On the Design and Evaluation of a Precise Scalable Fiducial Marker Framework

Vilar Fiuza da Camara Neto
Núcleo de Estudo e Pesquisa em Computação
Fund. Centro de Análise, Pesq. e Inov. Tecnológica (FUCAPI)
Manaus, AM, Brazil
Email: neto@dcc.ufmg.br

Daniel Balbino de Mesquita
Renato Florentino Garcia
Mario Fernando Montenegro Campos
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG, Brazil
Email: {balbino,renatofg,mario}@dcc.ufmg.br

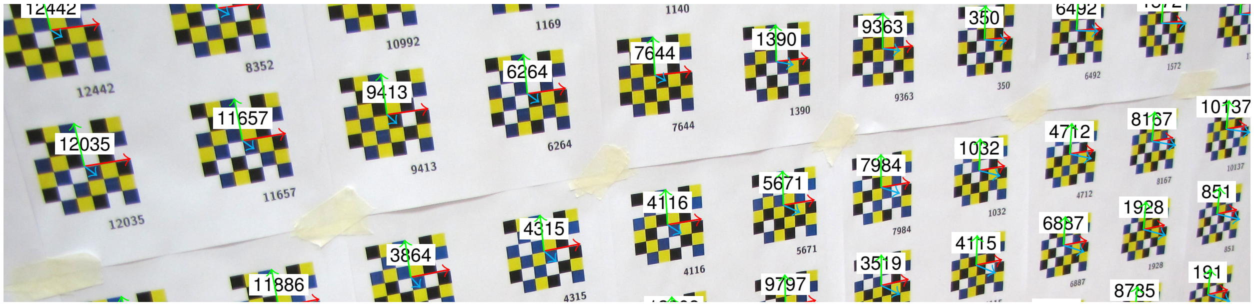

Figure 1. Overview of our fiducial marker framework: Each marker is a checkerboard with color cells that provides accurate three-dimensional pose estimation and tens of thousands of distinct markers.

***Abstract*—In this paper we present a planar fiducial marker system to be used with color cameras. Our system provides precise and robust full 3D pose estimation of the markers with superior accuracy when compared with many fiducial systems, while color information is used to provide more than 65,000 distinct markers. In contrast with most color-based fiducial frameworks, ours requires no prior classification training nor color calibration other than a rough white balance adjustment and can perform reliably under illumination changes. Finally, we also provide means of detecting when poor illumination conditions will compromise marker identification, thus avoiding to evaluate a false marker identification. We present several experiments that show significant improvement in accuracy of estimation of both position and orientation when compared with traditional techniques.**

***Keywords*-fiducial markers; localization; pose estimation; augmented reality.**

I. Introduction

Visual or *fiducial markers* are planar patterns designed to be easily detected when imaged by cameras and whose pose (either two- or three-dimensional) can be readily estimated with respect to the camera. The development of fiducial marker systems is commonly associated to Augmented Reality (AR) applications, but they have been used in several other applications, such as robot localization, medical, television, gaming, industrial, video conferencing, and human-machine interaction in general. Fiducial markers also have an important role in the study of robot swarms, where a large number of robots (possibly in order of tens of thousands) need to be individually tracked using camera systems.

It is often desired that fiducial marker tracking systems present at least the following properties [1]: *usability* (the system must be easily adaptable to several hardware and software platforms), *efficiency* (it must be fast and run at frame-rate speed), *accuracy* (identification must be exact and pose estimation must be precise), and *reliability* (the marker should be correctly tracked under significant perspective distortions and/or illumination changes). Some scenarios also require that a large number of markers (in order of hundreds or thousands) be uniquely identified.

To estimate the full 3D pose of a visual marker, at least 3 salient features must be identified and located in the image space (4 if perspective distortion is expected in the image [2]). The majority of fiducial systems [1], [3]–[7] was designed to be presented with a marker having a thick square border whose corners can be easily localized, thus providing a minimum number of features to recover pose, up to a 90-degrees ambiguity of the orientation. Features inside the square provide marker identification and orientation resolution. However, using only 4 features to track the marker also means that pose estimation is obtained with restricted precision, and can be severely affected by noise. Also, perimeter analysis of a dark object against a light background suffers from the unwanted effect if the illumination is abundant: light areas "bleed" through dark ones, which

