Decoding positional and color information from a coded pattern

Nelson L. Chang, Suk Hwan Lim, and Feng Tang

HP Laboratories
HPL-2010-131

Keyword(s):
single shot coded pattern, camera calibration, geometric calibration, color consistency, augmented reality

Abstract:
The paper presents decoding positional and color information using visual coded patterns for efficient geometric calibration and color consistency across multiple cameras. The patterns are generated from an alphabet of basis colors placed in unique spatial configurations called ChromaCodes. The imaged patterns are automatically decoded to derive the information needed for both geometric and color calibration. Previous 2-D structured patterns are typically designed for obtaining only geometry whereas the proposed patterns capture color information as well. Moreover, the unique decodability of the codes allows them to overcome the problem of pattern visibility in all views and enables effective geometric calibration and color consistency with even partial and/or occluded view of the pattern. Experimental results demonstrate that a single shot of the pattern is sufficient to encapsulate enough information to compute both geometry and color across one or more cameras, especially important for realtime or interactive applications.

External Posting Date: September 27, 2010 [Fulltext] Approved for External Publication

Internal Posting Date: September 27, 2010 [Fulltext]

Published and presented at the IEEE Internatioanl Conference on Image Processing, September 26-29 2010, Hong Kong.

DECODING POSITIONAL AND COLOR INFORMATION FROM A CODED PATTERN

Nelson L. Chang, Suk Hwan Lim, and Feng Tang

Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA
{ nelson.chang | suk-hwan.lim | feng.tang } @ hp.com

ABSTRACT

The paper presents decoding positional and color information using visual coded patterns for efficient geometric calibration and color consistency across multiple cameras. The patterns are generated from an alphabet of basis colors placed in unique spatial configurations called ChromaCodes. The imaged patterns are automatically decoded to derive the information needed for both geometric and color calibration. Previous 2-D structured patterns are typically designed for obtaining only geometry whereas the proposed patterns capture color information as well. Moreover, the unique decodability of the codes allows them to overcome the problem of pattern visibility in all views and enables effective geometric calibration and color consistency with even partial and/or occluded view of the pattern. Experimental results demonstrate that a single shot of the pattern is sufficient to encapsulate enough information to compute both geometry and color across one or more cameras, especially important for real-time or interactive applications.

Index Terms — single shot coded pattern, camera calibration, geometric calibration, color consistency, augmented reality

1. INTRODUCTION

It is common to use known reference patterns for many camera-based applications. For geometric calibration of cameras, such a pattern enables solving for and properly modeling various camera non-idealities such as lens distortion as well as the relative position and orientation of the camera with respect to some world origin. Typically, a pattern is comprised of robust, easy to detect features (e.g. the intersections of a black/white checkerboard pattern or corners of a multi-box pattern) that refer to specific world coordinates and are used to solve for the intrinsic and extrinsic camera parameters [2, 21]. Because the patterns do not explicitly encode positional information with respect to the pattern, they tend to be particularly sensitive to occlusions, incomplete visibility (i.e. not completely in view), or other violations of spatial coherence. Moreover, with cameras in general position with very wide baselines, it is likely that the reference pattern will not be completely visible in all cameras simultaneously.

For color calibration, a reference chart (e.g., the Macbeth chart) helps in the estimation of the color transformation between the color space of the reference to that of the camera(s), often to ensure color consistency across devices. Currently, it is necessary for someone to manually identify the squares of the chart. While hypothesis testing may be used to locate each of the squares, issues arise when the chart is partially obstructed or out of view. Also, while white balancing could be performed by measuring pixel values in a black/white checkerboard pattern, only a simple color model (i.e. gain and offset for each color) can be estimated. More complex color models require more colors in the patterns.

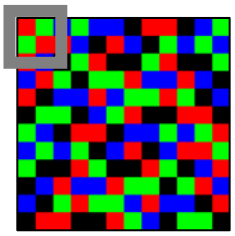


Figure 1. Example 12x12 pattern consisting of unique overlapping 2x2 ChromaCodes made up of four basis colors (red (R), green (G), blue (B), black (K)). 1-D sequence {R,G,R,B,R,K,G,B,G,K,B,K} (the leftmost column) was used to generate the pattern through successive column shifts.

In this paper, we present coded patterns that simultaneously encode the positional and color information with respect to some reference coordinate system. The patterns are constructed from *uniquely* decodable color codes referred to as *ChromaCodes*. When imaged by one or more cameras, a single shot of such pattern can be automatically analyzed to establish the geometric mapping and color transformation between the camera and reference coordinate systems simultaneously. Moreover, it is robust to incomplete visibility and occlusions, thereby reducing reliance on spatial coherence. Calibration using ChromaCoded patterns is very useful when the camera pose and/or illumination change continuously (e.g. in applications such as augmented reality), thereby requiring repeated automatic calibration. It is especially important when one wants to measure the color and/or illumination at precisely the same location in the scene.

