AUTOMATIC KEYSTONE CORRECTION FOR SMART PROJECTORS WITH EMBEDDED CAMERA

Baoxin Li and *Ibrahim Sezan*
Sharp Laboratories of America Inc.
5750 NW Pacific Rim Blvd., Camas, WA 98607, USA

ABSTRACT

We propose embedded algorithms for enabling a smart self-adjusting projector that is capable of fully automatic keystone distortion detection and correction, plus zooming and centering the image to fit a screen. The projector has an embedded camera and it automatically detects the screen and a specific pattern during self-adjustment. Proposed algorithms meet severe computational requirements of a portable projector and they exploit the physical constraints in a projector-(embedded) camera pair in a novel way in order to not compromise image quality under different usage conditions. We will demonstrate our projector system during the conference.

1. INTRODUCTION

As portable projectors becoming smaller and lighter, they are becoming ubiquitous personal mobile appliances that are used under a variety of different conditions by home users and professionals. Therefore, features that provide ease of use and added flexibility are becoming more and more important for this new class of projectors. In this paper, we are concerned with self-adjusting and self-correcting projectors that automatically adapt the projection image to different usage conditions, thereby enabling flexible positioning without compromising from image quality and without requiring manual user adjustment. These automatic functionalities are enabled by an embedded camera, and embedded computer vision and image processing algorithms. Embedded cameras are becoming pervasive in personal mobile devices, e.g. phones, at reasonable costs.

Most of the currently available projector products allow for manual push-button correction of vertical keystone only. The user presses a button to activate the correction system. The user visually inspects the projected image and stops pressing the button when he/she visually observes the corrected image. With these projectors, it is not possible to correct for keystone effect other than the vertical keystone. In other words, it is not possible to correct what is referred to as "diagonal" keystone, the most general form of keystone that is due to deviation from orthogonal projection where the deviation has a combination of both vertical and horizontal components. (See Fig. 1 for a depiction of horizontal deviation angle between the projection surface and the image plane of the projector.)

In this paper, we address the problem of fully automatic diagonal keystone distortion detection and correction, zooming, and centering the image on the projection screen. Automatic keystone detection and correction is achieved by a 3-step algorithm including (i) projection screen detection, (ii) detection of a specific projection pattern, and (iii) determining the parameters that are then used by the projector to pre-transform the image prior to projection to correct the keystone, using the results of (i) and (ii).

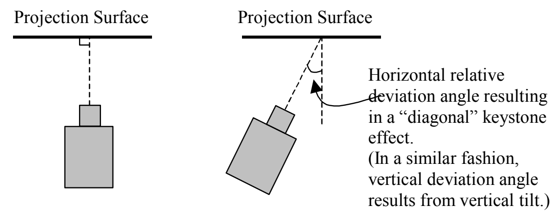

Figure 1. An illustration of horizontal angular deviation of the projector that causes diagonal keystone effect.

Our automatic keystone detection and correction algorithms are different than existing approaches (e.g., [1,2]), as will be detailed in Section 4. In particular, for the first time in the literature, we address and solve in this paper the automatic keystone correction problem for a projection screen which is not necessarily within a close proximity of a background surface, e.g., a wall, i.e., a projection screen placed in the middle of a large room, enabling a wider range of usage for our mobile personal projector.

In Section 2, we describe the proposed algorithms. We then report on the performance of our prototype system. Section 4 compares our algorithms and projector with related prior work.

2. PROPOSED METHOD

In this section, we present our approach to building a fully automatic self-adjusting projector. Our focus in this paper is on automatic keystone distortion detection and correction using screen detection and projection pattern detection.

2.1. Keystone Distortion Detection and Correction

When the projector's effective optical axis is not orthogonal with respect to the projection plane, we encounter keystone distortion. That is, the (typical) rectangular imaging panel of the projector will be projected as a non-rectangular quadrilateral on the projection plane. The purpose of keystone correction is to obtain a rectangular projection (even under a non-orthogonal projection angle) through adjusting the optics of the projector (e.g., lens shifting) and/or pre-warping the picture on the projector's imaging panel. In any event, the main problem is to determine the amount of distortion, which can be done in various ways. For example, one can directly measure the projection angle via range sensing devices, or by measuring the deviation of the distorted projection against known physical landmarks. In recent projectors featuring manual keystone correction functionality, the distortion is determined by a user's inspection of a projected picture.

