Interacting with Live Preview Frames: In-Picture Cues for a Digital Camera Interface

Steven R. Gomez
Brown University
115 Waterman St.
Providence, RI 02912 USA
steveg@cs.brown.edu

ABSTRACT
We present a new interaction paradigm for digital cameras aimed at making interactive imaging algorithms accessible on these devices. In our system, the user creates visual cues in front of the lens during the *live preview* frames that are continuously processed before the snapshot is taken. These cues are recognized by the camera's image processor to control the lens or other settings. We design and analyze vision-based camera interactions, including focus and zoom controls, and argue that the vision-based paradigm offers a new level of photographer control needed for the next generation of digital cameras.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Input devices and strategies.

General terms: Design, Human Factors.

Keywords: Digital photography, interaction, computer vision.

INTRODUCTION
In this project, we explore off-loading some interaction mechanisms to a vision-based framework that engages camera features or settings based on visual cues manipulated by the photographer. The concept is similar to Gleicher's *through-the-lens* camera control [2] for virtual cameras, but we use a real camera and consider settings other than position and orientation parameters. Our approach differs from common *automatic, vision-based mechanisms*, e.g. auto-focusing on detected faces in the frame, because it is interactive. This added interaction will be necessary for supporting embedded computational photography algorithms, like scene completion [3] or interactive alpha-matting techniques [4]. It may also increase the creative agency of the photographer by providing new interactions that engage him or her differently. Our work also differs from common *interaction mechanisms* by moving the interaction space into the image plane. By moving interaction away from the physical interface, we find great potential in removing the traditional limitation that it becomes more difficult to accommodate rich and usable features when camera size gets smaller, as is the historical trend.

UIST'10, October 3–6, 2010, New York, New York, USA.
ACM 978-1-4503-0271-5/10/10.

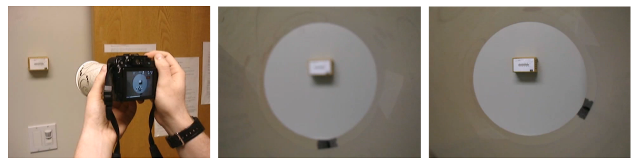

Figure 1: Manual focus with vision-based interaction. As the photographer rotates a cardboard barrel around the lens, the dark marker inside is positioned around a reserved region of the image, and the focal length changes from shallow to deep. Sample video of this feature is available at: http://www.cs.brown.edu/people/steveg/camvids.html.

We examine our use of two control cue schemes we designed and prototyped: changing the values of *reserved pixels* in the field of view, and making *gestures* in the field of view. With reserved pixels, the user may frame the shot deliberately (e.g. making the bottom-left corner of the picture dark) to engage or change a camera setting, then crop the snapshot as needed once it is taken. The motivation for this scheme is that pixel counts on digital cameras continue to increase beyond what is necessary for typical print sizes; sacrificing some pixels for new interactions may be acceptable. With gesture recognition, pixels are not sacrificed (as the gesture takes place over a series of frames before finishing), but this may require more computation than searching reserved pixel regions.

We argue that these techniques allow for fine-grained user control that will be needed in cameras with the next generation of image processing features.

IMPLEMENTATION
Our experiments use a modified version of the Canon Hack Development Kit (CHDK) [1] to process and respond to live preview frames on a Canon PowerShot G11 camera. CHDK is an open-source platform that provides an API to programmatically build camera features and scripts for many Canon digital camera models. For our prototype, we modified the CHDK distribution written in C to sample YUV-space pixel values in the 360x240 pixel viewport frame buffer and issue lens commands to the camera's firmware.

Interacting with Reserved Pixels in the Frame
We first explored whether manual focus – typically engaged by a physical interface on the camera – can be implemented using on-camera image processing of live preview frames.

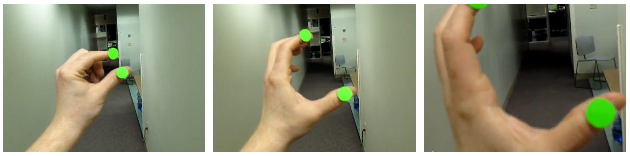

Figure 2: Frame sequence sampled during "pinch" zoom gesture.

