SMART DISPLAY: A MOBILE SELF-ADAPTIVE PROJECTOR-CAMERA SYSTEM

Tien-Ju Yang, Yi-Min Tsai, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
{denru,ymtsai,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

Owing to the diversity of projection surfaces, an effective mobile display system must be adaptive to the surface to avoid introducing a clipped scene. In this paper, we propose a smart mobile display system which automatically adapts to the location and motion of a surface. Firstly, an imperceptible structured light technique is adopted and continuous adaptation is accomplished. Secondly, a specifically designed code image with high distortion tolerance is proposed. Thirdly, we present a priority-based correction method to revise previous decoding results. Finally, a matching procedure resisting noise interruption is introduced. The system achieves 95% correct rate under the common indoor illuminance. In addition, the system performance is independent of the projected content and the surface shapes.

Index Terms— Projector-camera system, self-adaptive, imperceptible structured light

1. INTRODUCTION

Augmented reality (AR) is a technique trying to build a bridge connecting the real world and the virtual world. AR adds the information from the virtual world to the real world in a perceptual way. With the maturity of projector technology, the size and cost of a projector are decreasing. This makes mobile AR possible. By equipping a camera with a micro-projector, a mobile AR system is constructed. Users therefore can access the information from the digital world anywhere in a natural way. For example, the SixthSense project [1] proposes such kind of AR system. The system projects the object-related information on physical objects.

Unfortunately, the projection area is fixed, but the objects in our daily life are possible with any shapes and any size. Under such uncontrolled environments, if the surface is smaller than the projection area, dramatic information loss occurs. Similarly, a non-rectangular surface causes the same loss. To deal with these issues, a surface adaptation technique is applied to warp the content into the surface region before the content is projected.

In the work [2], a specially designed surface is used. By embedding light sensors into the surface, the surface is capable of detecting the signal sending from the projector. The relative position between the surface and the projector is therefore determined. However, the applications of this system are limited, because common objects which are not sensor-embedded can not be used. Lee et al. propose a method that uses a LED light source to project both visible light and infrared onto a surface [3]. This system locates the surface through the received information from infrared sensors installed on the projector side. By using invisible light, the approach can perform surface adaptation without disturbing users. The drawback of this method is that it needs a specially designed projector system, which has the ability to project and detect infrared. Leung et al. use pure computer vision techniques to track the projection surface [4]. The advantage is that the system does not need any specifically designed devices. However, this approach can only handle rectangular surfaces.

In this work, we propose a self-adaptive projector-camera display system to overcome the shortcomings mentioned above. This system is composed of a light-weight projector and a camera to have high mobility. It adopts an imperceptible structured light technique to achieve continuous adaptation to projection surfaces. The remainder of this paper are organized as follows: Sec.2 briefly describes the system flow. Sec.3 introduces the imperceptible structured light codification technique. Sec.4 describes the surface localization method. Experimental results and discussions are made in Sec.5. Sec.6 concludes this paper.

2. SYSTEM OVERVIEW

The proposed system flow is illustrated in Fig. 1. The system is separated into two main stages, imperceptible structured light codification and surface localization.

In the imperceptible structured light codification stage, the luminance of a pre-warped input frame is modulated imperceptibly according to a code image. A code image is composed of many pattern primitives. The spatial layout information is encoded by the combinations of these primitives. The code-image-embedded frame is projected and captured by a synchronous camera. The code image can be extracted through subtracting two adjacent frames. A primitive decoding step is then applied to detect and classify each pattern in the extracted code image into its corresponding primitive. A reconstructed code image is then obtained.

The second stage, surface localization, is performed once every N frames. N is a pre-defined parameter. The system checks previous N reconstructed code images and corrects the false-decoded primitives. After that, the final reconstructed code image is matched to the reference code image (the original one). According to the matching result, the visible portion of the projected frame is determined. We therefore locate the projection surface in the projector coordinate. Finally, the warping parameters are updated, and the system warps the coming frames to the surface.

