StitchRV: Multi-Camera Fiducial Tracking

Sijie Wang, Allen Bevans, Alissa N. Antle
School of Interactive Arts and Technology, Simon Fraser University
250-13450 102 Avenue, Surrey, BC V3T 0A3 Canada
{swa50, alb19, aantle}@sfu.ca

ABSTRACT

StitchRV is a fiducial and touch-tracking engine based on the popular reacTIVision fiducial tracking system. StitchRV combines video input from multiple cameras in real time, and can be customized for a wide range of hardware and fiducial tracking applications through the high-performace rapid prototyping environment openFrameworks. The multi-camera approach facilitated by StitchRV also allows greater diversity and flexibility than single-camera systems when designing computer vision based tangible and multitouch prototypes.

Author Keywords

Multiple camera fiducial tracking, visual marker tracking, fiducial markers, tangible interaction, reacTIVision, openFrameworks.

ACM Classification Keywords

H5.m. Information interfaces and presentation: Miscellaneous, I.4.9. Image Processing and Computer Vision Applications

General Term

Human Factors.

INTRODUCTION

Affordable digital video cameras and abundant computational resources have encouraged an explosion of experimentation with camera-mediated human-computer interaction, especially in the tangible, embedded and ubiquitous computing fields. Numerous prototypical object and finger tracking systems have been developed and evaluated which leverage these technologies [5, 8, 10, 11]. These prototypes and others like them have provided valuable insight into emerging human-computer interaction techniques. Because these prototypes can be designed and built at relatively low cost using easily attainable materials, researchers of modest means have been able to participate in these developments. However, most prototypes to date rely on a single camera to track user interaction, limiting their object tracking resolution as well as shape and size.

Single-camera multitouch and tangible prototype designs are limited by the placement and field of view of the camera, which must be positioned far enough away from the surface of the prototype to capture the entire interactive space. Single-camera systems that utilize projectors for visual output add an additional design challenge: The camera may need to be placed near the projector's image projection axis, without casting shadows or interfering with light-folding mirrors. Many designers have used wide-angle lenses to capture the interactive space at a shorter distance from the target surface, which can partially resolve some of these issues. However, this approach introduces image distortion, which may affect object recognition and fiducial tracking at the edges of the captured image. Single-camera systems are also limited by the resolution of the camera; low resolution video requires that markers and fiducials must be larger to track reliably to be used with the system. A single camera also limits the aspect ratio and orientation of an interactive surface to a single plane with an aspect ratio smaller than or equal to that of the camera.

If multiple cameras are used to overcome these issues, integrating the cameras into the prototype presents a non-trivial technical challenge. One approach is to combine video feeds from multiple cameras into an aggregate feed before passing them to the tracking engine. This approach usually requires using specialized video hardware (see [12] for an extreme example). Another approach is to track video from each camera on separate computers and combine the tracking data over a network before passing them to an application. Both of these approaches require extra hardware that may increase cost and complexity of prototypes and may negatively affect input processing performance (Eg. increase lag in the system's interaction feedback). StitchRV is the result of research addressing these issues, developed as part of the EventTable project [1]. The source code is available for download at: http://code.google.com/p/stitchrv/.

STITCHRV

Design requirements

The current phase of the EventTable project requires a multitouch and fiducial tracking table large enough to comfortably be used by four simultaneous users. Previous single-camera EventTable prototypes suffered from poor fiducial tracking at the edges of the display, partly due to wide-angle lens distortion and low camera resolution.

TEI 2010, January 25–27, 2010, Cambridge, Massachusetts, USA.
Copyright 2010 ACM 978-1-60558-841-4/10/01...$10.00.

In order to reduce equipment costs and prototype complexity, a software-based solution that would combine and analyze multiple cameras feeds on a single computer was pursued. This research resulted in the creation of StitchRV. In its current form, StitchRV combines high frame rate video from two USB cameras in real time to create a high-resolution feed for fiducial and touch tracking.

Software Architecture

StitchRV is implemented in openFrameworks, a C++ rapid-prototyping application framework. It combines the performance advantages and code portability of C++ with a set of libraries that allow easy customization and extension by researchers and artists with intermediate programming skills.