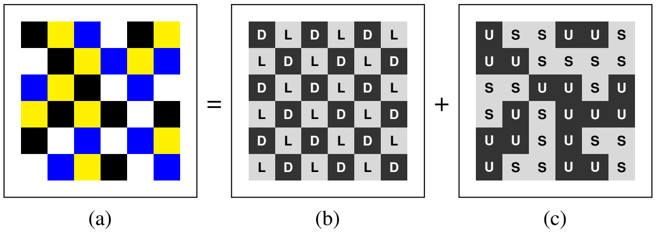

Figure 2. Proposed fiducial marker: (a) The marker is a checkerboard of dark (black or blue) and light (white or yellow) cells. The marker is composed by: (b) a checkerboard in the luminance channel and (c) several bits in the saturation channel. The meaning of the bits in the saturation channel is explained in the text. Luminance levels are depicted as: **D**=dark, **L**=light. Saturation levels are depicted as: **U**=unsaturated (black or white), **S**=saturated (colored).

causes shifting corner/edge position towards the interior of the perimeter. This way, the resulting estimated pose is farther from the camera than it should be.

Use of color in fiducial markers have already been investigated in the last years. Passive markers are generally composed of circle or square colored blobs and have been proeminently used for robot localization in swarms and robotic team competitions [8], [9]. However, in general these approaches give a limited precision pose recovery (since center of blobs are used as reference points) and require careful radiometric calibration in order to correctly classify pixel colors in images. On the other hand, some effort has been put into development of active markers based on color or infrared light emitting diodes (LEDs) [10]–[12], based on the premise that active markers suffer less influence from ambient illumination conditions. Marker identification is usually provided by recognition of temporal blinking patterns. However, these approaches present several drawbacks — for instance, they require power and embedded electronics, they can interfere with other sensors used in the scene, and marker identification cannot be carried with a single frame. Also, several frameworks presented in literature do not provide full 3D pose.

In this paper we propose a novel fiducial marker framework that overcomes the aforementioned limitations (Figure 1). It is even more precise than several other systems that use no color information, yet it does not require prior color calibration other than a rough white balance adjustment. Color is used to provide more than 65,000 unique marker identification and to detect poor illumination conditions that can degrade marker recognition.

The rest of this paper is organized as follows: Section II presents the proposed methodology for the 6×6-sized marker. Size generalization accounts for scalability and is discussed in Section III. The methodology is validated by several experiments whose results are shown in Section IV. Finally, Section V concludes with a discussion of the results and further research directions.

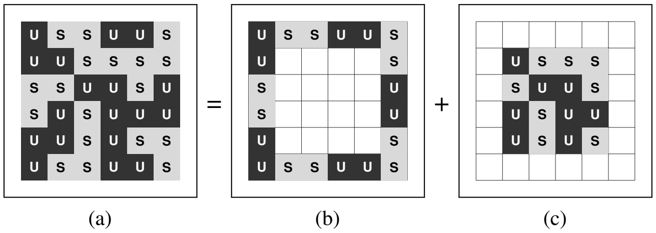

Figure 3. Composition of the saturation channel: (a) The saturation pattern is a combination of (b) a (constant) border that identifies the marker orientation and (c) a (variable) kernel that uniquely tags the marker.

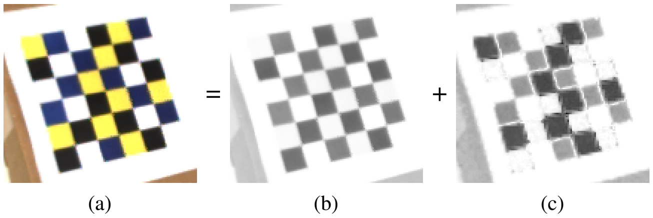

Figure 4. Channel separation of a real image, using CIE $L^*a^*b^*$ color space: (a) color image captured by the camera, (b) luminance (L^*) channel, (c) saturation ($|b^*|$) channel. For visualization purposes, the saturation image is shown in reversed scale (darker = more saturated).

II. Methodology

The proposed fiducial marker is depicted in Figure 2a. The marker is a checkerboard of dark (either black or blue) and light (either white or yellow) square cells. These colors were intentionally selected because an image containing instances of our markers can be easily decomposed in two main channels:

- the *luminance channel* (Figure 2b), which displaces the checkerboard in dark and light tones; and
- the *saturation channel* (Figure 2c), which tags the cells as either colored (highly saturated) or uncolored (unsaturated).