3. IMPERCEPTIBLE STRUCTURED LIGHT CODIFICATION

To find the geometric relationship between the camera and the projector, geometric registration must be applied. This can be achieved by directly matching the feature points extracted from the reference

978-1-61284-350-6/11/$26.00 ©2011 IEEE

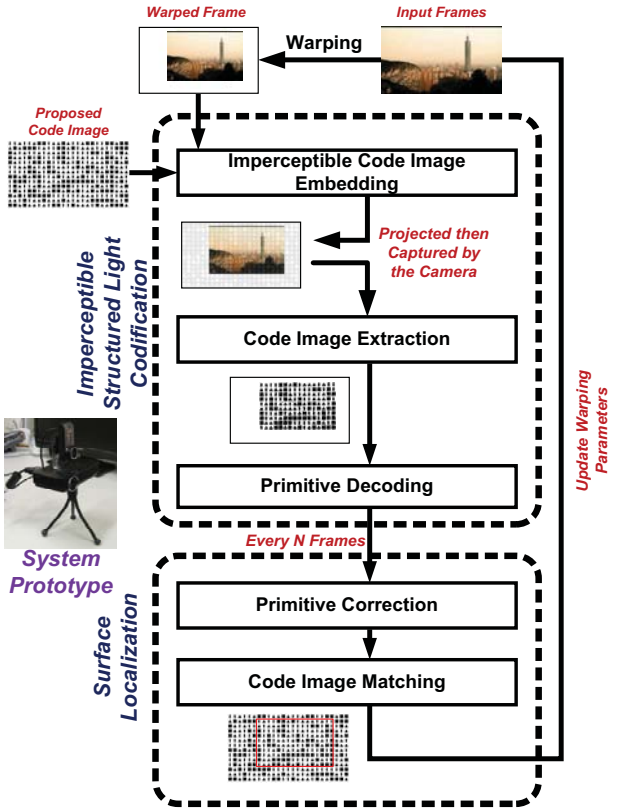


Fig. 1. The proposed system flow.

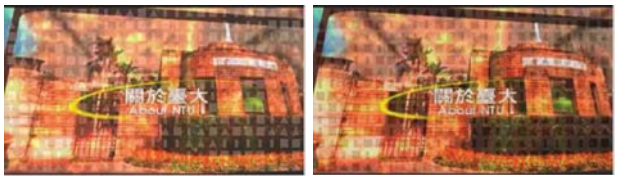
frame to the ones extracted from the projection result [5, 6]. However, the performance of this method depends on content. Instead, the imperceptible structured light technique is preferred. Owing to some weaknesses of the human visual system, some pre-defined code images are embedded into frames invisibly.

The first applicable imperceptible structured light technique is presented by Cotting et al. [7]. This method is based on the principle of DLP projectors. It modifies some specific time slots of a DLP projector to a pre-defined code image. The difference of time slots which are less than one millisecond can not be perceived by human, but can be captured by a synchronous camera. However, it may causes tone change of the scene.

Another possible method is based on the persistence of vision phenomenon. Human tends to combine the images within a short period to a single one. With this characteristic, modulate-then-compensate-based methods are proposed. Raskar et al. propose an approach to project the code image directly and compensate it in the following frame [8]. Let I be the original frame and I_{cod} be the code image. The frame for compensation I_{com} is computed by Eq. 1.

$$I = (I_{cod} + I_{com})/2 \tag{1}$$

However, the modulation of frames remains sensible during eye movements of observers. Grundhofer et al. exploit more human visual system limitations, and achieve truly imperceptible embedding [9]. This study modulates the luminance of a frame with some small change Δ to give a code-embedded image I_{emb}, and perform



(a) (b)

Fig. 2. An example of code-image-embedded frame pair. (a) The odd frame. (b) The even frame.

compensation in the next frame (Eq. 2).

$$I_{emb} = \begin{cases} I + \Delta, & \text{if } I_{cod}(x,y) = \text{binary-1} \\ I - \Delta, & \text{if } I_{cod}(x,y) = \text{binary-0} \end{cases}$$

$$I_{com} = 2I - I_{emb} \tag{2}$$

In this paper, we adopt the concept presented by [9] because of its excellent imperceptibility, and propose a modified method. We utilize this technique to embed a code image into frames, and achieve continuous surface adaptation.