StitchRV combines and tracks multiple camera feeds via two main processes:

- The "video stitching" process, which combines video from multiple cameras into a single, larger video frame (figure 1).
- The "fiducial and touch tracking" process, which tracks fiducials and finger-blobs, creating TUIO [7] messages with relevant data (figure 2). This process utilizes an existing openFrameworks port of the reacTIVision fiducial tracking system [6] called ofFiducialFinder, which performs image processing and tracking calibration as well.

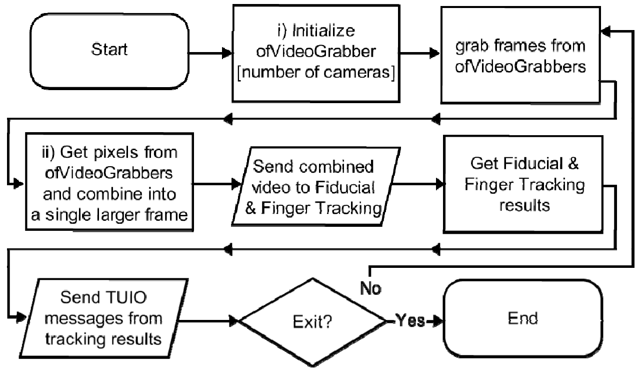
Start
i) Initialize ofVideoGrabber [number of cameras]
grab frames from ofVideoGrabbers
ii) Get pixels from ofVideoGrabbers and combine into a single larger frame
Send combined video to Fiducial & Finger Tracking
Get Fiducial & Finger Tracking results
Send TUIO messages from tracking results
Exit?
No
Yes
End

Figure 1. The video stitching process in the main loop of StitchRV.

Hardware

SitchRV is currently designed to work with a specific camera model used by the EventTable project: the Sony PlayStation Eye. This camera captures 640x480 pixel video at 60 frames per second, and can support even higher frame rates at lower resolutions. The PlayStation Eye delivers video via USB, and drivers are available for OSX, Windows, and Linux. The wide availability, high performance, and low cost of these cameras make them ideal for prototyping exploratory object and touch tracking systems.

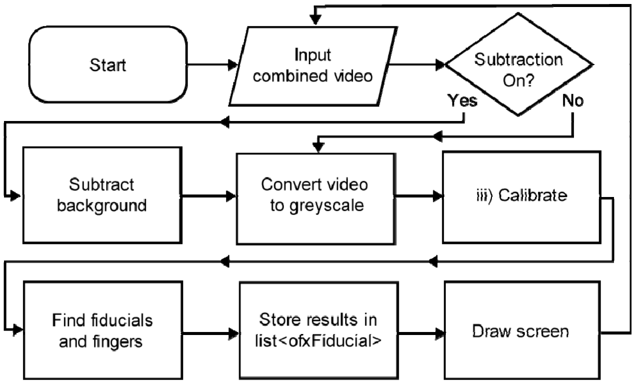

Figure 2. The fiducial and finger tracking process, implemented with the pre-existing ofFiducualFinder library.

Customizing StitchRV

Although the current version of StitchRV is limited to combining two feeds from a specific model of USB camera, it can be modified to work with additional cameras, different camera models, and unique camera orientations.

In order to utilize additional cameras in StitchRV, changes must be made to steps (i) and (ii) of the video stitching process. At step (i) an array of openFrameworks ofVideoGrabber objects determines how many cameras are accessed by the software. Each element of this array accesses the feed from a single camera. Increasing the number of elements in the array to correspond to the number of cameras attached to a system will allow the software to access each of their feeds. If StitchRV is customized for additional camera, the feed combination process at step (ii) will need to be altered as well.

Step (ii) can also be customized to switch the order and orientation the camera feeds before sending them to the tracking engine. Step (ii) creates a large, single frame by copying and appending the pixel data matrices stored in memory from each camera feed. Changing the order of the frame appending will alter the order of the stitched feeds in the resulting larger frame. Transposing an individual camera feed frame before it is appended to the large frame will change the orientation of that camera feed in the final stitched frame.

