Video-Based Document Tracking: Unifying Your Physical and Electronic Desktops

Jiwon Kim *Steven M. Seitz*
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195, USA
{jwkim|seitz}@cs.washington.edu

Maneesh Agrawala
Microsoft Research
Redmond, WA 98052, USA
maneesh@microsoft.com

ABSTRACT

This paper presents an approach for tracking paper documents on the desk over time and automatically linking them to the corresponding electronic documents using an overhead video camera. We demonstrate our system in the context of two scenarios, *paper tracking* and *photo sorting*. In the paper tracking scenario, the system tracks changes in the stacks of printed documents and books on the desk and builds a complete representation of the spatial structure of the desktop. When users want to find a printed document buried in the stacks, they can query the system based on appearance, keywords, or access time. The system also provides a *remote desktop* interface for directly browsing the physical desktop from a remote location. In the photo sorting scenario, users sort printed photographs into physical stacks on the desk. The system automatically recognizes the photographs and organizes the corresponding digital photographs into separate folders according to the physical arrangement. Our framework provides a way to unify the physical and electronic desktops without the need for a specialized physical infrastructure except for a video camera.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Graphical User Interfaces (GUI), Interaction Styles; I.4.8 [**Scene Analysis**]: Object Recognition, Tracking

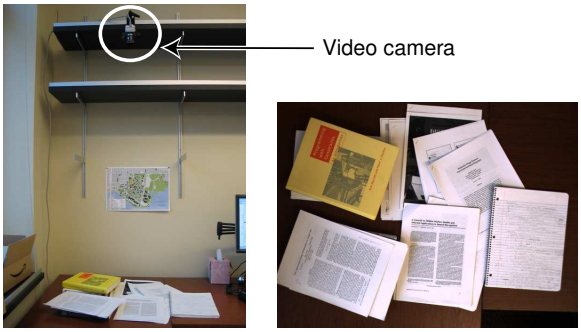
Additional Keywords and Phrases: Video analysis, document recognition, interactive desktop, intelligent office

INTRODUCTION

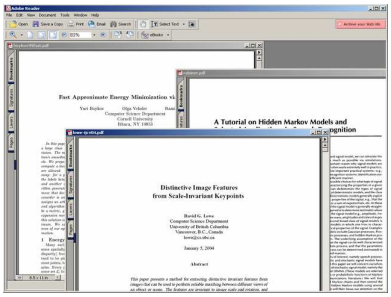
The demise of paper documents has been predicted since the advent of personal computers and electronic documents. However, paper and electronic documents still coexist in our working environment. As pointed out by Sellen et al. [22], this is due to the complementary nature of the conveniences that paper and electronic documents provide.

Paper is a natural interface that people tacitly know how to interact with [16]. It is usually a preferred medium for reading, navigation, and annotation. In addition, its physical presence makes it easy to lay out, sort and organize multiple paper documents into stacks on an extended physical space such as the desk. As noted by Kidd [9], the spatial layout of materials in an office environment is an important memory aid for knowledge workers. The memory cue can help users localize the search for a particular document to a region of the desk, but finding the exact document within the set of papers piled in the region can still be challenging, especially if the stack contains many documents.

On the other hand, electronic documents are inherently suitable for computational operations such as storage, retrieval, keyword search, sharing and version management, for which paper documents provide poor support. But the electronic interface is not as convenient and intuitive as direct manipulation of paper.



(a) Setup (b) Camera view



(c) Onscreen view of PDF's

Figure 1: Using a video camera mounted above a desktop (a, b), our system tracks and recognizes all documents and links them to the electronic versions on the computer (c).

UIST '04, October 24–27, 2004, Santa Fe, New Mexico, USA.

(a) Paper tracking sequence

(b) Photo sorting sequence

Figure 2: (a) Sample input frames from the paper tracking sequence. Printed paper documents and books enter, exit the scene and change location in the stacks as the user shifts them around. (b) Sample input frames from the photo sorting sequence. The user sorts photographs in two source stacks (one on the desk in the lower right corner, the other outside the scene) into three target stacks.

Consequently, people commonly keep both paper and electronic copies of the same document, to exploit the advantages of both media. However, the decoupling of physical and electronic versions makes it difficult to fully exploit these advantages. What is needed is an automated system that bridges this gap between paper and electronic worlds.

In this paper, we present a vision system that uses a video camera to track and recognize stacks of physical documents on the desk. The system captures the movement of documents with an overhead video camera (Figure 1). The video is then analyzed using computer vision techniques to link each paper document with its electronic copy on disk and track its physical location in the stack. The interface to our system allows users to issue queries about the documents in a few different ways: by appearance, keyword, access time and using a remote desktop interface. A key feature of our system is that it does not require obtrusive infrastructure such as physical tags and specialized readers.