The proposed method measures the amount of distortion through detection of a projection screen (reasonably assumed to be a rectangle of known aspect ratio) and specific projected patterns. The main idea is as follows. The projection process of a projector is modeled by a simple abstraction, i.e., a mapping of the image in the projector's imaging panel onto a projection plane outside the projector (e.g., a screen or a wall). This mapping can be approximated by a transformation function F_P of an ideal thin lens model. Similarly, the imaging process of a camera is described by a transformation function T_C of an ideal thin lens model. Therefore, referring to Fig. 2, the images in the three involved domains are related by the following set of equations.

$$\mathbf{I}_s = F_P(\mathbf{I}_{pj})$$
$$\mathbf{I}_c = T_C(\mathbf{I}_s)$$
$$\mathbf{I}_c = Q(\mathbf{I}_{pj})$$

Since the images in all the three domains can be assumed to be planar, all the involved transformations reduce to homographies of the planar images, and thus F_P, T_C, and Q are all modeled by an 8-parameter perspective transform [3].

In the proposed method, the keystone distortion is measured by computing F_P and then comparing the results with that of an orthogonal projection (distortion-free). Note that I_s is in fact not accessible by the projector-camera system, so it must be removed from the equations. Simple computation yields

$$F_P(Q^{-1}(I_c)) = T_C^{-1}(I_c)$$

In the above equation, T_C can be estimated after the image of a physical screen (assumed to be a rectangle of known aspect ratio) is detected in the camera domain. Thus, we can solve for F_P, and consequently determine the keystone distortion. Keystone correction is then achieved either by adjusting the projector optics or by pre-warping I_{pj} so that the effective F_P is equivalent to an orthogonal projection.

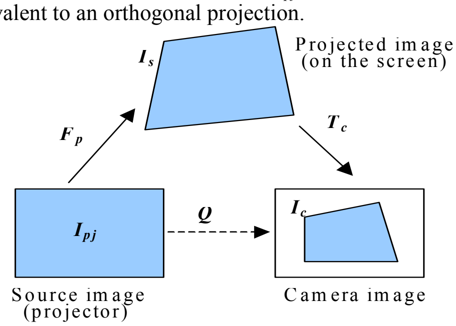

Figure 2. Relating the images in the projector, the camera, and the screen domains by transformations.

2.2. Screen Detection

It is obvious from the previous discussion that screen detection is critical to estimating keystone distortion. We will show later that it also plays other important roles in automatic setup of the projector. In this subsection, we describe a particular screen detection algorithm that is based on line-by-line processing of selected rows and columns of the input image from the imaging sensor. The line-by-line processing is needed especially when the on-board memory is limited for cost reasons. The idea and algorithm flow diagram are illustrated in Fig. 3, where we have used evenly-spaced 9 rows and 9 columns as an example. On each of the processed lines (in white), we carry out a 1-D detection procedure, as outlined in the diagram, which determines the position of the screen on that line (in green). Then, we integrate the results from all the lines (18 in the example shown in Fig. 3) to obtain the corners of the screen, through fitting the end points of the detected segments into four lines and then finding the intersections of the four lines.

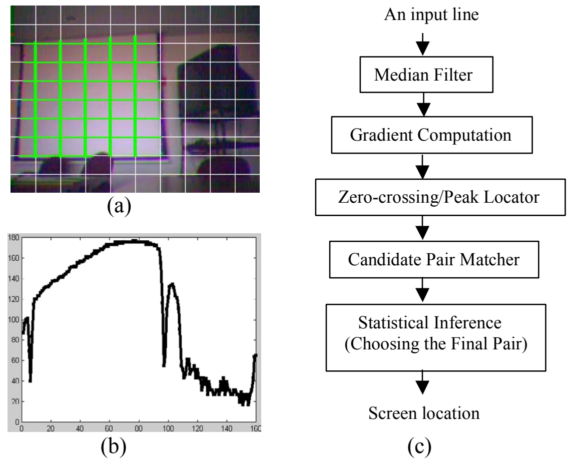

Figure 3. Illustration of screen detection. (a) A sample image. (b) The luminance values of the center row in (a) as a 1-D signal. (c) 1-D screen detection algorithm.

