An Alternative Real-Time Image Processing Tool

Shalom Darmanjian[1,2] A. Antonio Arroyo[1] Eric M. Schwartz[1]
oxygen@ufl.edu arroyo@mil.ufl.edu ems@mil.ufl.edu

[1] Machine Intelligence Laboratory
[2] Computational NeuroEngineering Laboratory

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6200

ABSTRACT

In order to perform real time image analysis, researchers are often required to purchase expensive processing equipment. The video frame grabber can account for the majority of the cost, with typical prices ranging between $700 and $3000. This prohibitive cost often discourages most college students and hobbyists from using these valuable learning tools. This paper seeks to address this problem with an alternative hardware solution that costs much less than traditional equipment (without sacrificing quality or speed). After motivating the discussion, this paper explores current systems on the market along with an assessment of their advantages and disadvantages. Next, we provide an overview of the proposed system as well as a comparison to features available in pre-existing systems. Finally, we will discuss current real time algorithms successfully used with the proposed system.

1. INTRODUCTION

Computer vision is often regarded as the key for autonomous agents to one day produce decisions similar to that of a human. The benefits of producing this higher-order decision ability are immeasurable to our society. Military applications and other security applications are only a few examples in which computer vision can lend great support. Other benefits include the empowerment of disabled people, in which sign language or lip reading is transformed into synthesized speech. Written text can also be converted into an audible voice for the blind. Incorporating the ability to identify certain objects, shapes or inscriptions can spawn new useful products in the commercial arena.

Unfortunately, research in computer vision is often limited to those with enough financial resources to obtain the necessary equipment. This often excludes student researchers and hobbyists from engaging in this highly beneficial form of research. Although cheap alternatives exist, they often fall short of the necessary features to truly make an impact in this field. The great leaps in the development of new technology are often brought about through the collaboration of many researchers. Limiting the mental computing power of this line of research negatively affects the arrival of new breakthroughs.

The creation of a cheap simple video frame grabber would allow for a variety of students and hobbyists to actively engage in the computer vision field. The benefits would be two fold. Firstly, students would be exposed to a new line of algorithms and new tools for design and implementation (often lacking in theoretical work). Secondly, through the assistance of these new student researchers, new advances could be made with less development time.

2. MARKET COMPARISIONS

Matrox Imaging is one of the leading producers of frame grabbers. Most of the cards are within $700-$1300 (with the purchase of a special cable) and offer a multitude of camera inputs and speeds of up to 30 frames per second (1024x768 resolution). In normal computer vision applications, multiple video sources are often unnecessary, which renders the high priced-feature baseless. Also noteworthy, the included software is proprietary and will not allow for easy adaptation of new algorithms. For developmental software, some researchers turn to Linux drivers and source code, which may leave students and hobbyists the daunting task of developing their own code libraries and device drivers.

Scion Products is another major source of research frame grabbers. Their CG-7 Model is a $1700 frame grabber that is comparable to the matrox line of hardware. This card only offers one video input with a max resolution of 640x480.

Digital web cams have been explored as a cheaper solution to computer vision. Though they offer an affordable price tag, they often lack in resolution and speed. Most high-end web-cams ($100+) can grab a maximum of 8-10 frames per second at a resolution of about 320x240. At these speeds and resolutions, most real-time applications are not feasible. Improving this web cam's resolution is not possible since the user must rely on the pre-built CCD camera inside. Alternatively, higher quality video cameras can be interchanged with a frame grabber (allowing it to be robust).

Another low-cost device can be found in USB frame grabbing devices. These products range from $80-$120 but often lack in resolution and/or speed. The Belkin USB VideoBus II is an example of a USB frame grabber that only has a max resolution of 352 x 288. The Zoran USBvision II can only capture at 15 frames per second and costs over $100.

3. PROPOSED SYSTEM

3.1 Hardware

General consumer products often drive down the price of technology. Going off this trend, we have seen a dramatic drop in computer video capture cards. A Hauppauge win TV card (Figure 1) is one such example, with prices ranging around $20 on Ebay. This card has the ability to capture 640x480 images at 30 frames/second. Also noteworthy is the ability for the card to take in either s-video or coaxial video (allowing for dual purpose VCR or camcorder interfaces). The PCI interface allows for the card to be used in virtually all computers.

Figure 1

3.2 Software

The Win TV Card contains the Brook-Tree chipset, which has been long supported under the video4linux driver. One software package that uses the Brook-Tree chipset is Xawtv. The Xawtv package is a TV viewing program for Linux. As a TV viewing software package, it has the ability to capture and store the images in memory (and display the images to the user). As is common with most software associated with Linux, the source code for this package is freely available. We incorporated our vision algorithms into this software program to convert this video capture device into an image processing frame grabber. Using pre-made software allows for two advantages. First, the development of device drivers is unnecessary. Second, this pre-built software allows for a nice user interface to see code implementations in real-time.

4. WORKING ALGORITHMS

One of the first algorithms implemented with this system was a color detection scheme. First the program is calibrated using a shirt presented to it. A histogram modeling approach then calculates the frequencies of colors and later matches those majority frequencies during real-time processing. The centroid of the color blob is tracked through time, using the mean of X and Y pixel locations. The variance of the X's and Y's of the pixels are used to track the closeness of the object (higher variance relates to more pixels, i.e. the closer the object is to camera). See Figure 2 for an example image of the color tracking.