2. CHROMACODED PATTERNS

The proposed patterns are formed from a set of C basis colors, configured in R x S overlapping codes referred collectively as ChromaCodes. The codes directly encode both positional and local color information with respect to a reference coordinate system. By design, every one is unique and quickly decodable, with each code appearing at most once in a single pattern.

One of the primary design parameters is the number of basis colors. As the number of C basis colors increases, so do the number of uniquely decodable codes and hence the effective resolution of the overall pattern. However, this increase comes at the cost of decreased discriminability and decoding robustness, especially for uncalibrated cameras. Thus, the number of basis colors governs the ability of the camera to reliably distinguish among the primaries, leading to a tradeoff between accuracy and maximum pattern resolution.

In a similar fashion, varying the size of the RxS spatial configuration leads to a tradeoff between effective spatial resolution and robust decoding in the presence of occlusions and surface deformations. At one extreme, the code of size $R=S=1$ is

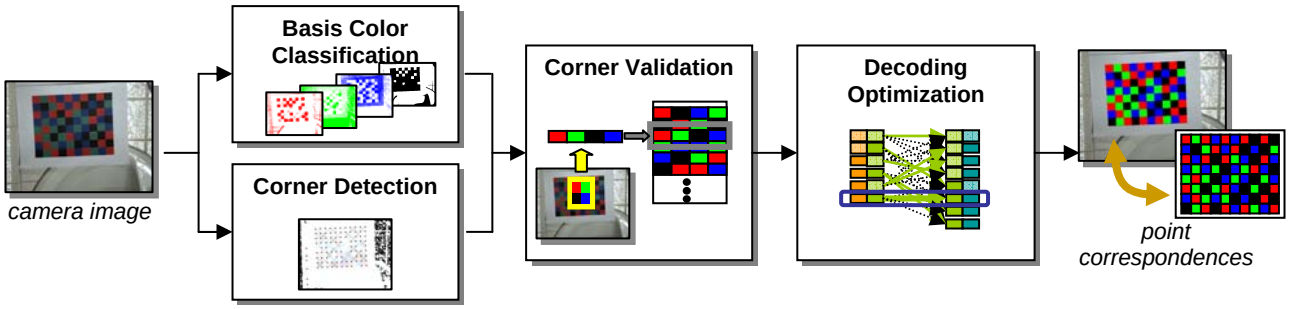


Figure 2. Analysis and decoding workflow

the least affected by pattern deformations, but the pattern resolution would be constrained to be equal to the number of basis colors and lead to many false positives. At the other extreme, large code dimensions leads to a bigger dictionary of uniquely decodable codes, each less likely to occur naturally in the scene. However, this advantage comes with increased susceptibility to occlusions, surface deformations, and spatial coherence.

As described, each code in a *RxS* configuration can take on any of *C* colors, leading to a maximum of C^{RS} possible ChromaCodes. One can view the ChromaCoded patterns more generally as 1-orientable aperiodic *C*-ary arrays [3], where each valid code appears at most once in a given pattern. Other patterns also include the general non-binary perfect and semi-perfect maps [11]. Here, a set of *C* basis symbols is considered to create a wraparound pattern with unique *R*x*S* configurations, enumerating every, or almost every, possible configuration. For odd number of basis colors *C*, there exists a simple pattern construction by taking a 1-D *C*-ary de Bruijn sequence and replicating shifted versions of the sequence to form a $(C^R,C^S;R,S)$ non-binary perfect map [7].

For many practical applications, the exact boundaries of the ChromaCodes will not be known a priori, and the imaged pattern cannot be assumed to be fronto-parallel and rectangular in shape. With patterns such as perfect maps, every possible code appears exactly once (including wraparound), resulting in configurations with adjacent cells of the same basis color. These configurations are more difficult to localize and should be avoided. Thus, we prefer to have codes whose adjacent cells differ in color and thus have as many discernable color edges as possible.

Since we require only a unique set of codes within the pattern and do not need properties like periodicity or perfectness, we adopt a similar strategy as above to maximize the adjacent cell edges for arbitrary *C* (not just odd). Instead of using a de Bruijn sequence, we start with a 1-D sequence formed by all *C(C-1)* pairs of basis colors; this starting sequence is equivalent to a universal cycle of 2-permutations from the set of *C* colors [5]. The initial sequence fills the leftmost column (0^{th} column) of the 2-D pattern, and each subsequent column is replicated from the previous one, shifted up (and wrapped around if necessary) by the column number. The underlying reference coordinate is specified at the center of each code, such that any adjacent *R*x*S* grouping results in a provably unique configuration and with at most two adjacent cells having the same basis color in a given code. We make the following observations about this construction:

Observation 1: The construction guarantees unique overlapping RxS codes for 2≤R≤C(C-1) and 2≤S≤C(C-1).