3.1. Imperceptible Code Image Embedding

Before embedding the code image into frames, we should duplicate each frame of a normal 30 fps sequence in the case of 60 fps projection rate. Duplicating frames maintains the same perceptual rate as the sequence is played under 30 fps. Moreover, it avoids the false-extraction under a dramatic scene change between successive frames. To modulate luminance without changing chrominance, we transform frames from RGB into YCbCr color space, and then modify Y. Eq. 3 describes the embedding process according to a binary code image I_{cod}. Let I_{odd} and I_{even} denote the odd and even frame in the duplicated sequence, and I'_{odd} and I'_{even} are the corresponding frames in the embedded sequence.

$$I'_{odd}(x,y) = I_{odd}(x,y) + D(x,y)$$
$$I'_{even}(x,y) = I_{even}(x,y) - D(x,y)$$

where

$$D(x,y) = \begin{cases} \alpha, & \text{if } I_{cod}(x,y) = 1 \\ 0, & \text{if } I_{cod}(x,y) = 0 \end{cases} \tag{3}$$

α is a modulation value. Fig. 2 shows an example of the successive code-image-embedded frames. Because an image has limited dynamic range, the overall luminance must be scaled into a smaller range to avoid saturation at the pixels of lower and higher luminance. The contrast reduction can be mitigated by dynamically choosing different α according to human visual characteristics.

3.2. Code Image Extraction

The system captures each projected frame pair by a synchronous 60Hz camera, and then extracts the embedded code image via Eq. 4.

$$I'_{cod}(x,y) = \begin{cases} 1, & T(I'_{odd}(x,y) - I'_{even}(x,y)) = 1 \\ 0, & T(I'_{odd}(x,y) - I'_{even}(x,y)) = 0 \end{cases}$$

where

$$T(x) = \begin{cases} 1, & x \geq th_{ex} \\ 0, & x < th_{ex} \end{cases} \tag{4}$$

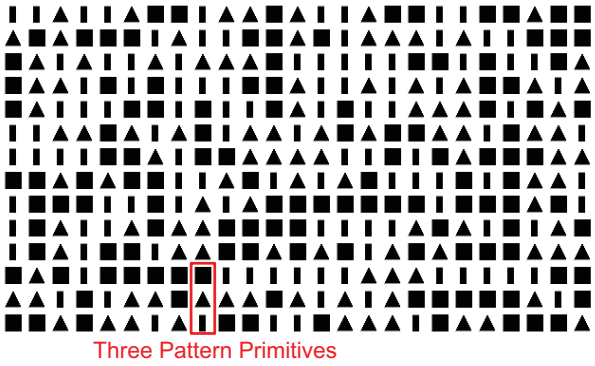


Fig. 3. The proposed code image and three proposed pattern primitives: rectangle, triangle, and stripe.

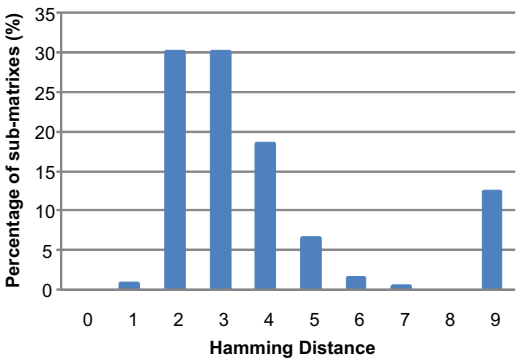


Fig. 4. The Hamming distance distribution of the proposed code image under 3×3 dimensions of sub-matrixes.

I'_{cod} is the extracted binary code image, and th_{ex} is an extraction threshold. To guarantee all patterns intact, th_{ex} must be chosen as small as possible. In our experiments, the threshold value of five is sufficient to remove great majority of noise.

3.3. The Proposed Code Image Design

In many previous geometric registration works, the authors choose checkerboard images as their code images. The main advantage of using such code images is the existence of simple detection algorithm and strong noise immunity. The registration can be done by comparing the positions of intersectional corners between the original and the projected checkerboard image. However, because the projected surface may be smaller than the projector's projection area, only partial projected image will be seen. The corners in a checkerboard image are all the same. Hence, we cannot determine the exact position without seeing the complete image.