Our system enables two scenarios: *paper tracking* and *photo sorting*. To illustrate the first scenario, suppose the user has a paper to review by tomorrow. He put it somewhere on the desk, but has difficulty finding it in the stacks of documents. Using our system, he can easily locate it in the stacks by performing a keyword search on all documents on the desk. Alternatively, if the desk is in a remote location, the user can directly search through the stacks using the remote desktop interface that allows the user to virtually manipulate the stacks by clicking and dragging on the image of the desk.

In the photo sorting scenario, the user has hundreds of pictures on his digital camera that he wants to organize into digital albums. It is cumbersome to go through the individual photographs on computer and place them in separate folders. On the other hand, once the photographs are printed on small sheets of paper, users can easily flip through and sort them into physical stacks. Our vision system observes the user as he sorts the photographs into stacks and automatically organizes the corresponding image files into separate folders.

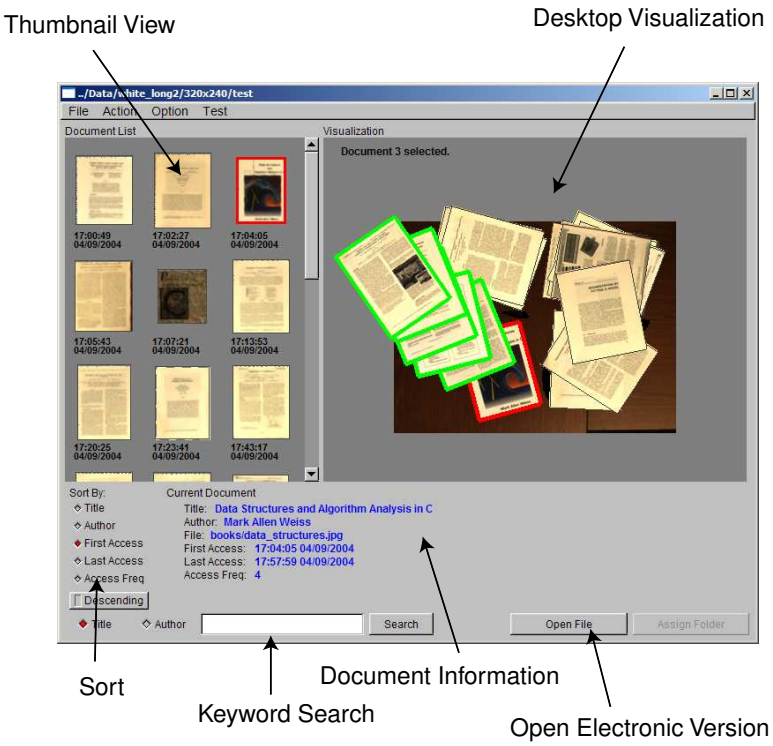
RELATED WORK

There exists a significant body of previous work on camera and projector based augmented desktop system [26, 24, 12, 2, 14]. However, their primary focus lies in supporting interaction with individual desktop objects or projected image using hand tracking. Although these systems are capable of simple object tracking, they require either manual registration of objects or the use of specially designed visual tags and backdrops.

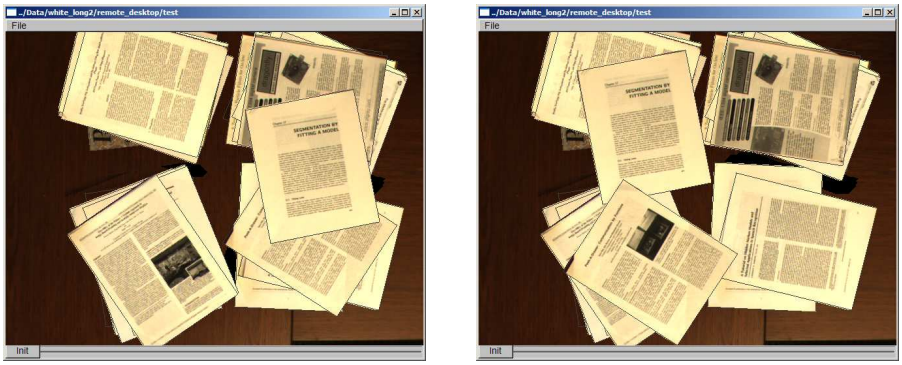
Various solutions have been proposed to bridge the gap between paper and electronic documents by using paper overlaid with specialized patterns and/or special reading devices [3, 5, 18, 1, 6]. However, these approaches require converting to a new physical infrastructure. Moreover, they mainly focus on digitally incorporating paper annotations, and lack the ability to track the document's physical location in stacks. Instead, we present an unobtrusive solution for location tracking that does not involve such a fundamental change to the working environment except for the installation of an overhead video camera.