Observation 2: The construction guarantees that at least three of the four interior color edges in any 2x2 subarray will be discernable.

Observation 3: Wraparound codes along the boundaries are not unique.

Observation 1 comes from noticing that the left two cells of any 2x2 subarray in the leftmost column will be matched with each pair from the universal cycle in succession, thus leading to uniqueness for any 2x2. With each 2x2 subarray unique, it follows that a larger subarray will also be unique. If we label the inter-cell boundaries as N,E,S,W, Observation 2 comes from the fact that a) there are no repeats in a given column (guaranteeing W and E color edges), and b) no two adjacent columns are identical (ensuring the N and/or S color edge). Observation 3 follows from the fact that every neighboring pair of the universal cycle is paired with every other one within the interior of the pattern.

We thus have a simple pattern construction to form a *C(C-1)xC(C-1)* pattern (*J=I=C(C-1)*) out of the total C^{RS} possible codes that maximizes the number of discernable inter-cell color edges. Smaller patterns are obtained by cropping a subwindow of the full pattern. Figure 1 shows an example 12x12 pattern of overlapping 2x2 codes using four basis colors (red (R), green (G), blue (B), and black (K)) generated by the 1-D sequence *{R,G,R,B,R,K,G,B,G,K,B,K}* (i.e. *C=4* and *R=S=2)*. The result is a 11x11 rectangular array such that any adjacent 2x2 grouping results in a unique configuration and with at most two adjacent cells having the same basis color in a given code.

The ChromaCoded patterns bears similarity to other patterns derived from de Bruijn sequences. Much of the previous work has focused on using these as structured light patterns to solve the correspondence problem and recover 3-D shape [1]. "One-shot" techniques fall predominantly into two camps: The first camp uses 1-D colored stripes generated by de Bruijn sequences for dynamic 3-D shape recovery; for example, Zhang et al. [20] use a 5-ary de Bruijn sequence of order 3 to encode the 1-D planes from a calibrated set up. 3-D shape is then obtained by solving for ray-plane intersections.

The second camp constructs 2-D patterns from distinct colored dots based on perfect maps or M-arrays [14]. For example, Morano et al. [12] created pseudo-random color patterns that maximizes hamming distance using a brute force technique. Claes and Bruyninckx [6] construct a 6-ary 3x3 de Bruijn pattern for visual servoing a robotic arm. With a slightly different take, Salvi et al. [13] used equally spaced horizontal and vertical slits using 3-ary de Bruijn sequences of order 3 for each orientation to encode geometric information at their intersections.

In contrast, the proposed ChromaCodes are formed by a grid of adjacent cells, without requiring equal spacing and enabling reasonable code density in a single pattern for a given spatial resolution. ChromaCodes and their code design are more resilient to spatial deformations and occlusions. In addition to encoding positional information, the colors of the cells may also be used for estimating local color information of the scene. The codes are designed to easily detected, and the pattern leads to a simple construction method for any *C*, *R*, *S*.

3. PATTERN ANALYSIS AND DECODING

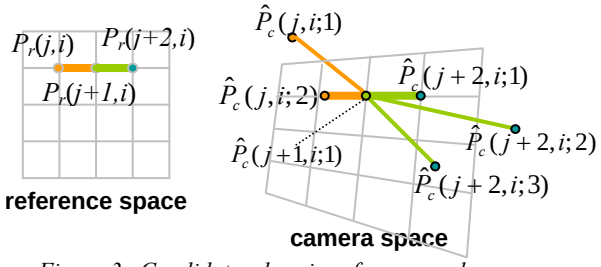


Figure 3. Candidate edges in reference and camera spaces for the dynamic programming framework.

By design, the ChromaCoded patterns encapsulate the correspondence mapping, as well as the appropriate color transformation, between a camera space and the reference space. When captured by one or more cameras, these patterns may be quickly decoded to robustly and accurately identify the camera space points and hence locate the ChromaCoded pattern in a single pass. As shown in Figure 2, the entire workflow for the decoding process for each camera image consists of corner detection, basis color classification, corner validation, and decoding optimization. In order to ensure that the pattern is always detected, the decoding process starts off with a simple detector/classifier such that there are many candidate points for the pattern. Each subsequent step imposes a different prior knowledge about the pattern and uses it to filter out the candidate points.

3.1. Corner detection

The first step is to detect salient corners in the image. A color-based extension of the Harris corner detector [19] is used to more robustly locate all relevant corners in the image. Non-maximal suppression and local clustered filtering are performed to further improve robustness. Only the top 10% of candidate corners are retained. At this stage, it is more important to capture all relevant corners even if there are a large number of false positives.