Channel separation can be carried out using several methods. In our experience, the most stable results were obtained by converting the image into the CIE $L^*a^*b^*$ color space and using the L^* band as luminance and $|b^*|$ as saturation.[1] A real image and corresponding channels can be seen in Figure 4. If speed is of serious concern, a faster (but less stable) approach is to convert the chromatic bands of the image into the HSV color space and use the V band as luminance and S as saturation.

The key advantage of the checkerboard pattern (luminance channel) is that its pose can be estimated with great accuracy. Pose estimation of a checkerboard is usually carried by finding the inner corners. A 6×6 checkerboard has 25 of such features, thus providing plenty of information redundancy. Pose estimation will probably undergo unaffected in presence

[1] The a^* band is useless with the chosen color set, since its value is almost the same for blue and yellow colors and grayscale tones.

of poor, noisy localization of a few corners (or even the failure to recognize one or two). Checkerboard identification and pose estimation is a well-studied subject and has been implemented in several freely-available libraries and toolboxes [13], [14].

The saturation channel is further composed by two pieces:

- the *orientation border* (Figure 3b), a fixed, 1-cell wide border with the same size of the checkerboard pattern; and
- the *identity kernel* (Figure 3c), a 4×4 cell pattern.

The orientation border pattern (alternating pairs of saturated and unsaturated cells) is the same for all fiducial markers. The pattern was designed with two properties in mind:

1) to provide a fast and stable way of overcoming the 180-degree orientation ambiguity of the checkerboard; and
2) since there are 5 cells of each color, it also provides a way of evaluating the distinctiveness of the information in the two aforementioned channels. If the high/low levels or either luminance or saturation channels becomes hard to distinguish, then illumination conditions are regarded as poor and a warning can be issued to the user or application. Analysis of channel levels is discussed in details in Subsection II-B.

The 4×4 identity kernel provides 16 bits of information, thus allowing a theoretical limit of $2^{16} = 65{,}536$ distinct markers. To ease the color classification and to improve the reliability of the marker identification, some constraints over the kernel pattern may be enforced in order to provide error detection and possibly error correction. Obviously, these constraints come at the expense of reducing the number of distinct markers. Such considerations will be further detailed in Subsection II-C.

A. Coordinates for color sampling

Checkerboard detector algorithms usually return only the coordinates of inner corners (red circle markers seen in Figure 5). Based on the intrinsic camera parameters, it is possible to recover the 3D pose of the marker — up to a $90n$-degree ambiguity about the axis normal to the marker plane at the center of the checkerboard. Correct orientation can only be fully recovered after analyzing the colors of the orientation border pattern. The marker identity is encoded in the color pattern of the identity kernel.

Colors are inferred by sampling the luminance and saturation channels around the center of each cell (Figure 5). Coordinates of each cell's center are evaluated according to its position in the checkerboard (refer to Figure 6):

- Centers of kernel (inner) cells (P_A) are evaluated as the mean of coordinates of surrounding corners:

$$P_\mathrm{A} = \frac{1}{4}(P_1 + P_2 + P_3 + P_4); \qquad (1)$$

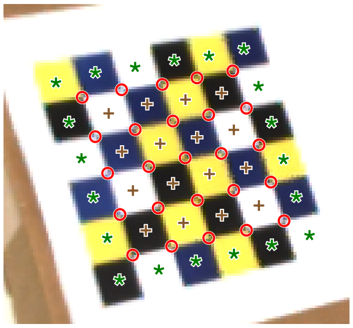

Figure 5. Interest points plotted over a real image: ○ are corners found by the checkerboard detector; + are the center of sampling regions used for kernel cell color analysis; and * are the center of sampling regions used for border cell color analysis.

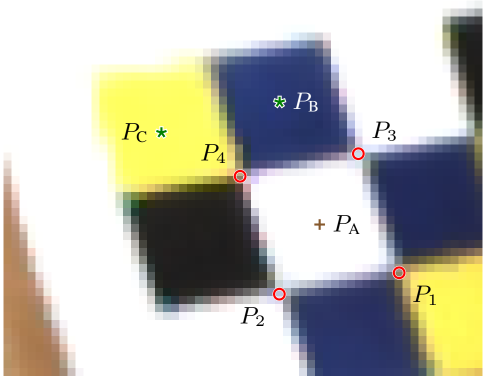

Figure 6. Conventions for explaining the evaluation of coordinates of sampling region centers. Refer to Subsection II-A for explanation.