The line-by-line approach saves memory (no need to store the whole image) and computation (only a small fraction of the data is processed). More importantly, since we use line-fitting to obtain the final screen corners, we gain robustness compared with any other approach that tries to detect the screen corner directly via 2-D corner detection (e.g., finding a point with curvature).

2.3. Projection Detection

In order to determine the projected area (i.e., the quadrilateral region on the screen when the user turns on a projector), we need to project a test pattern and then detect the corners of the quadrilateral. Note that, if we are only interested in making the projection rectangular, we may not need to determine the corners of the largest quadrilateral corresponding to the full-resolution image of the projector. Namely, we can use a smaller rectangular image within the projector's full resolution, as will be discussed in Section 2.4. However, knowing the corners of the largest projection enables the system to zoom down or up, to fit a given screen, and to center the projection if the projector is able to shift the projection (e.g., through lens shifting) (These are features that [1,2] cannot provide). Therefore, we want to determine the corners of the largest projection. For this purpose, we have to make sure the test pattern can provide us with sufficient information for determining the corners of the largest projection. One simple test pattern is illustrated in Fig. 4, where we project a green frame of the projector's full resolution. The outer corners of the green frame determine the largest projection.

Figure 4. A sample test pattern: A green frame.

Since the green frame is similar to the black frame of a screen, one may think that we can use the same algorithm for screen detection to detect the green frame. However, it is obvious from Fig. 4 that the projected green pattern interferes with the screen boundaries, and thus some of the major modules in the screen detection algorithm (such as the "candidate pair matcher") cannot work correctly. To overcome this difficulty, we modify the 1-D detection step by exploiting the following idea: with fixed camera-projector relationship, the four sides of the green pattern will lie within some fixed ranges that can be measured. Consequently, the 1-D detection step is modified and outlined as follows. A search range is determined for each of the four sides of the green frame. Within each search range, a 1-D matched filter is used for detecting the green frame, first in the chrominance domain (using U, V components of the input YUV data), then refining in the luminance domain (using Y data). After the 1-D detection step, as in the case of screen detection, we use line-fitting to integrate the individual points. At this stage, we utilize the following idea to make the line-fitting algorithm more robust: when the camera is located close to the projector's optical axis, the projected green frame appears to be nearly rectangular in the camera domain. In principle, if the camera's optical axis coincides with the effective optical axis of the projector, then the camera should always see an undistorted green frame. This enables the removal of unlikely candidates from the 1-D step.

2.4. Keystone Correction Using a Small Pattern

With both the screen and the projected pattern detected, the keystone correction can be achieved using the method outlined in Section 2.1. In practice, however, the detection of the projected pattern is subject to significant negative impact of the operation environment. Even with the ideas discussed earlier, which exploit many novel constraints, the projection detection algorithm may still not be accurate enough in situations such as when the initial projection happens to be on black boundaries of a screen (or on a dark-colored background wall), and thus the pattern is invisible to the camera. Other situations include cases where the screen stands in the middle of a room and thus the pattern cannot be seen except for those parts that happen to be within the screen.

Therefore, it is desirable that the pattern be projected inside the screen boundary. In our system, since we have a screen detection module, the system knows where the screen is and thus it is easy to control the projection so that the test pattern is inside the screen (provided that the initial positioning of the projector enables the projection to cover at least a significant portion of the screen, which should be assumed always anyway). The problem is, as discussed above, we need to know the corners of the largest projection even if we want to a use small test pattern such that the detection can be done in the clean background of the white screen. Note that, the largest pattern is not a simple Euclidian-transformed version of the small pattern because the camera optics does not coincide with that of the projector. We now present an algorithm for deducing the camera image of the largest pattern from that of a small pattern, without actually projecting that largest pattern. That is, if we know the image (from the camera) of a small test pattern, how can we determine the image of the largest pattern? The problem is illustrated in Fig. 5, and the proposed algorithm is given below.