3.2. Basis color classification

Every camera image pixel is classified to one or more of the basis colors. To support uncalibrated cameras with possibly arbitrary color shifts or spectral response, we opt for three parallel color classifiers where the false negatives for each classifier may be minimized. Each camera pixel is classified to be at most three colors, one from each complementary set {(red (R), green (G), blue (B)), (black (K), white (W)), (cyan (C), magenta (M), yellow(Y))}. Dividing the symbols into complementary sets allows us to have many basis colors while minimizing the false negatives during this classification stage. Determining which complementary set the pixel belongs to would be done in a later stage (decoding optimization) with the use of spatial structure of the ChromaCoded pattern. Additional classifiers may be added when using more basis colors.

3.3. Corner validation

The list of points returned by the corner detector may consist of many false positives and thus needs to be pruned. We combine the results of the previous two stages to validate the corners with respect to the pattern. Since a valid corner point should lie at the intersection of four basis colors, we analyze a 3x3 patch with respect to the corner point and determine whether the majority of each quadrant consists of a single basis color. Corner points that have four valid quadrants are compared to the list of approved ChromaCodes for further pruning. This step leads to a list of candidate corner points that correspond to a possibly valid code in the reference pattern. These camera space corner points are then added as the possible candidates for their corresponding ChromaCode coordinate.

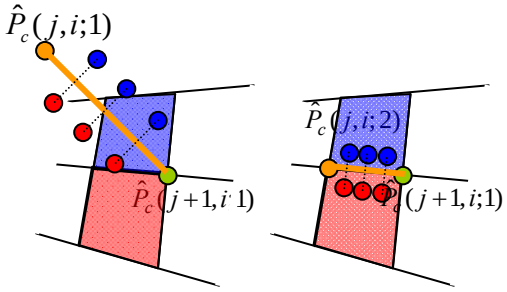


Figure 4. Computing likelihood of being the correct color edge.

3.4. Decoding optimization

The final step is to prune the existing corners and identify actual ChromaCodes in the scene. Each ChromaCode coordinate may be assigned possibly many candidate camera space corner points. At most only one such corner point can actually correspond to the ChromaCode coordinate. As such, this step needs to be robust to possibly spurious points that happen to decode to a valid ChromaCode as well as handle possible occlusions and non-planar deformations of the pattern.

We employ a dynamic program along each row to leverage the spatial relationship of the entire pattern and optimize the pruning and identifying process. The candidate points between adjacent ChromaCode coordinates are examined. An additional candidate placeholder is added to signify an occluded point. Instead of optimizing point positions directly, we enforce constraints with the edge segments to better exploit edge connectivity along a given row in the pattern. Edge nodes are formed between every combination of candidate points in adjacent coordinates, including the "occluded" placeholder.

We notate the underlying *JxI* reference grid coordinates as $P_r(j,i)=(x_{ji},y_{ji})$, indexed by $j=1\ldots J,\ i=1\ldots I$. Similarly, we define the corresponding points in the camera space as $P_c(j,i)=(u_{ji},v_{ji})$ (Figure 3). Also, denote $E_r(j,i)$ as the edge segment connecting reference points $P_r(j,i)$ and $P_r(j+1,i)$, i.e. $E_r(j,i)=\{P_r(j,i),P_r(j+1,i)\}$. Likewise, denote $E_c(j,i)=\{P_c(j,i),P_c(j+1,i)\}$ be the corresponding edge segment in the camera space. We define $E_c(j,i;m,n)$ as the edge segment connecting candidate camera points $\hat{P}_c(j,i;m)$ and $\hat{P}_c(j+1,i;n)$, i.e. $E_c(j,i;m,n)=\left\{\hat{P}_c(j,i;m),\ \hat{P}_c(j+1,i;n)\right\}$. The segment length and orientation

$$d(j,i;m,n)=\left\|\hat{P}_c(j,i;m)-\hat{P}_c(j+1,i;n)\right\| \quad (1)$$

$$\theta(j,i;m,n)=\cos^{-1}\frac{\hat{P}_c(j,i;m)\bullet\hat{P}_c(j+1,i;n)}{\left\|\hat{P}_c(j,i;m)\right\|\left\|\hat{P}_c(j+1,i;n)\right\|} \quad (2)$$

are recorded. Edges with nearly zero lengths (i.e. $d(j,i;m,n)<\varepsilon$) are discarded. Each edge node is assigned an initial cost $C_o(j,i;m,n)$ based on the likelihood of forming an edge that borders the two basis colors; as seen in Figure 4, points are sampled just above and below the edge and compared to the expected basis colors.