Using different camera models and different input sources (USB, firewire, tv tuner, etc.) in StitchRV is relatively simple. The ofVideoGrabber class relies on system level video libraries to capture video input, allowing any video imaging device recognized as such by the operating system to be accessed by the class. Only the frame rate and frame size parameters of the initializing ofVideoGrabber object need to be changed to match the capabilities of the hardware attached.

Color data from the camera feeds is kept until the fiducial tracking process. RGB pixel data from combined frames can be copied or passed to other functions for color tracking or for other uses by adding the necessary code immediately after step (ii).

Finally, calibration is performed after the camera feeds have been combined into a single frame. Included as part of the pre-existing ofFiducialFinder library, this calibration process can be used to correct for camera distortion and camera feed overlap before performing tracking on the combined frames. By bringing two neighboring points together on the calibration grid, image data shared between them can be eliminated, removing camera feed overlap. The size of the calibration grid can be modified in the part of the code as well, depending on the precision required for calibration.

The potential customizations outlined here cover the minimum necessary changes required to customize StitchRV for systems with requirements and specifications different from the EventTable project. However, other enhancements and extensions can be implemented as researchers see fit; the openFrameworks libraries encourage rapid development and playful experimentation.

ADVANTAGES

StitchRV enables the creation of multi-camera touch and object tracking prototypes requiring little extra equipment beyond additional cameras. Multi-camera systems have some distinct advantages over single camera systems: Multi-camera systems allow for higher resolution fiducial and touch tracking, as long as the resolution of each feed is maintained when the feeds are combined instead of being down-sampled into a single low-resolution feed. High-resolution tracking allows for consistent recognition of smaller fiducials and touches. This makes possible smaller and more densely packed tangible designs. It also supports larger touch surfaces, where finger blobs and fiducials appear smaller.

Larger surfaces beyond the shape and aspect ratio of a single camera can also be tracked by arranging multiple cameras in an imaging array. Multi-camera arrays could provide high-resolution image tracking with very high frame rates (120 fps+) by combining many low-resolution high-frame rate cameras, providing smoother system feedback and reduced interaction response lag. Larger and more responsive prototypes could provide richer research opportunities, especially in multi-user and collaboration studies.

Multi-camera tracking systems also allow for shorter distance between cameras and the interaction surface of a prototype. This is especially useful for prototypes that employ LCD displays or extremely short-throw projectors.

Multi-planar or non-planar surfaces can be tracked much more efficiently with multi-camera systems vs. single camera systems. While single-camera systems which track non-planar surfaces have been successfully implemented [2, 4], they require special lenses with extreme fields of view. This necessitates custom distortion correction and tracking analysis of their input video. Multi-camera tracking of these types of surfaces simplifies the software tracking process and minimizes the need for distortion correction in some cases. It can also simplify the design of multi-planar prototypes, especially those where all of the interaction surfaces are not viewable from a single point inside or outside the system.

Finally, multi-camera tracking systems could enable robust object tracking in three dimensions. By combining tracking data from cameras placed along different axes, relative position and rotation of a fiducial in a pre-defined space could be used for tangible computing and augmented reality research.

Because StitchRV is implemented in openFrameworks, it can utilize video streams from with different types of cameras with minimal customization. This allows researchers to leverage available cameras from previous research or lab equipment pools when creating multi-camera prototypes, without the need to purchase “matching” cameras. StitchRV also sends tracking data to other applications via the TUIO protocol, allowing integration with pre-existing prototypes and fiducial or multitouch applications.

LIMITS

In its current state, StitchRV only supports two camera feeds with a specific frame size and frame rate; it must be customized and re-compiled to be used with other camera and hardware setups. While customizing StitchRV at the source code level is intended to be a simple process, some researchers may still find openFrameworks (or its C++ foundation) intimidating. As it stands, StitchRV is not an appropriate solution for researchers without programming experience, available time to customize the source code, or a programmer available to assist them.

Because it is a software-based video stitching solution, tracking performance in StitchRV is limited by processing power. A customized system with many cameras may not have the processing resources available to efficiently analyze a large summative feed, let alone run a client application. Our current prototype, based on a Macbook Pro with dual-core 2.4 GHz processor and 2 GB memory, is able to handle up to 3 simultaneous camera feeds while maintain enough system resource to handle basic interactivity. Prototypes involving many cameras or computationally-intensive client applications may require another computer to distribute the processing load. However, even with greater processing capacities, a prototype will be limited by the data transfer capabilities of connections used by the cameras. For example, a prototype using USB cameras such as the PlayStation Eye could handle up to 8 cameras before being limited by USB’s data transfer rates.