Tracking and ID technologies such as barcodes, IR tags and RFID tags are already commonplace and becoming more prevalent in the context of finding lost objects [20, 25, 19]. Although these techniques can be applied to paper documents, they all necessitate the use of physical tags and a specialized reader. Furthermore, they are not suitable for accurate tracking of object locations. Vision-based tracking systems [15, 17, 7] avoid the need for special tags and readers, but do not support tracking papers in stacks. More recently, Fujii et al. [4] demonstrated an experimental system for tracking stacked objects using stereo vision. However, as they used the physical height of the stacked objects to detect changes, their technique is not applicable to stacks of relatively thin paper documents.

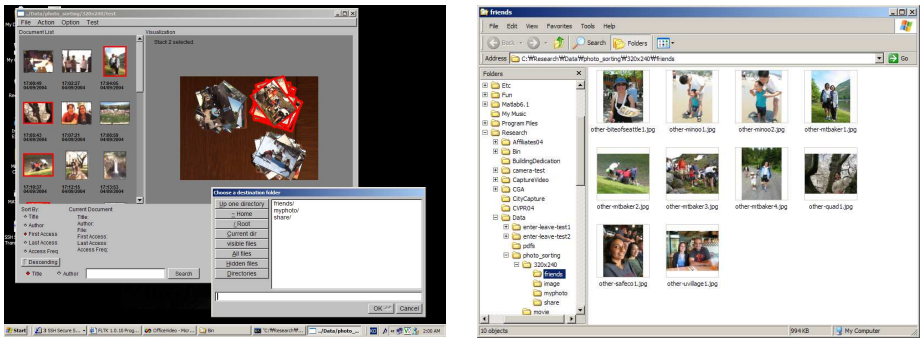
Noticing the easy and intuitive interaction that papers provide, some researchers have explored the use of paper as a tangible user interface to the digital space [16, 11, 8, 23].



(a) Screenshot of the desktop browser interface



(b) The interface in remote desktop mode



(c) The user assigns a stack of photographs to a folder on disk

Figure 3: (a) Screenshot of the desktop browser interface. The user selects a document by either clicking on its thumbnail on the left or performing a keyword search. The view of the desktop on the right expands the stack (green items) and highlights the document in red. (b) Screenshot of the interface in remote desktop mode. Figures show the current state of the desk (left) and a new state after the user moves around the document images to search for a document (right). (c) Screenshot of the interface showing the user select a stack of photographs and assign it to a folder (left). The system copies the corresponding digital photographs into the folder on disk and pops up the folder in thumbnail view (right).

Palette [16] is a paper interface for giving presentations where both human- and computer-readable index cards are used to navigate and control the slide show. In The Designer's Outpost [11], web designers use post-it notes to author web site information architectures in a collaborative setting. Both systems make use of paper props spread out in physical space to drive certain tasks in the digital domain, as in the photo sorting scenario that we demonstrate using our system. However, neither system recovers and utilizes the stack structure of the physical space as our system does. Moreover, our system is able to handle general documents, whereas these systems require special props of known shape and appearance.

Perhaps most closely related to our work is the Self-Organizing Desk [21] which is also a camera-based system for tracking paper documents in stacks. But this system constrains the input in a few important ways, e.g., the papers must be of known size and are only allowed to translate. In earlier work [10], we overcame these limitations. However, that system could only handle distinctive looking objects like multi-colored books, and did not work for papers and other documents with less distinct appearances. In this paper, we present a new framework that incorporates recognition at its core, a key capability that is not supported by either of the above systems. The incorporation of recognition techniques allows us to reliably track visually similar paper documents (i.e., text on white paper) and to link physical documents with their electronic versions on the computer.

SCENARIOS

We focus on the two scenarios, paper tracking and photo sorting. However, we believe our system can also be useful for other applications where physical and electronic documents are used.

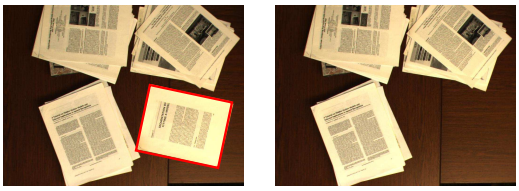
Paper Tracking

In the paper tracking scenario, the user moves around printed documents and books stacked on his desk, and the system records these changes over time. Some sample frames of an example input video are shown in Figure 2 (a). The captured video is subsequently analyzed to recognize each document by automatically matching it with the corresponding electronic document (e.g., PDF). The system also tracks the location of every document in the stacks. Users can then query the system in a variety of ways to find particular documents of interest.