The dynamic program finds the minimum cost solution through the candidate edges. The cost $C(j,i;m,n,o,p)$ of connecting edge

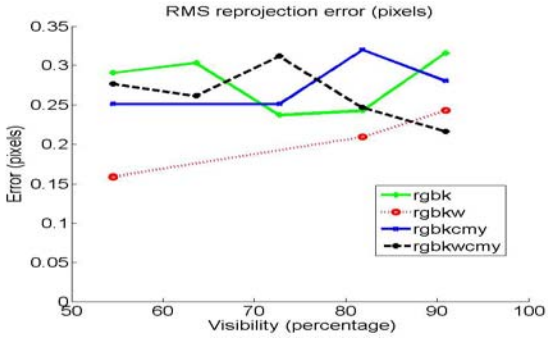


Figure 5. Camera calibration performance (in terms of RMS reprojection error) of different ChromaCoded patterns as a function of pattern visibility (percentage).

segment $E_c(j,i;m,n)$ and $E_c(j+1,i;o,p)$ is given by

$$C(j,i;m,n,o,p) = C_o(j,i;m,n) + C_{dist}\Delta d(j,i;m,n,o,p) + C_{angle}\Delta\theta(j,i;m,n,o,p) + C_{occl} \quad (3)$$

where $\Delta d(j,i;m,n,o,p) = \left\| d(j,i;m,n) - d(j+1,i;o,p) \right\|$ is the edge length difference, $C_{dist}\Delta d$ penalizes for large differences in edge lengths (the pattern edges are assumed to be roughly similar), $\Delta\theta(j,i;m,n,o,p) = \left\|\theta(j,i;m,n) - \theta(j+1,i;o,p)\right\|$ is the edge orientation difference, $C_{angle}\Delta\theta$ penalizes for large differences in orientation (the edges along a given row are assumed to be locally linear), and C_{occl} penalizes if selecting an occluded placeholder (is zero otherwise). To ensure edge connectivity, only $n=o$ cases are considered. Suppose $\vec{g}_{ji}$ represents the set of candidate edge segments for $E_c(j,i;m,n)$. A given solution path G through the data incurs a cost $T(G) = \sum_x C(x,i;\vec{g}_{xi})$. Thus, the dynamic program finds the path G that minimizes $T(G)$. Additional processing may be used to prune invalid candidate edges and discard outliers. In the end, we obtain the point correspondences between coordinates $P_r(j,i)$ in the ChromaCoded reference grid and their counterparts $P_c(j,i)$ in the camera image(s).

4. EXPERIMENTAL RESULTS

In this section, we describe various experiments using up to four heterogeneous cameras (labeled A to D) to demonstrate the effectiveness of the proposed ChromaCoded patterns. Without loss of generality, coded patterns with $R=S=2$ and varying number of basis colors are considered. In a first series of experiments, ChromaCodes may be used to automatically identify robust features for camera calibration. Since the focus of the approach is on the automatic detection and decoding of the pattern features and not on the actual calibration algorithm, the experiments assume a fixed calibration methodology (in this case, Bouguet's MATLAB Camera Calibration toolbox [2]). Reprojection error is used as a relative measure of the accuracy for the set of feature points.

Pattern\Camera	A		B		D	
RGBK	14.577	**2.88**	11.7453	**2.63**	17.6316	**8.84**
RGBKW	19.6786	**3.86**	26.9602	**6.96**	14.8948	**2.82**
RGBKCMY	19.0233	**3.95**	17.3272	**4.51**	14.617	**1.49**
RGBKWCMY	28.9973	**6.56**	23.3144	**6.40**	21.3142	**3.79**

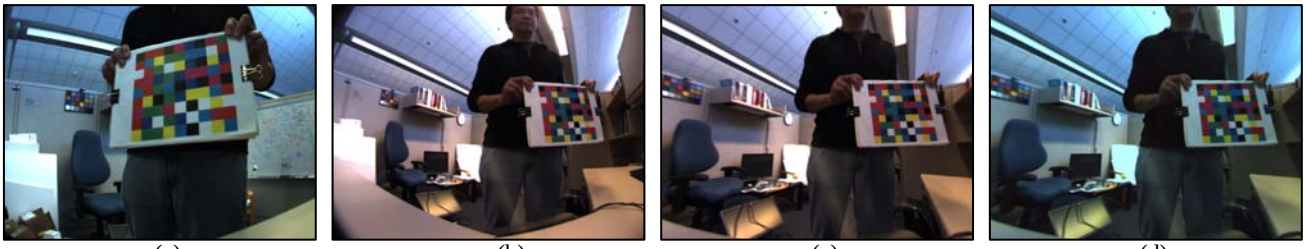
Table 1. RMS error (pixel) of the feature point colors before and after applying the estimated color transformation to camera C.

We compare the results for different 9x9 (cropped) patterns, wherein each cell of the pattern measures 22mm x 22mm and the printed pattern is matted onto a flat surface. In each case, the four heterogeneous cameras capture a total of six different views of each pattern. The feature points are automatically computed and serve as direct input for camera calibration. The features are then refined to subpixel accuracy based on initial results and the reprojection error is finally computed. Regardless of the number of frames used, the error is roughly around **0.25 pixels**, suggesting good calibration accuracy using any of the patterns. We find these results to be comparable in accuracy to manually selecting features on a similarly sized checkerboard pattern [17]. The error does not decrease with additional frames because we believe it falls within the accuracy of the subpixel interpolation.