We propose a specially-designed code image based on the Perfect Maps theory (Fig. 3). The Perfect Maps theory has been adopted in [10, 11]. Let M be a $r \times m$ matrix which is composed of s primitive types. If the Hamming distance between each $p \times q$ sub-matrix ($p \times q \leq r \times m$) of M is at least one, M is called a Perfect Map. With the uniqueness of each sub-matrix in a Perfect Map, the surface position can be determined under the condition that only partial projected frame is seen.

The designed code image has fourteen rows and twenty-five columns. To simplify the primitive decoding procedure and increase the robustness, we choose three simple geometric patterns as pattern primitives in the code image (Fig. 3). In the proposed code image, 3×3 dimensions of sub-matrixes are adopted and the Hamming distance of at least one is achieved. The Hamming distance distribution of the proposed code image is shown in Fig. 4. We can observe that the sub-matrixes having the minimum Hamming distance of one are only 1%. Over 12% of sub-matrixes have the maximum Hamming distance of nine. Moreover, the distance increases if the dimensions of sub-matrixes are higher than 3×3.

3.4. Primitive Decoding

The contours of all patterns are extracted from the extracted code image, and further approximated as a group of polygons. A noise removal procedure is then executed to increase noise immunity. We use a simple rejection function NR to reject polygons whose area are larger or smaller than some thresholds and preserve the shapes having rational areas (Eq. 5).

$$NR(poly) = \begin{cases} \text{preserve}, & min_{ar} < area < max_{ar} \\ \text{reject}, & \text{otherwise} \end{cases} \quad (5)$$

where $poly$ and $area$ denote a polygon and the polygon's area, respectively. min_{ar} and max_{ar} are minimal and maximal area.

After removing noise, a decoding function DA based on analyzing the geometric property of approximated polygons is applied (Eq. 6).

$$DA(poly) = \begin{cases} \text{triangle}, & angle = 3 \\ \text{stripe}, & angle = 4 \text{ and } ratio > th_{htw} \\ \text{rectangle}, & angle = 4 \text{ and } ratio \leq th_{htw} \\ \text{unknown}, & \text{otherwise} \end{cases} \quad (6)$$

where $angle$ is the number of angles, $ratio$ denotes a height-to-width ratio, and th_{htw} is a threshold. Firstly, search all shapes and pick out the shapes having three angles, and label them as triangle primitives. Secondly, the four-angle shapes from the remaining polygons are checked, and classified according to their height-to-width ratios. If the ratio exceeds a pre-defined threshold, it corresponds to a stripe primitive; otherwise, it is a rectangle primitive. Finally, the remainder are labeled unknown.

4. SURFACE LOCALIZATION

Owing to the existence of non-ideal effects, the extracted patterns may be dramatically distorted, which leads to wrong primitive decoding. Moreover, some noise could be falsely-included and introduce extra primitives. To solve this problem, we utilize the temporal information to correct the primitives and reject noise. This stage, surface localization, is therefore performed once every N frames.

4.1. Primitive Correction

In primitive correction step, the decoded primitives in the previous N frames are analyzed. Primitives are corrected or rejected according to the following criteria. The first criterion is that the primitive must appear in more than $N/2$ frames. Otherwise, it is removed. This criterion removes falsely-decoded noise because of its randomness nature.

Table 1. Results of decoded primitive types with respect to the true primitive types. Each row denotes a true primitive type, and each column represents the decoded primitive type.

	Total Pattern	Triangle	Stripe	Rectangle	Unknown
Triangle	11556	43.23%	0.17%	44.99%	11.60%
Stripe	12198	2.09%	78.92%	6.29%	12.70%
Rectangle	13696	0.16%	0.18%	83.51%	16.16%

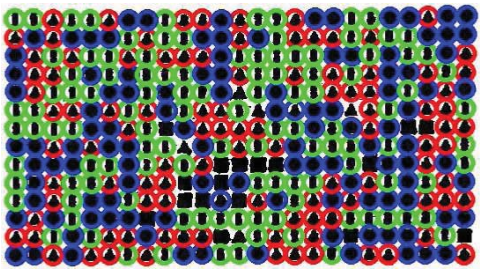


Fig. 5. An example of a reconstructed code image. The red circle represents a triangle primitive, the green circle denotes a stripe primitive, and the blue circle represents a rectangle primitive.

