tracking before. Typical AR fiducial finders, such as ARToolKit, have used a global threshold to binarize and segment the image. We will now compare our approach with the standard global threshold technique based on the same set of images. For the first image from Figure 9, the mean value, 71, has been computed and used as a threshold with the result shown in Figure 11(a). Clearly, the result is absolutely unacceptable.

Then we applied histogram analysis trying to obtain a better threshold value. The histogram is shown in Figure 10(a), and the result of binarization with threshold 17 is shown in Figure 11(b). The result is certainly better, but still 2 fiducials on the left are not detected, and those on the right are very noisy and unconnected, which makes them unusable.

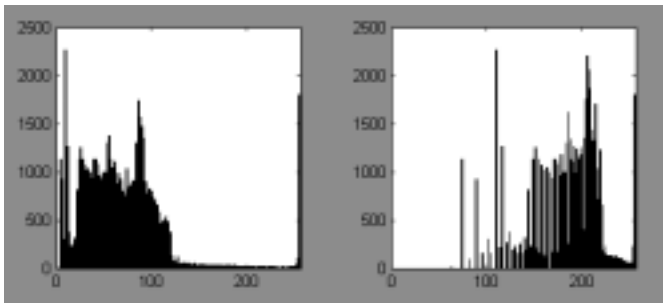

Figure 10: Histograms of Figure 9 (a) and (b).

Figure 10(b) shows the histogram of the contrast-enhanced image, which is even more difficult to analyze.

Figure 11: Binarization of image on Figure 9 (using mean value, using histogram, using manually found threshold)

Finally, Figure 11(c) shows a binarization with threshold 27, which was manually found to be the best threshold value. Still, it is far less successful than our homomorphic image processing strategy, as the two left fiducials are already disconnected, while the two right ones are touching the black background and cannot be segmented out.

5.3. Tracking in low light

The tracking-mode image processing algorithms are also very robust to different lighting conditions. Figure 12(a) illustrates a typical low-light scenario, where black grayscale values are 0-1, and white values are 1-3. Nothing is visible to the human eye. Applying the logarithmic contrast enhancement operation, we managed to get a visible result as shown in Figure 12 (b), resulting finally in an accurate centroid as shown in Figure 12(e). There are additional sequences on the video clip which demonstrate that tracking works fine in very low light conditions, as you would expect from Figure 12.

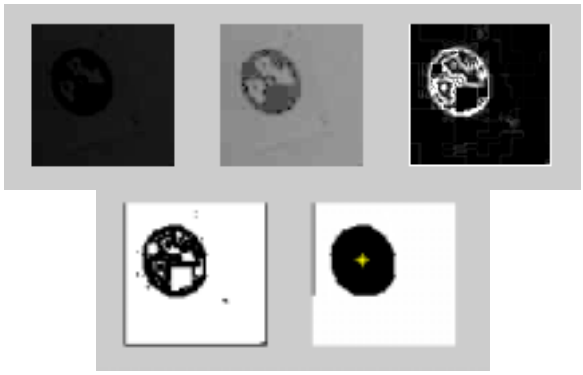

Figure 12: Stages of tracking for extreme light conditions

6. Discussion and conclusion

In this paper we presented three main accomplishments:

- A wearable hybrid vision-inertial tracker design and implementation.
- A new 2D barcode fiducial design that can generate thousands of different codes and can be used for wide area tracking.
- Robust image processing algorithms to extract fiducials from the scene in tracking, acquisition and tracquisition modes and to read their barcodes.

Unlike most previous papers published on vision-based tracking for AR, this work was done with the goal of developing a real commercial product that can be put to work in factories and other semi-hostile environments where AR is likely to make an economic impact. As such, we had to focus a lot of energy on some practical problems that are normally glossed over by AR researchers. The results have been very satisfactory, and the product can be put into production within months as soon as a volume application for AR emerges. However, there are a few more issues we still would like to address to make the system even better:

- We will test an adaptive shutter speed enhancement that should let it operate outdoors in bright sun.
- We suspect we can reduce the minimum image size of fiducials for barcode reading in acquisition and tracquisition to about 12 x 12 pixels, yielding a big improvement in the ceiling height/fiducial size ratio. This will be done by reading each data bit from the barcode by taking a weighted grayscale average of the four nearest pixels.
- We would like to vastly reduce the number of artificial fiducials needed by using mostly natural features. This is easy to get working for tracking mode, once a map of the natural features exists. We are working on techniques to initialize natural features into the map during automapping mode, even when their distance can not be guessed from the first sighting, and to identify constellations of natural features which are sufficiently unique for acquisition.

