CPSC 490 Proposal: *Generating Videos from Barcodes*

Qingyang Chen — Advisor: Holly Rushmeier

February 4, 2017

1 Motivation

Imagine being able to scan a visual code on any object and be able to watch a video on how to use that object. QR codes, for example, can be used to encode URLs for videos that can be watched over the Internet. However, this requires the video be stored on a remote server and the viewer to be connected to the Internet.

The proposed project will solve these two issues by generating the video directly from an image of the object itself that contains the visual code.

2 Overview

The main problem to solve is how to encode a video into a visual code. For example, a 49x49 QR code can store about 194 bytes of data with low error correction [1]. Even a minimally acceptable video bitrate of 16 kbps [2], a one-second video would already be over 10x the storage capacity of this QR code.

3 Proposed Methodology

The solution is to reduce the amount of data to be encoded in the visual code as much as possible. The proposed solution is to encode just the animation of shapes in a photo taken of the object. This consists of two main parts — 1) encoding an animation of polygon outlines, and 2) processing an image to apply the animation to matching shapes.

A use flow would thus look like:

1) The *creator* would use an animation tool to define polygon outlines and animation transitions applied to those outlines over certain periods of time.
2) The *creator* exports this as a visual code to be printed onto the designated object.
3) The *user* takes a photo of the visual code along with the object.

4) An image processing tool uses the visual code to:
   a) Match the polygon outlines to the shapes in the image.
   b) Apply the animation transitions to the matched parts of the image.
   c) Fill in any holes in the image.
   d) Render the animation for the *user* to view as a video.

Some relevant techniques may include edge detection, Hough transform, and poisson blending.

4 Deliverables

The project deliverables consists of two parts: 1) an animation tool for defining polygons and animation transitions (this generates the code), and 2) an image processing tool that converts a photo and a corresponding code into an animated video.

The deliverables will *not* include a methodology for representing the code visually, as that can be accomplished by a multitude of existing technologies.

5 Example

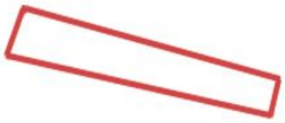

Figure 1

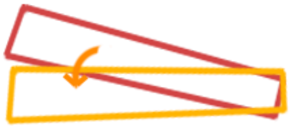

Figure 2

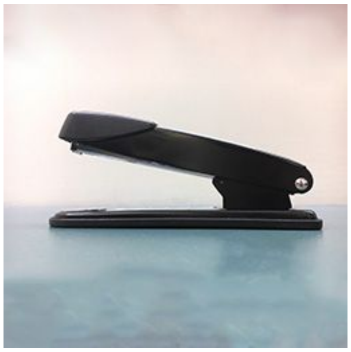

Figure 3

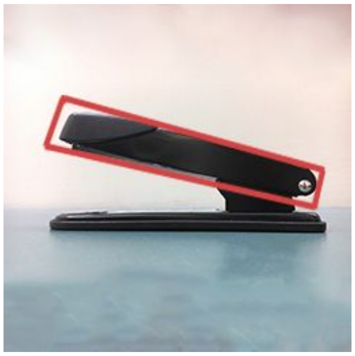

Figure 4

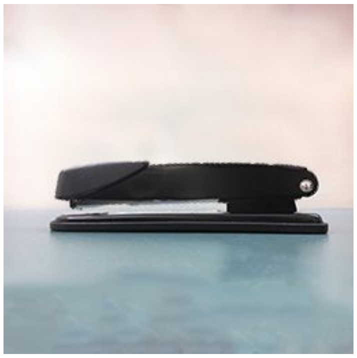

Figure 5

The above figures show an example of the generation of an animation video on how to use a stapler. The animation parameters would be encoded by some visual code printed onto the stapler.

Figure 1 shows the *creator*-defined polygon to be matched. *Figure 2* shows the *creator*-defined animation to be applied to the matched portion of a photo. *Figure 3* is a photo the user would take of the stapler. *Figure 4* overlays the polygon on top of the photo to show where it is matching to. *Figure 5* shows the end result of the animation applied to this photo. The intermediary frames were omitted in this example, but would constitute a video of the top side of the stapler depressing.

6 References

[1] "Information Capacity and Versions of the QR Code." *Information Capacity and Versions of QR Code*. DENSO WAVE INCORPORATED, n.d. Web. 04 Feb. 2017. <http://www.qrcode.com/en/about/version.html>.

[2] "Bit Rate." *Wikipedia*. Wikimedia Foundation, n.d. Web. 04 Feb. 2017. <https://en.wikipedia.org/wiki/Bit_rate#Video>.