The second criterion is that the primitive type is redefined based on a primitive priority according to the decoding results in previous N frames. The stripe has the first priority, the triangle has the second, and the rectangle has the last. For example, if a primitive is decoded as a rectangle four times and as a triangle one time, it is corrected to a triangle. In our experiment (Table 1), we observe that the rectangle and triangle primitive are both possible to be decoded as a rectangle. The primitive decoded as a rectangle in more than $N/2$ frames cannot guarantee it is a rectangle primitive. Therefore, the primitive priority is adopted to solve the problem. The stripe has highest priority because of its one-to-one mapping relationship between the true type and the decoded type. A rectangle is nearly impossible to be classified as a triangle. We can undoubtedly redefine a primitive as a triangle if there is at least one triangle in the previous N frames. After the ambiguity between triangle and rectangle primitives is eliminated, the rectangle type can be correctly assigned. An example of a reconstructed code image is shown in Fig. 5.

4.2. Code Image Matching

Although the correctness of the reconstructed code image is verified, a few falsely-decoded noise may still exist. If these misclassified primitives are outside the boundary of the surface, the locating result would be wrong when the matching procedure is applied totally inside the reference code image.

We propose an overshoot full-search matching (OFSM) procedure. Fig. 6(a) illustrates the flow of OFSM conceptually. The OFSM starts from matching the lower right corner of the reconstructed code image to the upper left corner of the reference code image. The matching window then shifts in a zigzag-like manner. Matching is performed at each windowed portion of the reference code image. The window shifting step is a primitive unit. Until the windowed part only contains the lower right primitive in the reference code image, OFSM finishes. We then pick out the best matching window which includes least different primitives and take the bounding box of correctly-matched primitives in it (Fig. 6(b)). The bounding box is the visible portion of the projected frame, and is also the projection surface position in the projector coordinate. The next coming frames are therefore warped to fit the bounding box.

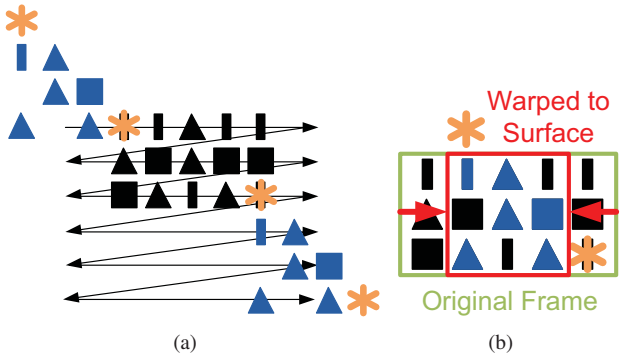


Fig. 6. Overshoot full-search matching (OFSM) procedure. (a) Conceptual illustration of OFSM flow. The black part denotes the reference code image, and the blue part represents the reconstructed code image. Falsely-decoded noise is denoted as the '*' symbol. (b) The image is warped to fit the bounding box of the best matching window.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

To validate the proposed system, we have built a prototype system. This system consists of a micro-projector and a camera. The projector is an Optoma PK301 DLP projector. It provides 1280×1080 resolution and 20 ANSI lumens brightness. The camera is a Logitech C905 with 640×480 resolution. These two devices are both suitable for mobile usage and with affordable cost. The projection surface localization stage in these experiments is performed once every five frames (N is five). Furthermore, the testing environment is the one with 320 Lux illuminance, which is the common indoor scenario. In the following, we divide the experiments into static and dynamic parts. The static experiments evaluate the robustness of the system under different conditions by using a fixed surface. The dynamic experiments demonstrate the system performance when a surface is moving.