A second series of experiments validate that ChromaCodes perform well in the presence of occlusions. To examine the performance as a function of visibility, an LCD monitor is used to display each of the four patterns, with a portion of the pattern obscured and the remaining points used for camera calibration. As seen in Figure 5, the overall reprojection error is somewhat higher but suggests good calibration performance even with close to 50% of the points obstructed.

In addition to geometric calibration, ChromaCodes can simultaneously improve the color consistency across multiple cameras. The proposed codes directly encapsulate the color information of the scene. Feature points are automatically decoded and used to solve for the intrinsics of each camera. We model the inter-camera color transformation as a 3x4 affine matrix, and solve using linear least squares with the center colors of each cell after gamma correction. Camera C is selected as the target color space and the other cameras are transformed to it. Table 1 illustrates the RMS component color pixel error before and after applying the estimated color transformation. Note the significant improvement in using the patterns for color consistency across all the frames.

Figure 6 illustrates some of the calibration and color results. Two representative frames from cameras C and D are shown in Figures 6(a) and (b); note the wide baseline and severe color



(a) (b) (c) (d)

Figure 6. Representative results for camera calibration and color consistency: Original images from (a) camera C and (b) camera D; (c) undistorted image after calibration; (d) resulting image after color correction.

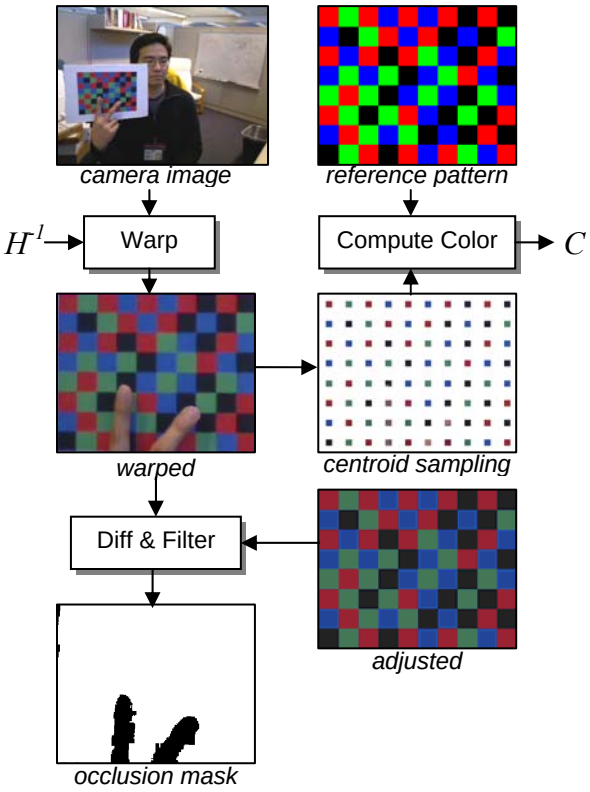


Figure 7. Framework for computing geometry and color transformations.

differences. Figure 6(c) shows the result of using the computed geometric parameters to undistort the image. Here, the barrel distortion from camera D has been automatically corrected. Figure 6(d) shows the color corrected result. Note how consistent the result is compared with the reference camera shown in Figure 6(a).

The proposed ChromaCoded patterns can also be used for automatic content insertion, especially important for real-time augmented reality applications. A coded pattern is used to define the region to be replaced, instead of segmenting objects through Chroma Keying [15] or computational matting [10, 16]. For each camera, the desired content undergoes a geometric and color transformation to bring the content to the camera's space (see Figure 7). By design, the pattern and its decoding are not restricted to non-planar surfaces. However, we assume a planar pattern to simplify real-time warping and solve for the homography using linear least squares with outlier rejection using the pairwise correspondences. Each measured camera color is related to its expected reference color by a 3x4 affine color transformation matrix, computed using linear least squares with the color of local neighborhoods around the centroids of each warped cell. Points that deviate significantly from the predicted color are masked off and are considered to be occluders, thereby creating an occlusion mask as a byproduct.

The proposed approach currently runs unoptimized at about 5-10 fps, thus approaching real-time video rates. Figure 8 shows an example for two cameras capturing a moving pattern. The top row shows the captured images. The next row demonstrates the results of texture mapping an input image using just the computed geometric mapping. The following row demonstrates incorporating the computed color transformation. Notice how well the two

Figure 8. Example of automatic content insertion: (a) original camera images; (b) applying only geometric warps; (c) correcting for color; (d) factoring in the occlusion mask.

inserted images blend with their respective camera's color space. The final row incorporates the computed occlusion mask to give a reasonable approximation of preserving depth ordering.