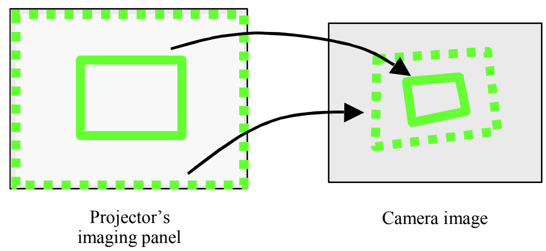

Figure 5.A small rectangle pattern in the projector domain (the solid line in the left image) induces a small quadrilateral in the camera domain (the solid line in the right image). We want to deduce the camera image (the dashed line in the right image) of the largest rectangle in the projector domain. Note that, the dashed line in the left image indicates that the pattern is not projected in reality.

Algorithm: Determinining the image of the virtual projection of the full resolution

Step 1. In the projector domain, denote a small rectangular pattern by $\mathbf{I}_{s\_pj}$, and the largest rectangle by $\mathbf{I}_{b\_pj}$, with $T_0(\mathbf{I}_{s\_pj}) = \mathbf{I}_{b\_pj}$, where T_0 is a selected transformation (e.g. a pure scale transformation by a factor of *2*). In practice, T_0 can be determined according to the detected screen size and location, so that $\mathbf{I}_{s\_pj}$ can be kept well inside the detected screen

Step 2. In the camera domain, detect the image $\mathbf{I}_{s\_cm}$ of $\mathbf{I}_{s\_pj}$.

Step 3. Construct a small rectangle $\mathbf{V}_s$ of the same aspect ratio as that of $\mathbf{I}_{s\_pj}$.

Step 4. Compute the transformation T_1 between $\mathbf{I}_{s\_cm}$ and $\mathbf{V}_s$ such that $\mathbf{I}_{s\_cm} = T_1(\mathbf{V}_s)$.

Step 5. Construct a big rectangle $\mathbf{V}_b$ as follows:

$$\mathbf{V}_b = T_0(\mathbf{V}_s).$$

Step 6. Compute the virtual image $\mathbf{I}_{b\_cm}$ of $\mathbf{I}_{b\_pj}$ as follows:

$$\mathbf{I}_{b\_cm} = T_1(\mathbf{V}_b)$$

With the above algorithm using a small pattern, the projection detection can be performed only within the detected screen, which improves the accuracy greatly since the detection is done on a clean background. The virtual largest projection is then computed from the above algorithm, and then keystone correction (plus other operations such as auto zooming) is performed. The whole procedure is illustrated in Fig. 6, where we have included images with highlighted results corresponding to the major steps from a typical execution.

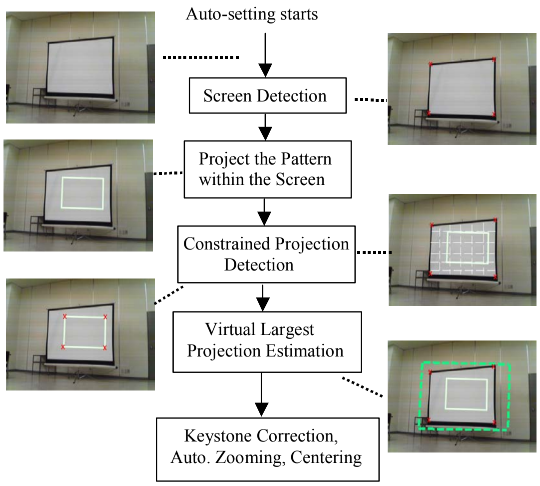

Figure 6. The procedure of automatic correction. The images represent either the input to or the output results of the corresponding blocks (connected by dotted lines). In particular, the center image on the right-hand side illustrates that the detection of the projected pattern is achieved within the detected screen region.