5.1. Static Experiments

The experiments in this part are taken with different sequences, different surface shapes, and different distances. Table 2 summaries the results. The correctness is defined according to two criteria. The first is the size difference between the located surface and the ground truth is no more than 20% of the ground truth's size. The second is the center position variation between the located surface and the golden is less than 10% of the golden's size. Because the correctness of center position is more important than that of size, the limitation of center difference is stricter. Moreover, results of 2000 frames are averaged to give a final correct rate. As observed, the system achieves 95.75% correct rate. In the following, we discuss each testing condition in detail.

Different Sequences. Three sequences are used for testing. The NTU sequence is an introductive video with moderate scene changes and texture. The MI3 video is an action movie trailer

Table 2. Results of static experiments under different conditions.

	Different Sequences			Different Shapes			Different Distances		
Sequence	NTU	MI3	WT	NTU	NTU	NTU	NTU	NTU	NTU
Shape	Rectangle	Rectangle	Rectangle	Rectangle	Circle	Trapezium	Rectangle	Rectangle	Rectangle
Distance	80 cm	80 cm	80 cm	80 cm	80 cm	80 cm	40 cm	80 cm	120 cm
Correct Rate	94.25%	87.75%	95.75%	94.25%	88.50%	90.50%	93.75%	94.25%	82.75%

which has rapid scene changes and complex content. The WT is a software tutorial video which consists of monotony texture and is lack of features. From the results, all the correct rates exceed 87% and the highest one is 95%, which proves that the proposed system possesses the property of content-independence.

Different Shapes. In this case, we choose diverse projection surfaces with different shapes, including rectangle, circle, and trapezium. The lowest correct rate occurs when using circular shape surface. The reason lies in that the extracted patterns are partially clipped at the boundary of such surface, which causes the unstable primitive decoding performance. However, the correct rate over 88% is still satisfactory.

Different Distances. Based on the length of human arms, three distances between the system and the surface, 40 cm, 80 cm, and 120 cm, are chosen. The correct rates are above 93% for both 40 cm and 80 cm conditions. It slightly drops 10% under the 120 cm situation. The farther the distance is, the less primitives are shown on the surface. Only small amounts of primitives available will lead to the decrease of stability and the increase of size variation. However, the results demonstrate the system maintains 82% correct rate at 120 cm range, which exceeds the arm length of most people.

5.2. Dynamic Experiments

In the dynamic experiments, we test the system performance with a moving surface. Three surface shapes are chosen for demonstration, which are the same as that in Sec. 5.1. The setting takes the 80 cm distance and the MI3 sequence as the projected video. Fig. 7 demonstrates the results by showing some snapshots taken during the projection period.

Before applying surface adaptation, only part of a frame can be seen, and an observer can not perceive complete content. When surface adaptation is performed, the frame shrinks to fit the surface and intact content is seen. Once the surface is moved, the surface's position is re-calculated. The frame is then warped to the new location. In some cases, a few patterns which are not primitives would be extracted, such as the contours of an arm. These unexpected patterns do not affect the correctness of surface adaptation.

6. CONCLUSIONS

In this work, a robust self-adaptive projector-camera display system is proposed. Through using a micro-projector, the system achieves high mobility and automatically fits the projected content for any available object. We adopt the imperceptible structured light technique to accomplish continuous adaptation. By using the proposed code image, the projection surface is uniquely located even under the condition that only part of the projected image is seen. To increase noise immunity, the primitives are redefined based on a primitive priority according to the decoded results from the previous frames. Finally, a proposed OFSM procedure is taken to match the reconstructed code image to the reference code image, and the surface location is found. According to experimental results, the highest correct rate achieves 95.75%, and the lowest one exceeds 82%. The results prove that the proposed system can adapt to different surfaces under diverse conditions with satisfactory precision and robustness. This system can be extended to a smart human computer interface for various applications in the future.

7. REFERENCES

[1] P. Mistry, P. Maes, and L. Chang, "WUW-wear Ur world: a wearable gestural interface," *International Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 4111–4116, 2009.

