A Motion Grid State Machine for Triggering a Document Camera

Eric Saund
Doron Kletter
Subhrendu Sarkar
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, Ca. USA
{saund, kletter}@parc.com
ss3295@columbia.edu

Abstract

This paper reports on initial experiments in adopting a motion grid state machine based approach to determining the stable regions of a scene for the purpose of triggering document capture in a casual desktop environment. A demonstration prototype system was built to illustrate desktop document capture in a document sorting application scenario.

1 Introduction

In one of the most influential envisionment videos in our field, the concept of the Digital Desk was introduced over fifteen years ago [11]. We still don't have it, yet signs are encouraging that people may some day be able to exploit computational power and network resources in our casual interactions with physical documents [1, 2, 4, 6]. Desktop computing power has increased about a hundredfold since 1993; memory and disk capacity have increased likewise; digital cameras and digital video are now ubiquitous; and knowledge in the fields of computer vision and document image analysis has grown. Arguably, work on the algorithmic and applications aspects of desktop document cameras will soon gain the opportunity to bear fruit in experimental and even practical deployment.

One of the important issues in camera-based document capture is framing documents and triggering document capture in a field of view that will include various activities. While it is possible to design a system where a document to be captured is placed in a special location and a deliberate trigger action is taken by the user, casual use scenarios will benefit from the ability to track and image documents in the course of their normal handling, wherever they appear in the field of view.

An example application is document sorting. Consider the problem of organizing a stack of grant application forms in alphabetical order, by applicant name. The user is faced with two different aspects of the task. One is the physical handling of the documents to expose each, one at a time for visual inspection, and subsequently to manually place and move it in the workspace to such locations as to lead to a sorted order. A small sorting task can be done in one pass by placing each document more or less where it belongs in an array. A larger sorting task may have to be broken into subtasks, with documents placed on several piles, each of these groups to be sorted again.

The second aspect of this task is to read the relevant information off the document and decide where it should be placed. This is where computation can help. Every child learns to sort alphabetically. Yet, the task is tedious and demanding of a certain amount of cognitive capacity. After

a while, even professional knowledge workers tire and start to slow down and make mistakes. The task could proceed much faster if an automated mechanism could immediately indicate to the user where to place each document as soon as it appears on the top of the unsorted stack. Many means for this are possible. For example, a motorized laser pointer could be used to direct the user to target document placement locations.

In order for a document camera to participate in this scenario, the system must detect where the target document appears in the field of view, and it must detect the moment at which to trigger the capture operation so that the necessary fields will be in view and stationary enough to capture a clear image. This must take place in real time while the user may be moving vigorously within the field of view, placing documents in their correct sorted locations.

Recent work on digital desk rigs has employed global motion analysis and strong simplifying assumptions about how documents can be manipulated and rearranged by the user [2]. Kim et al's Desktop Browser Interface performed all video processing offline; this work was later replicated to operate at frame rates—provided the user moved documents slowly enough—but at the cost of additional tradeoffs in lighting constraints and degraded accuracy [5].

Clearly a simple motion sensor or any frame-wide global motion detection will not suffice for a practical system because it would be triggered constantly and would not differentiate stationary from moving objects in the field of view. Methods to segment human hands based on skin color and fingertips based on hand shape [4, 3] have also been attempted. These do not address the motion of documents in the scene. Yet another popular approach is to employ tangible artifacts with carefully designed visual targets that can be more easily tracked and identified by the camera [7, 10].

In order for a visual system to detect the starting and stopping of the motion of documents in one part of the the scene while the user may remain actively in motion in another part of the scene, a practical real-time system must possess the ability to divide its attention across the scene and make decisions on a spatially local basis.

To accomplish this, we report on our initial experiments in adopting a motion grid state machine based approach to determining the stable regions of a scene for the purpose of document capture in a casual desktop environment. Our system, called *DocuTable* is implemented using separate video and still capture cameras, for technical hardware reasons. The motion grid state machine operates on the video stream, while trigged document capture occurs via the higher resolution still camera.

2 Background: ZombieBoard

Our approach was first developed in 1996 for an office whiteboard scanner application called ZombieBoard, and has proven its reliability through continuous routine use at our research center since that time [9]. See Figure 1. The whiteboard scanning application is similar to the desktop document capture application in an interesting way. ZombieBoard employs an experimental Diagrammatic User Interface, whereby users trigger actions by drawing on the whiteboard itself. Two control actions are deployed. The "Scan" operation is triggered by drawing a button and then drawing an "X" or check mark in it. The "Change Boards" operation is triggered by drawing a button with an arrow pointing to the left or right. ZombieBoard's camera is a video camera on a pan-tilt device, capable of turning toward any scanable surface in the field of view.