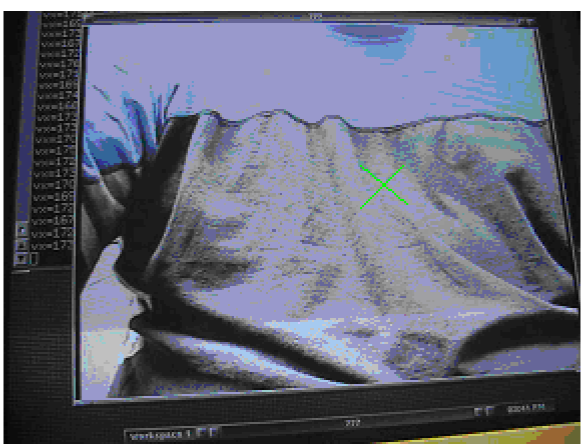

Figure 2

Another algorithm implemented in this system is a motion detection algorithm. First, we compute an image subtraction from the previous frame and current frame. Then we use a threshold to bias out the noise of the frame. The remnant of this transformation is the motion along object's edges within the frame. Figure 3 illustrates an example of this motion-tracking algorithm.

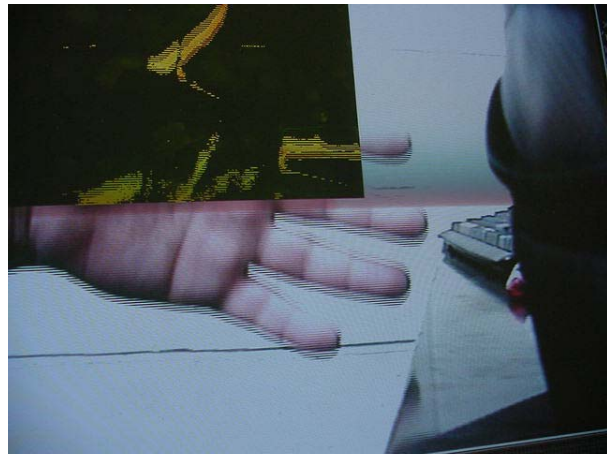

Figure 3

From the motion-tracking algorithm we are able to implement a simple video stabilization algorithm. Looking within the motion-transformed space, we search for vertical and horizontal edges to track. Currently, our algorithm is only searching for a single horizontal and vertical edge in this motion-space. Ideally we would want to track all of the horizontal and vertical features in this space (example in Figure 4).

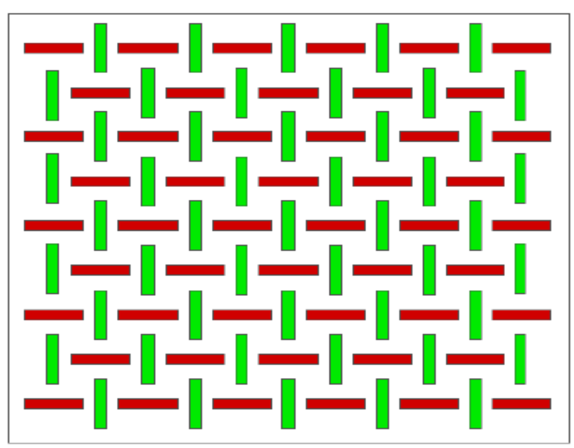

Figure 4

Once we find these features in the space, we then track them from frame to frame to acquire localized motion vectors. Next, we average the local motion vectors from the individual features to get a global motion vector for the whole frame (removing bias from outliers). See Figure 5 for a representative example.

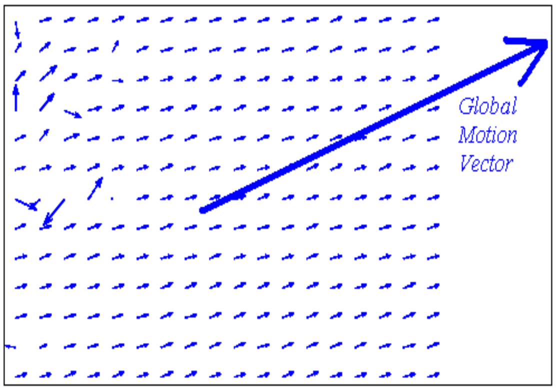

Figure 5

Finally, we use the global motion vector and transform the full frame in the opposite direction of the vector. To make sure that the global motion is not an intended motion (as in camera is turning to a purposeful direction) a damping constant is used to adjust the transformation in a desired direction. Figure 6 shows an example of this stabilized motion in the vertical direction (paper fails to display the entirety of a stabilized image).

Figure 6

5. CONCLUSION

The computer vision field will lead our society to new and beneficial technologies. Distributing the problem-space across an expanded base of researchers can accelerate and improve this technological development process. One way to expand the researcher workforce is to provide the ability for students and hobbyist to join in this endeavor. The open source movement has seen gigantic leaps in software development by offering free software and tools to the throngs of idealistic and eager students and hobbyists. With computer vision, breaking the barriers of costs will also entice students and hobbyists to work on new projects (expanding the technology to unknown lengths). Weighing speed, resolution and cost, the win TV card offers students and hobbyist the best tool in the image-processing field.