StitchRV is ultimately a multi-camera enhanced extension of the reacTIVision fiducial tracking system; thus it shares many of reacTIVision’s limitations, including being limited to tracking the reacTIVision fidcuial set. Researchers hoping to create multi-camera systems that require color tracking, robust multi-touch tracking, or tracking of other marker sets (Eg. [9]) must either customize other software packages to make them multi-camera capable or heavily customize StitchRV in order to fulfill their tracking requirements.

FUTURE WORK

Since StitchRV currently must be customized at the source code level, it is of limited usefulness for many end users, researchers, and designers. Developing a user-friendly interface that enables StitchRV to be customized and configured at run-time is an important next step if StitchRV is to be a useful prototyping tool for as many people as possible.

StitchRV has been created to assist those exploring computer-vision based object and touch tracking. As multi-camera tracking has the potential to provide greater flexibility and diversity when designing research prototypes, it is appropriate that future research also include evaluations of multi-camera prototypes; the advantages and potential of multi-camera tracking must borne out by application and experience. An important part of future research is to build multi-camera prototypes and test them out!

CONCLUSION

The rapid pace of tangible and multitouch research has been made possible by inexpensive hardware and technical and theoretic knowledge sharing within the academic and commercial communities. Within this research area, tracking touch and objects with multiple cameras allows for greater flexibility and diversity in prototype design than single-camera tracking. StitchRV, originally created to facilitate the continuation of the EventTable project, contributes a software-based multi-camera fiducial tracking engine that can utilize inexpensive, high frame-rate cameras. StitchRV also serves as a template that can be easily extended by researchers to customize it for their own multi-camera tracking projects.

ACKNOWLEDGMENTS

Special thanks to Chris O'Shea, Zachary Lieberman and Theo Watson. This research was funded in part by NSERC.

REFERENCES

1. A. N. Antle, N. Motamedi, K. Tanenbaum, and Z. L. Xie. The EventTable technique: Distributed fiducial markers. In *Proc. TEI 2009*, ACM Press (2009), 307-313.
2. H. Benko, A. D. Wilson, and R. Balakrishnan. Sphere: Multi-touch interactions on a spherical display. In *Proc. UIST 2008*, ACM Press (2008), 77-86.
3. F. Bridell. Hardware – openFrameworks Wiki. http://wiki.openframeworks.cc/index.php?title=Hardware &oldid=10947.
4. J. B. de la Rivière, C. Kervégant, E. Orvain, and N. Dittlo. Cubtile: A multi-touch cubic interface. In *Proc. VRST 2008*, ACM Press (2008), 69-72.
5. F. Echtler, M. Huber, and G. Klinker. Shadow tracking on multi-touch tables. In *Proc. AVI 2008*, ACM Press (2008), 388-391.
6. M. Kaltenbrunner and R. Bencina. reactivision: A computer-vision framework for table-based tangible interaction. In *Proc. TEI 2007*, ACM Press (2007), pages 69-74.
7. M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. TUIO: A protocol for table-based tangible user interfaces. In *Proc. GW 2005*, Vannes, France, 2005.
8. K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proc. GI 2009*, Canadian Information Processing Society (2009), 119-124.
9. A. Kumpf. Trackmate: Large-scale accessibility of tangible user interfaces. Master's thesis, Massachusetts Institute of Technology, June 2009.
10. P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Illmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's min, don't touch!: Interactions at a large multi-touch display in a city centre. In *Proc. CHI 2008*, ACM Press (2008), 1285-1294.
11. M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. D. Hollan, and J. Borchers. Slap widgets: Bridging the gap between virtual and physical controls on tabletops. In *Proc. CHI 2009*, ACM Press (2009), 481-490.
12. B. Wilburn, N. Joshi, V. Vaish, E. V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In Proc. SIGGRAPH 2005, ACM Press (2005), 765-776.