- Centers of edge (non-corner) cells of the orientation borders (P_B) are evaluated by extrapolating coordinates of corners of the adjacent cell:

$$P_\mathrm{B} = \frac{1}{2}\left[P_3 + \frac{1}{2}(P_3 - P_1) + P_4 + \frac{1}{2}(P_4 - P_2)\right]; \qquad (2)$$

- Centers of outer corner cells (P_C) are evaluated by extrapolating coordinates of corners of the nearest inner cell:

$$P_\mathrm{C} = P_4 + \frac{1}{2}(P_4 - P_1). \qquad (3)$$

The pixels around each of these coordinates are sampled to recover the color of each cell. This procedure will be explained next.

B. Color classification

Color classification of all pixels are performed in three steps:

1) In the first step, only the border cells are analyzed to recover marker orientation (i.e., to overcome the $90n$-degree ambiguity);

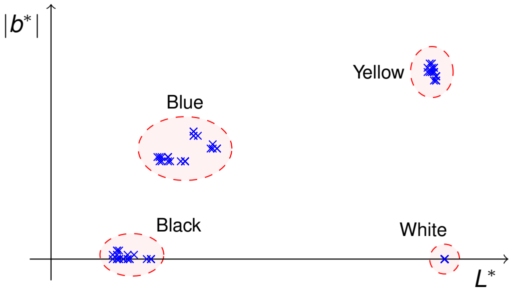

Figure 7. Example of values sampled from luminance and saturation channels.

2) Next, an analysis is performed to check if there is a significant probability of color misclassification (which indicates poor illumination conditions and/or inadequate camera white balance);
3) Finally, the saturation levels of inner (kernel) cells to extract the unique marker identification are checked.

Let us call $l_{r,c}$ and $s_{r,c}$ respectively the luminance and saturation of the cell located in r-th row and c-th column of the checkerboard, assuming an arbitrary selection of the top-left cell. For each cell, our current implementation evaluates the average luminance and saturation levels of the pixel in the center of each cell and the corresponding 4-connected neighbors. More complex methods could be used to provide more robustness against noise, but as it will be shown later, this simple approach provides good results.

While the cell colors are not yet known, the orientation border has a clever property: when read in clockwise order (see Figure 2a), cells present a cycle of "black, yellow, blue, white" colors. This means that the following cells have approximately the same luminance, regardless of actual marker orientation:

$$l_{1,1} \approx l_{1,5} \approx l_{4,6} \approx l_{6,4} \approx l_{5,1} \tag{4a}$$
$$l_{1,2} \approx l_{1,6} \approx l_{5,6} \approx l_{6,3} \approx l_{4,1} \tag{4b}$$
$$l_{1,3} \approx l_{2,6} \approx l_{6,6} \approx l_{6,2} \approx l_{3,1} \tag{4c}$$
$$l_{1,4} \approx l_{3,6} \approx l_{6,5} \approx l_{6,1} \approx l_{2,1}, \tag{4d}$$

and the same applies for saturation values. We evaluate the mean of each group's luminance. The two lower values correspond to darker (black and blue) colors and the remaining correspond to lighter (white and yellow) colors. Evaluation and comparison of the average value of saturation ultimately tags each cells with corresponding color, which eliminates the aforementioned orientation ambiguity. Figure 7 shows an example of color classification plotted in the color channels space.

Statistical analysis of sampled values also gives the classification quality. For each color we evaluate the Gaussian distribution over sampled values of luminance (Figure 8a). If the overlap between a darker and a lighter color is significant, then there is a high probability of color misclassification. This may be due to (i) over- or under-exposed images, (ii) too much noise or (iii) the marker being exposed to a highly uneven illumination. Same goes for Gaussian distribution curves of saturations (Figure 8b): in this case, curve overlaps may indicate poor white balance.

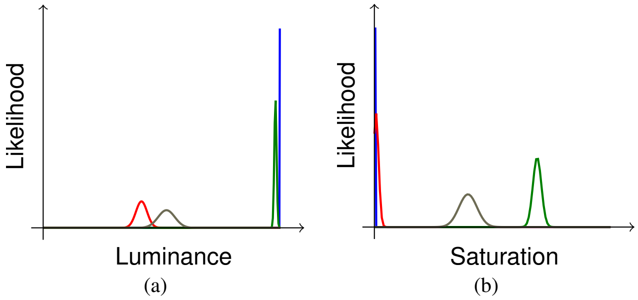

Figure 8. Example of Gaussian distribution curves over (a) luminance and (b) saturation of each border's cell groups, evaluated from image of Figure 4.