As shown in Figure 1, we fit a cardboard cone around the lens of the camera that is seen in the photograph at all pixels beyond some radius, which we call the *image radius*, from the center of the frame. These pixels appear white in hue and are largely consistent in illumination under ambient light because the material is diffuse and white. Inside the rim of the cone, we drew a black mark at one point. Under this setup, we are able to rotate this black mark around the center of the image frame by rotating the cardboard fixture.

In computing focal length, we iterate over the viewport image buffer on each frame for the y-position of the darkest pixel outside the image radius; focal length is proportional to the height of the mark in the frame, over a 10 meter range, and is given by $1000 * y_{darkest}/viewport\_height$ centimeters. By updating the lens on each frame, we effectively create a vision-based SLR-style focus mechanism that allows the photographer the rotate the cardboard lens barrel to adjust focal length.

Gestures

We also developed a finger pinch gesture in the field of view to change the zoom length on the lens. "Pinching" zooms out; "unpinching" zooms in. This scheme is similar to multi-touch gestures that use pinching to examine images, e.g. on Apple's iPhone. When the desired zoom is reached, the user removes his or her fingers from the field of view and releases the shutter.

In our prototype, the *zoom step* parameter is proportional to the change in Euclidean distance between the highest and lowest "finger" pixels recognized between consecutive frames from the viewport pixel buffer. For testing purposes, we identify fingers using neon green markers, which have distinct hue and saturation against most backgrounds, on the thumb and index finger. This could be extended to include more robust finger detection. We check for these finger pixels in the buffer during iterations of CHDK's histogram update function, then set the computed zoom step at the end of this function.

DISCUSSION

Motivating New Camera Interactions

We want interaction methods on cameras that will keep up with the next generation of on-board camera features. In automated scene completion [3], for instance, we need a region-of-interest selection tool that allows the user to identify a hole in the image plane to fill automatically with internet image data. The user could indicate this region quickly with the swipe of a finger. Data-driven techniques like scene completion will be useful to have on the camera because their results can vary greatly based on the photographer's vantage point,

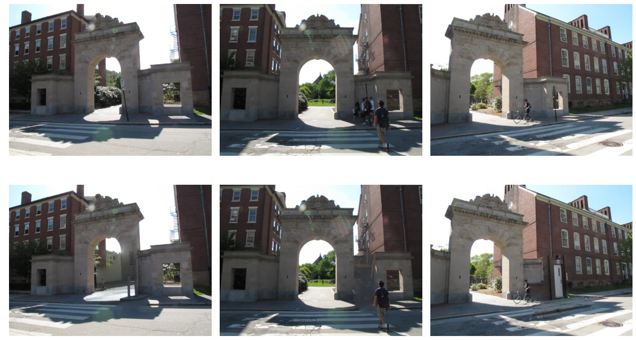

Figure 3: Automatic scene completion results from our implementation of [3] can differ greatly based on the photographer's vantage point. The bottom row shows the results of removing the road sign to the right of the arch in each corresponding photograph from the top row.

as demonstrated in Figure 3. Getting immediate results on the camera allows the user to retry from a different vantage point until the desired shot is taken.

Preliminary Evaluation

The design and informal testing of these features is encouraging. Our focus and zoom interactions show that both gesture recognition and manipulating reserved pixels in the field of view can yield viable camera control paradigms. A general challenge we discovered is that designing fluid live preview interactions can be inhibited by underlying camera limitations, but we expect usability to improve as manufacturers improve their devices' computational resources and failure modes.

Our exploration of in-picture cues also reveals a unique challenge for an interface with feedback: altering the camera will affect recognition of future in-picture cues. For instance, after pinching to zoom, the fingers will appear positioned differently depending on how the lens has moved, even if the user has stopped moving his or her fingers. We are exploring ways of modeling common photographic interactions, like lens focus and zoom changes, to account for camera changes dynamically in recognizing a photographer's cues.

REFERENCES

1. Canon Hack Development Kit. `http://chdk.wikia.com/wiki/CHDK`.
2. Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *SIGGRAPH Comput. Graph.*, 26(2):331–340, 1992.
3. James Hays and Alexei A. Efros. Scene completion using millions of photographs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 4, New York, NY, USA, 2007. ACM.
4. Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.