Under ZombieBoard, the diagrammatic user interface must operate even when the user remains in the field of view. Due to the computational cost of analyzing an image to interpret possible diagrammatic commands, it is important to trigger analysis only at appropriate times and locations, namely, as soon as the user has drawn a command, but only once per region of the whiteboard that has been newly drawn on. This means that stationary regions of the board must be analyzed even while other parts of the scene are in motion. More precisely, diagrammatic image analysis must be triggered whenever a part of the scene becomes stationary, following a period when that part of the scene has been in motion.

3 Grid State Machine

For this purpose we developed a grid state machine approach, which we applied also to Do-

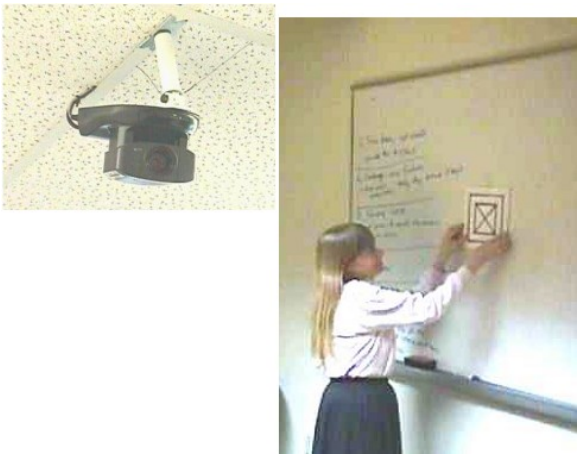

Figure 1. The PARC ZombieBoard whiteboard scanner. The pan/tilt/zoom video camera is mounted in the ceiling. The user is placing a magnetic pre-drawn "scan whiteboard" command button.

cuTable. The image is divided into tiles that define coarse regions of the board. Each tile operates an independent state machine. States are, STABLE, IN-MOTION, STATIONARY-AFTER-MOTION (Figure 2) Intially, all tiles are STABLE. At each time step, successive frames are compared. Image processing is performed to filter camera jitter and slight global illumination changes. Image differencing is applied with image processing to trigger transition to the IN-MOTION state if sufficient pixels are sufficiently different within that tile between frames. When, in IN-MOTION state, the differences between successive frames falls below a threshold, the state for that tile transitions to STATIONARY-AFTER-MOTION state. Quite often, a user standing in front of the whiteboard will move, then remain relatively still for a slight period of time. Similarly, a user will move around in the field of view of the DocuTable camera. While persons are perfectly capable of remaining very still for a long time, we set a heuristic threshold on the period of time that a tile shall remain in STATIONARY-AFTER-MOTION state, before it transitions automatically to STABLE state. Four seconds is a typical threshold. At that transition, the tile is declared to possibly contain new material written on the whiteboard or a document newly positioned on the table. It and surrounding tiles STABLE tiles are collected to form a local region which is sent for analysis.

The motion grid state machine works very effectively for the ZombieBoard whiteboard scanner. It has operated continuously in between 10 and 20 conference rooms and offices for over 12 years in our research center. Over that time it has evaluated an estimated 20 million frames, and triggered an estimated 10,000 whiteboard captures. Based on an informal sampling of frames which resulted in a whiteboard capture command, in approximately 50% of these the user remains in motion within the field of view of the camera.

4 DocuTable

For implementation on the DocuTable camera-based document capture experimental prototype, the grid motion state machine approach was applied in a straightforward manner. We divided the 640x480 video frame into 100 by 100 tiles. By coloring tile borders according to state, it is possible to visualize state transitions across the image in real time, as illustrated in Figure 4. As tiles transition from STATIONARY-AFTER-MOTION to STABLE state, they become candidates for considering that a document has come into view and should be captured at high resolution. A single tile represents only a small portion of the scene. It is when a clus-

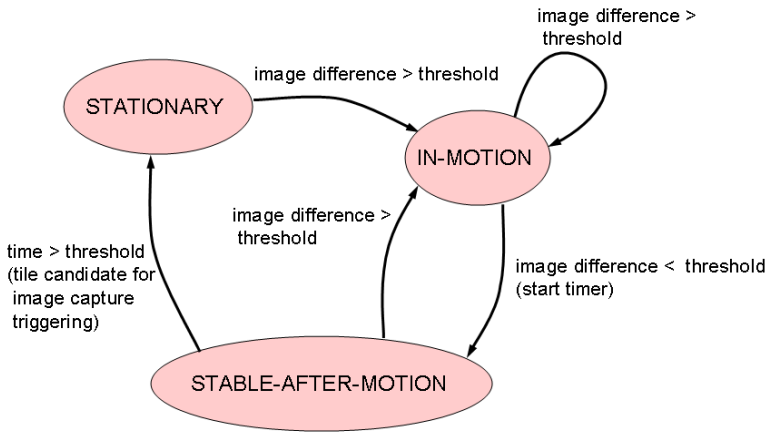

Figure 2. Per-tile state transition diagram.

ter of contiguous tiles all have recently transitioned from STATIONARY-AFTER-MOTION state that a region becomes defined for possible document capture.