Figures 9 and 10 highlight the clear benefits of using the proposed ChromaCoded patterns. As shown in the top row of Figure 9, much of the pattern is oblique to the camera and offscreen in both cases. The unique pattern encoding allows robust recovery of the necessary geometry and color transformations to properly insert the content (bottom row of Figure 9). Similarly, Figure 10 demonstrates the approach's automatic responsiveness to global brightness changes. Unlike tracking frame corners [9] or conventional patterns for augmented reality [8], ChromaCodes does not require full visibility or planarity. In contrast to scene-dependent natural images [18], it is more resilient to occlusions and better suited for simultaneous color recovery with its uniformly distributed codes. Figure 11 shows additional frames from a video sequence using ChromaCoded patterns for seamless content insertion.

5. CONCLUSIONS

We have presented coded patterns designed to efficiently encode both geometric and color information. The result is the ability to use a single pattern to quickly solve geometric calibration as well as color consistency across multiple cameras. The advantage of

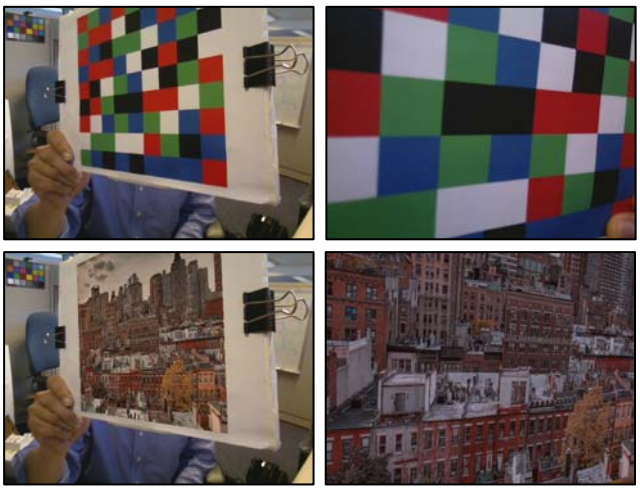


Figure 9. Automatically inserted content in the presence of oblique angles and partial pattern visibility.

using ChromaCodes comes from its simple construction, the addition of color information, as well as improved resilience to partial visibility and occlusions. Co-located positional and color information helps for interactive applications like augmented reality when the camera pose and/or illumination frequently change. We are currently investigating the use of ChromaCoded patterns for these and other applications.

REFERENCES

[1] J. Batlle, E. Mouaddib, J. Salvi, "Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem: A Survey," *Pattern Recognition*, vol. 31, no. 7, pp. 963—982, 1998.

[2] J.-Y. Bouguet, *Camera Calibration Toolbox for MATLAB*, June 2008.

[3] J. Burns and C. J. Mitchell, "Coding Schemes for Two-Dimensional Position Sensing," *HPL-92-19*, January 1992.

[4] Y. H. Chen et al., "A Vision-Based Augmented-Reality System for Multiuser Collaborative Environments," *TMM*, vol. 10, no. 4, pp. 585—595, 2008.

[5] F. Chung, P. Diaconis, R. Graham, "Universal Cycles for Combinatorial Structures," *Discrete Math.*, 110: 43—59, 1992.

[6] K. Claes and H. Bruyninckx, "Robot Positioning using Structured Light Patterns suitable for Self Calibration and 3D Tracking," *International Conference on Advanced Robotics*, Jeju, Korea, 2007.

[7] T. Etzion, "Constructions for Perfect Maps and Pseudo-random Arrays," *IEEE Transactions on Information Theory*, 34(5):1308—1316, 1988.

[8] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," *2nd Int'l Workshop on Augmented Reality,* 85--94, San Francisco, CA, Oct. 1999.

[9] M. C. Leung et al., "A Projector-based Hand-held Display System," *Proc. IEEE CVPR*, June 2009.

[10] A. Levin, A. Rav-Acha, D. Lischinski. "Spectral Matting". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1699—1712, 2008.

[11] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random Sequences and Arrays," *Proceedings of the IEEE*, 64:1715—1729, 1976.

[12] R. A. Morano et al., "Structured Light using Pseudorandom Codes," *PAMI*, vol. 20, no. 3, pp. 322—327, 1998.

[13] J. Salvi et al., "A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement," *Pattern Recognition Letters*, vol. 19, pp. 1055—1065, 1998.

[14] J. Salvi et al., "Pattern Codification Strategies in Structured Light Systems," *Pattern Recognition*, vol, 37, no. 4, pp. 827—849, 2004.

[15] A. R. Smith and J. F. Blinn, "Blue Screen Matting," *ACM SIGGRAPH*, pp. 259–268, 1996.

Figure 10. Automatically inserted content responding to global brightness changes.

[16] J. Sun, J. Jia, C.-K. Tang, H.-Y. Shum, "Poisson Matting," *ACM SIGGRAPH*, pp. 315—321, 2004.