2.5. Centering and Zooming

With a screen detection module described in Section 2.2, we can automatically center the projection by moving the projector or the projector lens (the current prototype uses the lens-shift function). For example, if the projection is to the left of the screen, the screen detection algorithm can detect that and outputs a set of control parameters that cause the projector/lens to be shifted towards the right-hand side. In our current implementation, centering is realized by aligning the centers of the detected two quadrilaterals corresponding to the screen and the projection, respectively. Correct zooming parameters are similarly estimated by comparing the size of the detected two quadrilaterals.

2.5. Lens-Distortion Compensation

The embedded camera needs to have a wide-angle lens to capture various sizes of screens in any environments. Low-cost, wide-angle camera lens suffers from significant lens distortion, which will affect the accuracy of the keystone correction algorithm, even if the detection results are perfect in the image domain. Therefore, lens-distortion needs to be compensated before keystone correction. Any suitable lens-distortion correction algorithm can be applied (e.g., see [4]).

3. IMPLEMENTATION AND PERFORMANCE

A prototype system has been built based on the algorithms proposed in this paper, using a very inexpensive CMOS imaging sensor. An on-board chip with only a 48MHz processor plus a 6K bytes of data memory is used to run all the algorithms in this paper. In current simulation, this system is able to complete all the correction steps within 5 seconds. The prototype has been tested with nine different screens in various rooms and buildings under various projection angles and ambient lighting conditions. Hundreds of runs have been carried out, and the system is proved to be robust and accurate for a consumer application. The accuracy was evaluated by a focus group of engineers.

4. COMPARISON WITH EXISTING WORK

There is no existing projector that can achieve the same level of automation as the presented prototype. There are two research papers [1,2] that have presented related work. In contrast to [1,2], our algorithms are embedded algorithms designed to satisfy severe limitations on computational and memory requirements, but at the same time, provide required speed of operation, accuracy (i.e., image quality), and robustness under a variety of usage conditions, necessary for a consumer product. For example, severe memory limitations prevent storing a 2-D image data, forcing processing of only a number of lines and pixels data only. Proposed algorithms also take advantage of physical constraints in a projector–(embedded) camera pair in a unique manner in order to increase accuracy and robustness.

More specifically, in [2], an independent camera is used for keystone correction only. It cannot enjoy the benefits of inexpensive and robust algorithm design achieved only through exploiting an integrated projector-camera pair. Furthermore, [2] cannot fit the projection to the screen through auto-zooming and/or centering. In [1], the projector-camera pair is utilized to achieve stereoscopic vision for keystone correction only. It does not provide screen-fitting capability either. Even for the keystone correction part, it demands an accurate calibration of the projector-camera pair, which may be easily lost by any movement of the projector lens (e.g., during zooming or focusing), rendering the approach impractical (see [5] for examples of how small lens movement can result in big changes in the camera calibration parameters). In addition, neither of the above papers deals with the lens distortion issue, which is unavoidable for the projector.

5. SUMMARY AND CONCLUSION

We have developed and implemented image processing and analysis algorithms for fully automatic detection and correction of most general form of keystone effect, satisfying severe computational requirements of a low-cost portable projector. The algorithms are robust in a wide variety of usage conditions.

5. REFERENCES

[1] R. Raskar and P. Beardsley, “A self-correcting projector”, IEEE Conf. on Computer Vision and Pattern Recognition, 2001.

[2] R. Sukthankar, R. Stockton, and M.D. Mullin, “Smarter presentations: exploiting homography in camera-projector systems,” IEEE ICCV Conference, 2001.

[3] G. Wollberg, Digital Image Warping, IEEE Computer Society Press, 1990.

[4] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations”, Proc. 7th Int. Conf. on Computer Vision, 1999.

[5] R.G. Willson and S.A. Shafer, “What is the center of the image”, Journal of the Optical Society of America, Vol. 11, No. 11, 1994.