C. Marker identification

Saturation values of identity kernel cells ($s_{2,2}$ to $s_{5,5}$) provide a total of 16 bits of information, thus allowing $2^{16} = 65{,}536$ unique markers. We call this the *unconstrained variant* of our marker identification framework. The problem of this variant is that color misclassification of a single kernel cell will provide another valid identification: there is no way to check the validity of the acquired bit pattern or to identify and correct wrong bits.

There are several solutions to constrain the pattern in order to detect identification errors. Below we present three possible simple approaches:

1) In the *saturation-constrained variant*, exactly half of the cells are saturated, meaning that the kernel has 8 color cells (yellow/blue) and 8 black/white cells. This gives $C_8^{16} = 12{,}870$ distinct markers;
2) In the *color-constrained variant*, there are exactly 4 cells of each color, thus giving $C_4^8 \cdot C_4^8 = 4{,}900$ distinct markers;
3) In the *SECDED variant*, there are no specific constraints on the cell colors count. Instead, the pattern is regarded as a 16-bit sequence, where each bit represents a cell (1 iff the corresponding color is saturated). This pattern is coded using a Hamming(15,11) coding with an additional odd parity bit. Since there are 11 data bits, this variant provides $2^{11} = 2{,}048$ distinct markers.

One advantage of the first and second variants is that a single misclassification (actually, any odd number of misclassifications) will output an invalid identification. however, they do not provide error correction. On the other hand, the SECDED ("single error correction, double error detection") variant can detect a single- or double-bit error in 100% of the cases and provides correction for any single-bit error.

Table I
PROBABILITY OF INCORRECT MARKER IDENTIFICATION ACCEPTANCE IN FACE OF WRONG COLOR CLASSIFICATION.

	Misclassification amount			
Ident. variant	*1 bit*	*2 bits*	*3 bits*	*4 bits*
Unconstrained	100.0%	100.0%	100.0%	100.0%
Sat.-constrained	0.0%	53.3%	0.0%	43.1%
Color-constrained	0.0%	26.7%	0.0%	18.0%
SECDED	0.0%	0.0%	100.0%	7.7%

The odd parity bit guarantees that the completely saturated and completely unsaturated kernels (which may happen under complete saturation misclassification) are both invalid patterns.

Table I gives the probability of generating a valid but incorrect identification for each variant in face of different number of misclassifications. The end user must select a variant as a balance of the quality of available imaging process and the need of large number of distinct markers. Also, we stress that these are not the only possible solutions: the end user can adopt any other identification schema if these variants are somehow inappropriate for a given scenario.

III. SCALABILITY

Although we discuss in details the 6×6-sized marker, our methodology trivially scales to any $N \times N$-sized marker, where $N \geq 4$ and N is even. Under these constraints, the smaller possible marker has 4×4 cells, thus providing a 2×2 kernel with 16 distinct markers and no error detection/correction capabilities. This may be suitable in situations where a small marker is required and there is no need to place several markers in the scene.

On the opposite direction, bigger markers provide room for much more distinct and/or highly redundant coding schemes. For instance, an 8×8-sized marker can have its identification directly encoded as a binary number with no error detection/correction (which gives $2^{64} \approx 1.84 \times 10^{19}$ distinct markers — probably too much for most practical purposes), or it can be encoded with the extended binary Golay code, thus providing room for $2^{12} = 4{,}096$ distinct markers with automatic correction of up to 3 wrong bits and error detection of 4 wrong bits. However, bigger markers are practical only if high-resolution cameras are available.

Strictly speaking, there is no need to adopt square-sized, even-sized marker patterns. Actually the framework can be generalized to any $M \times N$-sized marker (with $M, N \geq 3$ in order to provide room for kernel cells), but the proposed "black, yellow, blue, white" border pattern requires the number of border cells to be multiple of 4. The generalization algorithm for the border patter is currently under development.