[2] J. Lee, S. Hudson, J. Summet, and P. Dietz, "Moveable interactive projected displays using projector based tracking," *User Interface Software and Technology, ACM Symposium on*, pp. 63–72, 2005.

[3] J. Lee, S. Hudson, and P. Dietz, "Hybrid infrared and visible light projection for location tracking," *User Interface Software and Technology, ACM Symposium on*, pp. 57–60, 2007.

[4] M.C. Leung, K.K. Lee, K.H. Wong, and M.M.Y. Chang, "A projector-based movable hand-held display system," in *Proc. IEEE CVPR*, June 2009, pp. 1109–1114.

[5] S. Gupta and C. Jaynes, "The universal media book: tracking and augmenting moving surfaces with projected information," *Mixed and Augmented Reality, IEEE/ACM International Symposium on*, pp. 177–180, Oct. 2006.

[6] T. Takahashi, N. Numa, T. Aoki, and S. Kondo, "A geometric correction method for projected images using SIFT feature points," *Projector Camera Systems, ACM/IEEE International Workshop on*, vol. 1, no. 212, pp. 1, 2008.

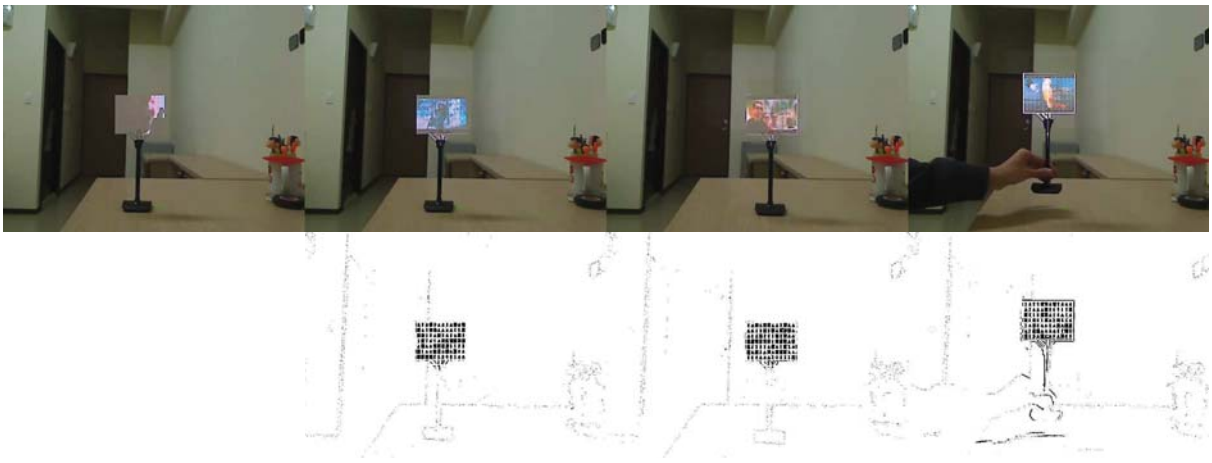
[7] D. Cotting, M. Naef, M. Gross, and H. Fuchs, "Embedding imperceptible patterns into projected images for simultaneous acquisition and display," *Mixed and Augmented Reality, IEEE/ACM International Symposium on*, pp. 100–109, 2004.

[8] R. Raskar, G. Welch, M. Cutts, A. Lake, and L. Stesin, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *Proc. ACM SIGGRAPH*, 1998, pp. 179–188.