In both the whiteboard and desktop scenarios, it is common for the user to move and gesture in front of stationary material, and it would be pointless to re-analyze this stationary material over and over again. For this reason, in both ZombieBoard and DocuTable the grid motion state machine operates as the inner loop of a nested system. The outer loop maintains a model of the stationary material in view, in the form of an image. As tiles transition back to STABLE state, their contents are compared to this model. Only if there is sufficient difference between these image patches is it determined that new stationary document material may have come into view. In such a case, high resolution capture is triggered and image analysis is performed on these regions of the scene. Obviously regions of blank white paper will not appear to change between the time documents are placed on the scene. These are to be treated as possibly changed and are subject to capture and analysis along with any darker, truly changed material, they are proximal to.

Figure 3. DocuTable fixture.

5 Application Scenario

Our initial application scenario was designed to approximate the document sorting application discussed above. In our trials, we did not attempt

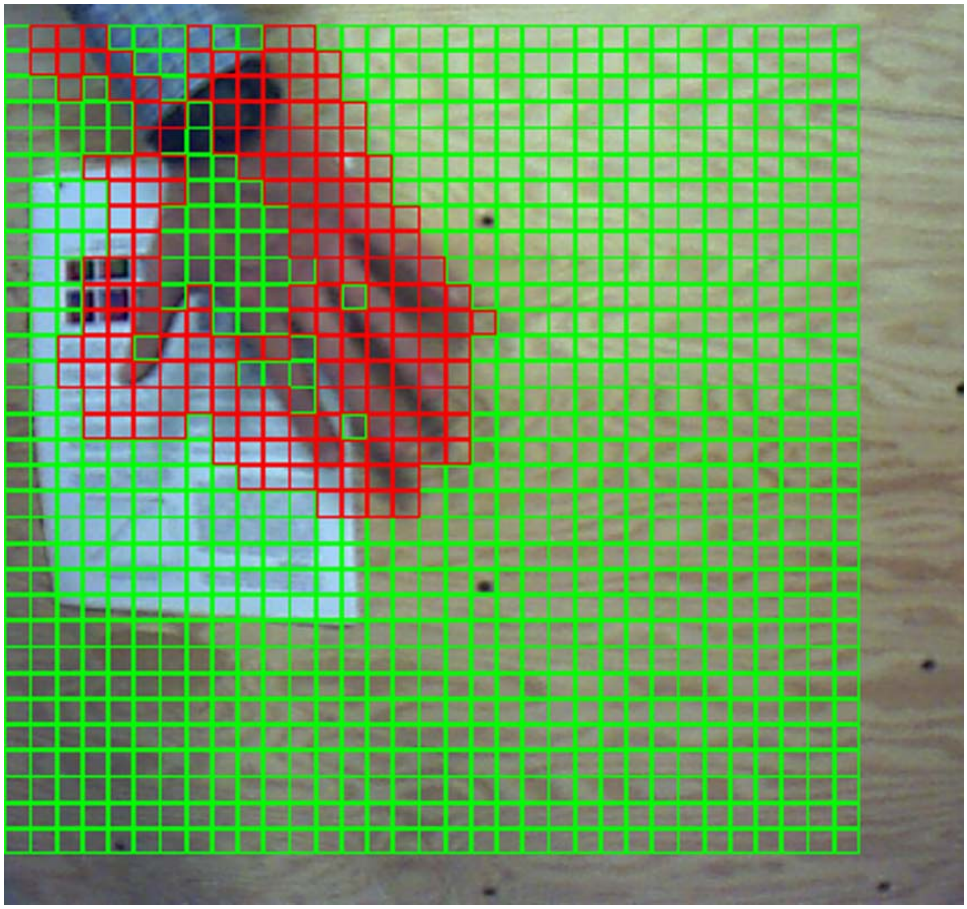

Figure 4. Tile state visualization. IN-MOTION **tiles are shown in red.**

to perform OCR or other conventional document recognition which would be subject to image quality considerations. Instead, we modeled our scenario on one in which the documents are initially viewed in a particular order, then are scattered around and the ordering lost. We suppose that the user wants to arrange the documents back in their original order. To do so does not require reading the documents, but only recognizing documents that have previously appeared in the scene.