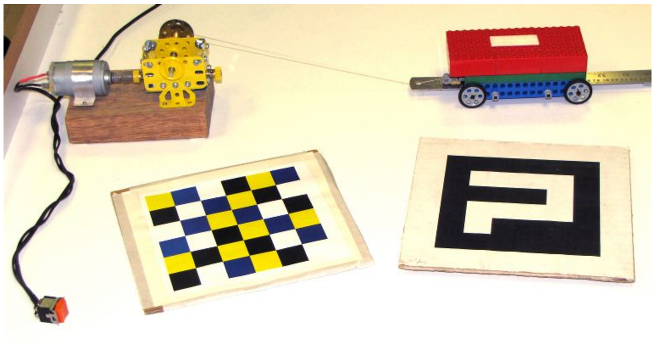

Figure 9. Markers and motion device used in comparative experiments.

Table II
TEST CASES: CAMERA POSES RELATIVE TO THE TABLE.

Test case	*Vertical distance to the table*	*Angle between camera axis and table plane*
CloseDist	60 cm	90°
MedDist	90 cm	90°
FarDist	140 cm	90°
Inc30d	90 cm	60°

IV. EXPERIMENTS AND RESULTS

The experiments were conducted in order to compare the precision and robustness of our fiducial marker framework against open-sourced *ARToolKitPlus* project [6], which is an improvement of the very popular *ARToolKit* [4].[2]

The experimental setup consists of a small moving wheeled base constrained to move in a straight track pulled by an electric motor in order to travel at fairly constant speed (Figure 9). Both markers are square and equally sized at $120\,\mathrm{mm}$ on a side. For checkerboard detection we have used the *OpenCV 2.0 library* [13] with some modifications to detect small checkerboards.

For each test run, one of the markers is fixed on the wheeled base. A 640×480 PointGrey DragonFly CCD camera with $f = 4\,\mathrm{mm}$ lens is fixed over the scene and captures the marker movement at $15\,\mathrm{fps}$. The wheeled base is then reset to the original position, where markers are exchanged and a new set of images is obtained. Poses of the camera relative to the table are described in Table II. We ran each experiment twice (each run is tagged with suffix "1" or "2" in the results discussion session). Each run captures approximately 280 to 400 frames.

From each run we recover the complete 3D pose of the marker being under analysis: three-dimensional coordinates of the fiducial marker center and its orientation described by a quaternion.Since the wheeled base (hence the marker) is

[2] Several variants of ARToolKit/ARToolKitPlus emerged in last years, however, in general, the pose estimation based on outer square corners has not changed. So, we have elected ARToolKitPlus as a representative of an entire class of fiducial markers systems.

traveling in a straight path, then (i) all 3D marker coordinates must be collinear and (ii) the orientation must not change throughout the run. From these constraints we can extract two noise measurements:

1) *Linear errors:* Given $\{P_f\}$, the set of 3D positions of the marker recovered at each frame f, the line that best describes the path followed by the marker (in terms of quadratic distance minimization) is defined by the line that crosses $\bar{P}$ and goes in the direction of the eigenvector correspondent to the maximum eigenvalue of the 3×3 matrix $\mathbf{P} \cdot \mathbf{P}^\top$, where $\mathbf{P}$ is defined as:

$$\mathbf{P} = \begin{bmatrix} P_1 - \bar{P} & P_2 - \bar{P} & \dots & P_F - \bar{P} \end{bmatrix}. \quad (5)$$

The *linear error* of a point P_f, called ε_f^P, is defined as the Euclidean distance between P_f and the best fit line;

2) *Angular errors:* One of the advantages of using quaternions to represent three-dimensional orientations is that it is easy to evaluate the rotation between two arbitrary quaternions q_a and q_b, i.e., to obtain a rotation axis $\vec{\mathbf{v}}$ and an angle α (where $0° \leq \alpha \leq 180°$) that describes the rotation that can be applied into the orientation q_a to reach q_b. Here we will adopt the notation $\alpha = \measuredangle(q_a, q_b)$ to denote the operation that evaluates the angle α (the rotation axis has no interest in this paper).
Given $\{q_f\}$, the set of quaternions that represent the orientation of the marker recovered at each frame f, we define the *best fit orientation* as a quaternion q_{fit} that minimizes the quadratic angular distance to all q_f, or:

$$q_{\text{fit}} = \arg\min_q \sum_f \left[\measuredangle(q_f, q)\right]^2. \quad (6)$$

The *angular error* of an orientation q_f, called ε_f^q, is simply defined as:

$$\varepsilon_f^q = \measuredangle(q_f, q_{\text{fit}}). \quad (7)$$

Both error measurements are always non-negative (zero under ideal circumstances).