[17] R. Y. Tsai, "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*, vol. RA-3, no. 4, pp. 323—344, 1987.

[18] D. Wagner et al., "Pose Tracking from Natural Features on Mobile Phones," *Proc. ISMAR*, 2008.

[19] J. van de Weijer et al., "Edge and Color Detection by Photometric Quasi-Invariants," *IEEE PAMI*, vol. 27, no. 4, pp. 625--630, 2005.

[20] L. Zhang et al., "Rapid Shape Acquisition using Color Structured Light and Multi-pass Dynamic Programming," *International Symposium on 3D Data Processing Visualization and Transmission*, pp. 24—36, 2002.

[21] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE PAMI*, vol. 22, no. 11, pp. 1330—1334, 2000.

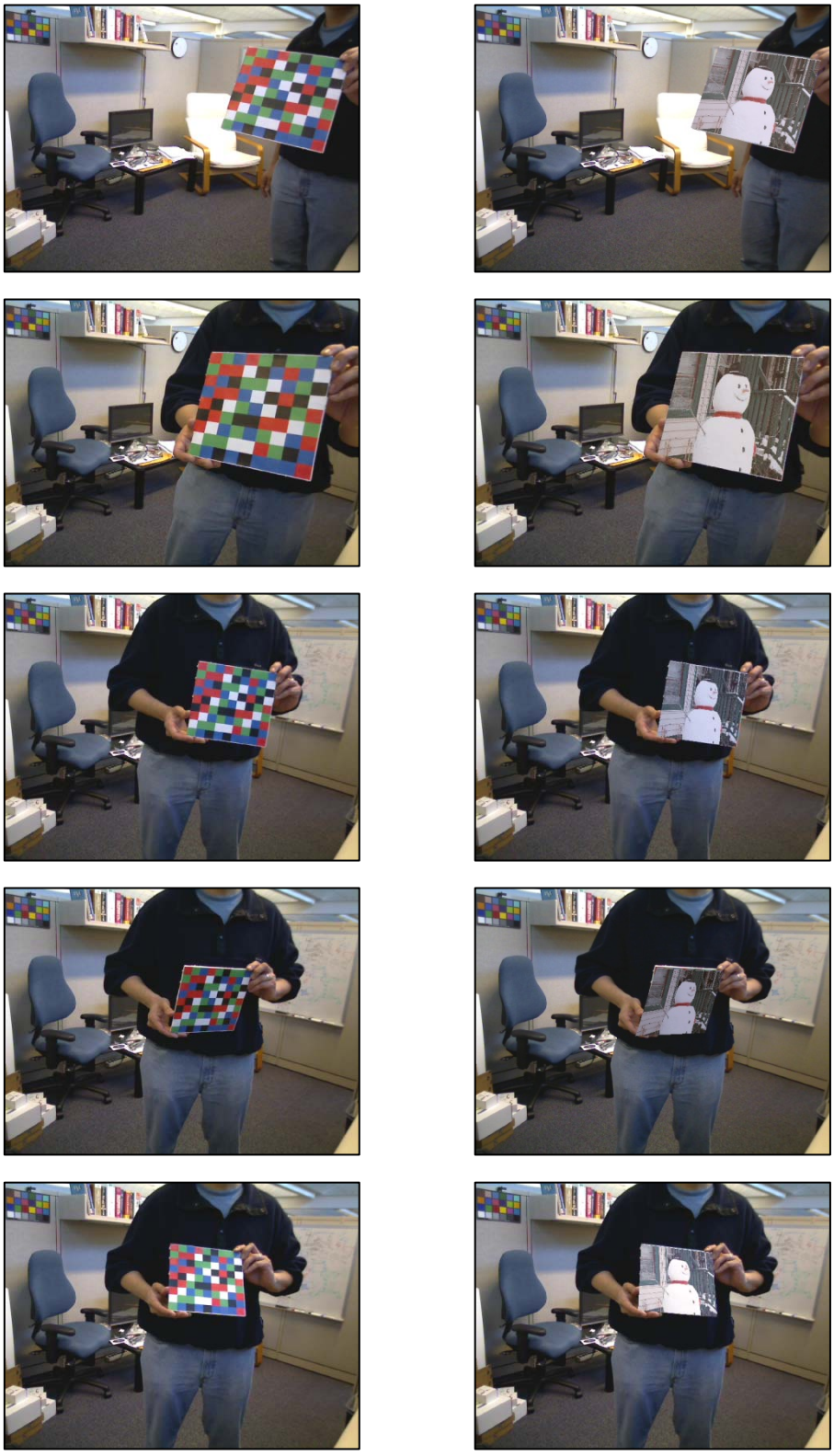


Figure 11. Representative frames of a video sequence with automatically inserted content using ChromaCoded patterns.