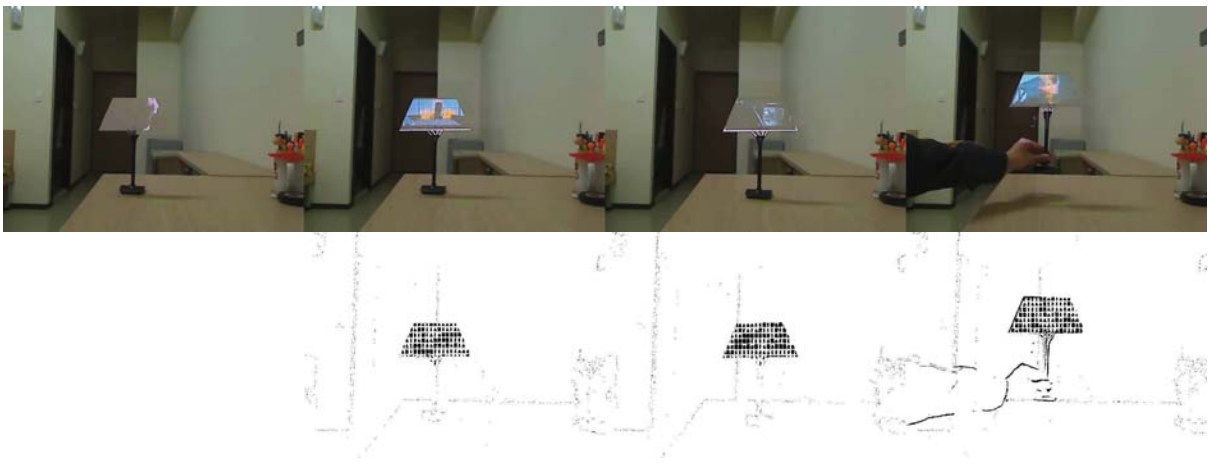
[9] A. Grundhofer, M. Seeger, F. Hantsch, and O. Bimber, "Dynamic adaptation of projected imperceptible codes," *Mixed and Augmented Reality, IEEE/ACM International Symposium on*, pp. 181–190, Nov. 2008.

[10] C. Albitar, P. Graebling, and C. Doignon, "Robust structured light coding for 3D reconstruction," in *Proc. IEEE ICCV*, 2007, pp. 7–12.

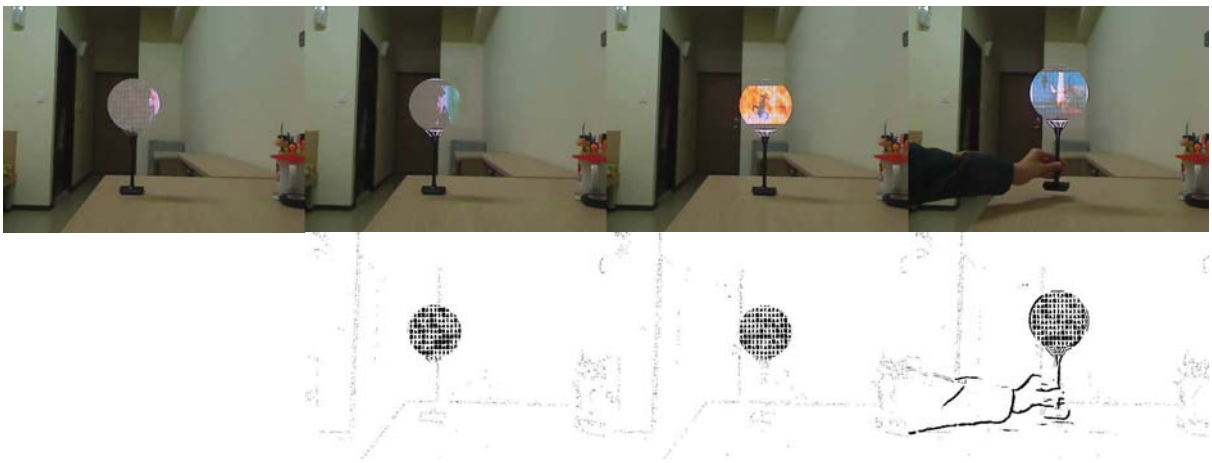
[11] J. Pages, C. Collewet, F. Chaumette, and J. Salvi, "An approach to visual servoing based on coded light," in *Proc. IEEE ICRA*, May 2006, pp. 4118–4123.



(a)



(b)



(c)

Fig. 7. The demonstration of dynamic experiments. In each subfigure, the first row shows the images from the camera, and the second row represents the corresponding extracted code images. The first column shows the snapshots before the surface adaptation is performed. The second, third, and fourth column are snapshots at different times and with different surface locations after the surface adaptation is applied. (a) Rectangular surface. (b) Trapezoidal surface. (c) Circular surface.