For each test case we analyze two global error measurements: the *Root Mean Square Error*, RMSE(ε), defined as

$$\text{RMSE}(\varepsilon) = \frac{1}{F}\sqrt{\sum_f {\varepsilon_f}^2}, \quad (8)$$

and the *Mean Absolute Error*, MAE(ε), defined as

$$\text{MAE}(\varepsilon) = \frac{1}{F}\sum_f \varepsilon_f, \quad (9)$$

where ε can refer both to linear and angular errors. Smaller values obtained for RMSE(ε) and MAE(ε) indicate better results. RMSE is more sensitive to outliers than MAE.

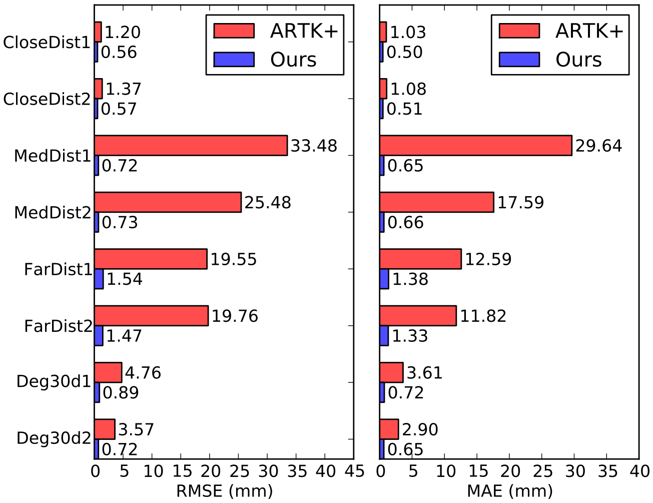

Figure 10. Linear error measurements for all described test cases and error measurements.

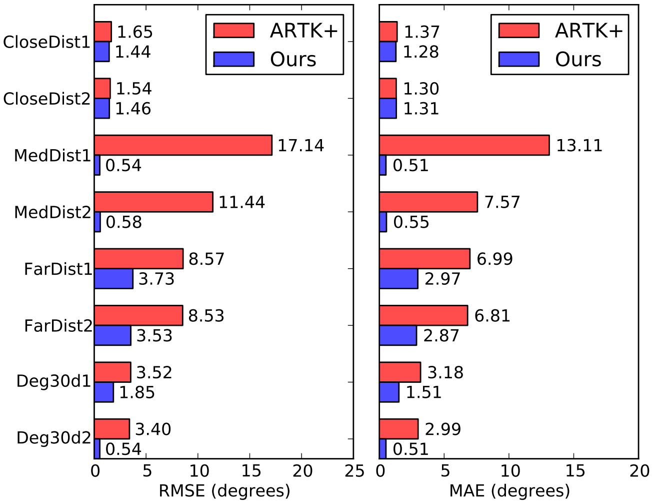

Figure 11. Angular error measurements for all described test cases and error measurements.

All global error measurements are presented in Figures 10 and 11. For almost all test cases, we have observed a noticeable improvement in the pose estimation precision. In general, RMSE and MAE values decreased significantly when our method was used when compared with ARToolKitPlus, for both linear and angular measurements. Also, notice that linear errors of about $1\,\mathrm{mm}$ were measured for objects placed about $1\,\mathrm{m}$ far from a low-resolution camera. These results indicate that our methodology may be used in applications that require high-precision estimation of marker positions.

Table III shows the marker recognition rate, i.e., the percentage of the frames where a visible marker was actually recognized. In this specific test, ARToolKitPlus performs slightly better than our method. This is due to the fact that the ARToolKitPlus framework needs to recognize only 4

Table III
MARKER RECOGNITION RATE.

Test case	*ARTK+*	*Ours*
CloseDist1	100.0%	98.0%
CloseDist2	100.0%	98.6%
MedDist1	100.0%	100.0%
MedDist2	100.0%	99.7%
FarDist1	100.0%	97.7%
FarDist2	100.0%	98.1%
Inc30d1	100.0%	98.4%
Inc30d2	100.0%	100.0%

visual features (the square borders), while ours requires up to 25 features (the checkerboard inner corners). However, as explained in Section II, we stress that our current implementation rejects the marker if a single checkerboard corner is not recognized. This restriction can be relaxed to accept a small number of misses. Nevertheless, it is worth noting that recognition rates approach 100% even in worst cases.