7. Bibliography

[1] AIM (Automatic Identification Manufacturers). (1999). Understanding 2D symbologies. Halifax, UK. (webcite: www.aimglobal.org).

[2] Azuma, R., Hoff, B., Neely, H. and Sarfaty, R. (1999). A Motion Stabilized Outdoor Augmented Reality System. Proceedings of IEEE VR99, pp. 252-259.

[3] Billinghurst, M. and Kato, H. (1999). Collaborative Mixed Reality. Proceedings of 1st Intl. Symposium on Mixed Reality (ISMR99), pp. 261-284.

[4] British Broadcasting Corporation (BBC) (1998). Position Determination. International Patent No WO 98/54593, 1998.

[5] Cho, Y., Lee, W.J. and Neumann, U. (1998) A Multi-ring Color Fiducial System and Intensity-Invariant Detection Method for Scalable Fiducial-tracking Augmented Reality. Proceedings of the 1st Intl. Workshop on Augmented Reality (IWAR 98), San Francisco

[6] Feiner, S., Bell, B., Gagas, E., Guven, S., Hallaway, D., Hollerer, T., Lok, S., Tinna, N., Yamamoto, R., Julier, S., Baillot, Y., Brown, D., Lanzagorta, M., Butz, A., Foxlin, E., Harrington, M., Naimark, L. and Wormell, D. (2001). Mobile Augmented Reality Systems. ACM SIGGRAPH '01 Emerging Technologies, Los Angeles, CA, Aug. 12-17, 2001.

[7] Foxlin, E. (2002). Generalized Architecture for Simultaneous Localization, Auto-Calibration, and Map-building. IEEE/RSJ Intelligent Robotic Systems Conference (IROS 2002), Lausanne, Switzerland.

[8] Foxlin, E. and Naimark, L. (2002). Wearable Vision-Inertial Self-Tracker. In preparation.

[9] Gatrell, L., Hoff, W. and Sklair, C. (1991). Robust Image Features: Concentric Contrasting Circles and Their Image Extraction. SPIE Proc. Conf. Intelligent Robotic Systems, Boston, Vol. 1612.

[10] Heikkila, J. and Silven, O. (1997). A Four-Step Camera Calibration Procedure with Implicit Image Correction. IEEE Computer Vision and Pattern Recognition, pp. 1106-1112.

[11] Hoff, W., Nguyen, K. and Lyon, T. (1996). Computer Vision-Based Registration Techniques for Augmented Reality. Proceedings of IRCV, SPIE Vol. 2904, pp. 538-548.

[12] Kanbara, M., Fujii, H., Takemura, H. and Yokoya, N. (2000). A Stereo Vision-Based Augmented Reality System with Inertial Sensor. Proceedings of ISAR2000, pp 97-100.

[13] Lim, J. (1990). Two-Dimensional Signal and Image Processing. Prentice Hall, Englewood Cliffs, New Jersey.

[14] Neumann, U. and Cho, Y. (1996) A Self-Tracking Augmented Reality System. Proceedings of ACM VRST 96, pp. 109-115.

[15] Sklair, C., Hoff, W. and Gatrell, L. (1991). Accuracy of Locating Circular Features Using Machine Vision, SPIE Proc. Conf. Intelligent Robotic Systems, Boston, Vol. 1612.

[16] State, A., Hirota, J., Chen, D.T., Garret, B. and Livingston, M. (1996). Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. Proc. of SIGGRAPH'96, pp. 429-438.

[17] Stricker, D., Klinker, G. and Reiners, D. (1998). A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications. Proceedings of IWAR 98.

[18] Welch, G. (1995). Hybrid Self-Tracker: An Inertial/Optical Hybrid Three-Dimensional Tracking System. University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, USA TR95-048, 1995.

[19] Yokokohji, Y., Eto, D. and Yoshikawa, T. (2001). It's Really Sticking! –Dynamically Accurate Image Overlay Through Hybrid Vision/Inertial Tracking, Proceedings of ISMR, pp. 196-197.

[20] You, S. and Neumann, U. (2001). Fusion of Vision and Gyro Tracking for Robust Augmented Reality Applications. Proceedings of IEEE VR2001, pp 71-78.

[21] Zhang, Z. (1999). Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. ICCV'99, Corfu, Greece, pp. 666-673.

[22] Zhang, X., Navab, N. and Liou, S. (2000) E-commerce direct marketing using augmented reality. Proc. of IEEE Int. Conf. On Multimedia & Expo.