For this purpose we employ an image-based document indexing module that is capable of storing and retrieving document images in terms of "fingerprints" based on 2D spatial arrangement of word blobs. Our implementation is heavily inspired by the pioneering work of Kise et al, who have demonstrated this sort of capability in many venues [8]. Of course, to build any practicable system requires many refinements and enhancements of the core ideas.

In our initial implementation, for document detection we employ a video camera positioned over the table to have a view area of approximately 90 x 60 centimeters. For document capture, we use a Canon SX100. In order to obtain good quality images with this inexpensive consumer camera, we confine target document placement to one region of the DocuTable platform approximately 30 x 40 cm in size. This is significantly larger than a letter-size page, allowing for capture with only casual placement of the document in the capture region of the table. In principle, the grid motion state machine approach supports general document placement anywhere on the table, but would require sufficient imaging quality over a wider field of veiw, either through the use of a higher resolution camera, or through a rapid pan/tilt foveation mechanism.

Our application employs a video GUI control. There is a "training" phase in which the user reveals the document stack, one page at a time. The grid motion state machine automatic triggering mechanism permits the user to do this at a casual pace without further deliberate action. Then the user manually switches the application to a "sort" mode. They may dis-order the documents, and present them one at a time in the capture region. The system automatically detects the appearance of the document, triggers capture by the high-resolution camera, extracts the document fingerprints matches with stored documents from the training phase, and determines where this document belongs in the sort. A directive is given the user on a video display as to which of a few piles to place the document on in order to achieve a correct sorted result.

6 Lessons Learned and Conclusion

While the main grid motion state machine technical approach has proven successful, our demonstration system is not net responsive enough to support the casual, effortless experience we all hope for from the original Digital Desk envisionments. Our main issue is speed, and the main barrier we encountered was the simple mechanics of triggering capture of images and getting them into memory fast. While the grid motion state machine process operates at an adequate 10-12 frames/second, high-resolution capture and other operations take 1-2 seconds which forces the user to pause their activity unnaturally. From an implementation point of view, this is "simply a matter of engineering," yet such engineering remains nontrivial even today. Cameras that are potentially inexpensive enough for practical deployment in real application settings are inexpensive because of their manufacturing scale. In other words, these are consumer cameras. Consumer cameras, however, are narrowly designed for consumer photography, not for laboratory experimentation. Their API's are often inadequately documented, and their data transfer capabilities are inadequate for digital desk/CBDAR purposes.

These shortcomings are important because they make it difficult both to to perform advanced experiments, and to produce compelling demonstrations that can spark the imaginations of researchers, developers, and potential users and clients, which is necessary to drive developments in this promising yet ephemeral area. We look forward to future developments on the hardware front that will enable document recognition to at last become the primary choke point for research and development of algorithms and novel applications. When that happens, the motion grid state machine can offer a vital ingredient to real-time performance in practical settings.

References

[1] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 606–616, 2003.

[2] J. Kim, S. Seitz, and M. Agrawala. Video-based document tracking: Unifying your physical and electronic desktops. *ACM Symposium on User Interface Software and Technology (UIST)*, pages 99–107, 2004.

[3] C. Kofler, D. Keysers, A. Koetsier, J. Laagland, and T. M. Breuel. Gestural interaction for an automatic document capture system. *International Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, pages 161–167, 2007.

[4] H. Koike, Y. Sato, and Y. Kobayashi. Integrating paper and digital information on enhanced-desk: A method for realtime finger tracking on an augmented desk system. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8:307–322, 2001.

[5] E. LaForce. Comp776 final project - document tracking. *http://wwwx.cs.unc.edu/ elaforc/data/DocumentTracker.pdf*, 2008.

[6] C. H. Lampert, T. Braun, A. Ulges, D. Keysers, and T. M. Breuel. Oblivious document capture and real-time retrieval. *International Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, pages 79–86, 2005.

[7] T. Moran, E. Saund, W. van Melle, A. Gujar, K. Fishkin, and B. Harrison. Design and technology for collaborage: Collaborative collages of information on physical walls. *ACM Symposium on User Interface Software and Technology (UIST)*, pages 197–206, 1999.

[8] T. Nakai, K. Kise, and M. Iwamura. Camera-based document image retrieval as voting for partial signatures of projective invariants. *ICDAR*, pages 379–383, 2005.

[9] E. Saund. Bringing the marks on a whiteboard to electronic life. *Cooperative Buildings. Integrating Information, Organizations and Architecture, Springer Lecture Notes in Computer Science Volume 1670*, pages 69–78, 1999.

[10] M. Terroy, J. Cheung, J. Lee, T. Park, and N. Williams. Jump: A system for interactive, tangible queries of paper. *Graphics Interface 2007*, pages 127–134, 2007.

[11] P. Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, 36(7):87–96, July 1993.