V. CONCLUSION AND FUTURE WORK

We have presented a novel fiducial marker framework that provides full 3D pose recovery. Compared to current methods, our approach provides significantly better pose estimation accuracy, as evidenced by the considerable decrease in both RMSE and MAE error measurements. Color information is used to provide tens of thousands of distinct markers (as for the 6 × 6-sized markers), which is enough for most applications in several interest areas. We also provide identification variants embedded with redundant information in order to provide confident identification and optional error correction. On top of that, our framework is also robust in the sense that it detects poor illumination conditions, inadequate white balance and other conditions that would otherwise lead to invalid results.

Although our method is generally more precise than traditional alternatives, working with color imaging poses some drawbacks. Factors such as image/video compression and color aberration can impair the color classification step, thus compromising orientation resolution and marker estimation. Also, since our algorithm relies on the identification of several image features (compared to only 4 needed by square-border methods), it is expected that marker recognition rates are lower when compared to those of traditional techniques, specially when used with low-resolution cameras. In general, one must consider the tradeoff between pose accuracy and recognition rates when selecting between our framework and other approaches.

Our current implementation is highly unoptimized and must be further developed in order to be evaluated in terms of efficiency. Also, pose estimation in face of misrecognition of a few checkerboard corners is currently under evaluation. This requires careful study to estimate the loss of precision caused by progressive corner misses. This is being currently addressed by the authors, as part of ongoing researches in swarm robotics and augmented reality.

ACKNOWLEDGMENTS

This work was partially supported by *Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (Analysis, Research and Technological Innovation Center) — FUCAPI*, Manaus, AM, Brazil; *Conselho Nacional de Desenvolvimento Científico e Tecnológico (National Counsel of Technological and Scientific Development) — CNPq*, Brazil; and *Fundação de Amparo à Pesquisa do Estado de Minas Gerais — FAPEMIG*, MG, Brazil.

REFERENCES

[1] X. Zhang, S. Fronz, and N. Navab, “Visual marker detection and decoding in AR systems: A comparative study,” in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, 2002, pp. 97–106.

[2] M. Fiala, “Designing highly reliable fiducial markers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, Jul. 2009.

[3] J. Rekimoto, “Matrix: A realtime object identification and registration method for augmented reality,” in *Proceedings of the 3rd Asia Pacific Computer Human Interaction (APCHI ’98)*, 1998, pp. 63–68.

[4] H. Kato and M. Billinghurst, “Marker tracking and HMD calibration for a video-based augmented reality conferencing system,” in *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*, San Francisco, USA., Oct. 1999.

[5] M. Fiala, “ARTag, a fiducial marker system using digital tecnhiques,” in *Proceedings of the Computer Vision and Pattern Recognition*, Jun. 2005, pp. 590–596.

[6] D. Wagner and D. Schmalstieg, “ARToolKitPlus for pose tracking on mobile devices,” in *Proceedings of 12th Computer Vision Winter Workshop (CVWW’07)*, Feb. 2007.

[7] D. Flohr and J. Fischer, “A lightweight ID-based extension for marker tracking systems,” in *Eurographics Symposium on Virtual Environments (EGVE) Short Paper Proceedings*, 2007, pp. 59–64.

[8] J. Bruce and M. Veloso, “Fast and accurate vision-based pattern detection and identification,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, Sep. 2003.

[9] H. Zhang, F. Yang, and Y. Wu, “Robust color circle-marker detection algorithm based on color information and Hough transformation,” *Optical Engineering*, vol. 48, no. 10, 2009.

[10] Y. Hada and K. Takase, “Multiple mobile robot navigation using the indoor global positioning system (iGPS),” in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Maui, HI, Nov. 2001, pp. 1005–1010.

[11] N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, J. Rekimoto, and Y. Yamamoto, “ID CAM: A smart camera for scene capturing and ID recognition,” in *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA: IEEE Computer Society, Oct. 2003, pp. 227–236.

[12] M. Sakata, Y. Yasumuro, M. Imura, Y. Manabe, and K. Chihara, “ALTAIR: Automatic location tracking system using active IR-tag,” in *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2003)*, Jul. 2003, pp. 299–304.

[13] “Open Source Computer Vision library,” available at http://sourceforge.net/projects/opencvlibrary/. Accessed in May, 2010.

[14] J.-Y. Bouguet, “Camera calibration toolbox for Matlab,” available at http://www.vision.caltech.edu/bouguetj/calib_doc. Accessed in May, 2010.