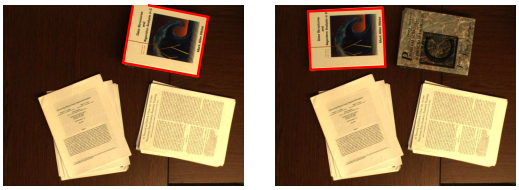
Photo Sorting

The photo sorting scenario is based on the observation that paper provides a very natural interface for sorting photographs. In this scenario, the user prints out a set of digital photographs on paper and sorts them into physical stacks on the desk. Figure 2 (b) shows sample frames from the captured video of one such episode. Our system analyzes the video to identify the photographs and infer the stack structure. The user then associates each stack to a folder on disk. This example makes use of the user's arrangements of physical documents to organize the corresponding electronic documents, demonstrating the potential use of our system as a way to support tangible interfaces for document related tasks.

(a) An entry event



(b) An exit event



(c) A Move event

Figure 4: We model three event types: (a) entry, (b) exit, and (c) move. We have annotated the document that has moved in red. The left and right images correspond to I_{e-} and I_{e+}, images immediately before and after the event e.

INTERFACE

Our interface allows users to take advantage of benefits of both physical and electronic documents.

Desktop Browser Interface

We have developed an interface to support the tasks in each scenario that we call the *desktop browser interface*. Some screenshots are shown in Figure 3 along with descriptions of each element of the interface. The interface provides four different ways to browse the document stacks: visual query, keyword search, sort and remote desktop.

Visual Query To query the location of a particular document on the desk, the user can browse the thumbnail images of the documents discovered by the system, shown on the left panel of figure 3 (a). When the user finds the document of interest and selects it by clicking on its thumbnail image, the visualization of the desk on the right of figure 3 (a) changes to show its location in the stack by expanding the stack containing that document and highlighting the document in red.

Keyword search If the user knows the title or the author of the document, he can perform a keyword search to find it, instead of browsing the thumbnails. The title and author for each paper were manually entered for the results shown in this paper, but these could instead be automatically obtained, e.g., by extracting text from PDF, or parsing XML metadata.

Sort The thumbnails can be sorted based on various criteria, such as author, title and usage statistics to facilitate the

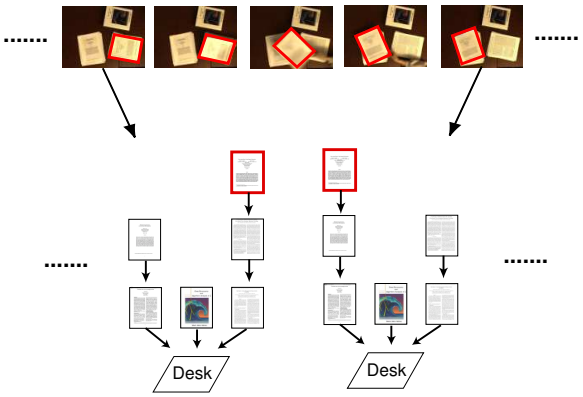


Figure 5: A sequence of scene graphs represent the evolution of the desktop over time. The nodes correspond to documents and edges encode the occlusion relationship between them.

search. For example, the user can sort them in order of their last access time to find recently used, or old documents on the desk.

Remote desktop The user can directly search through the stacks by click-and-dragging on the image of the desk, as shown in Figure 3 (b). We call this mode of interaction the "remote desktop" mode, as it provides a way to search for a document on a desk in a remote location, analogous to the Remote Desktop application on a Microsoft Windows system or the VNC application (http://www.realvnc.com) that allow the user to interact with the electronic desktop of a remote machine. This interface mode can be useful when the user wants to quickly find out what is on the desk from a remote location. The user can also open the electronic version of a document by shift-clicking on its image.

When a document is selected, various information related to the document is displayed, including its title and author, the pathname of the electronic file on disk, and usage statistics, such as the first and last access time, and the total number of accesses.

In the photo sorting scenario, the user can select each stack in the visualization panel by clicking on it and assign a folder, as shown in Figure 3 (c). The system then copies all digital images in the stack into the folder and pops up the folder in thumbnail view.

DOCUMENT TRACKING AND RECOGNITION

In this section, we present a detailed description of how the system tracks and recognizes the documents from the input video. We first give a problem definition, then explain the algorithm used to solve the problem. Note that the input video is processed offline.

Problem Definition

Given an input video of a desktop, the goal of the system is to reconstruct the configuration of documents on the desk at each instant in time. We use the term *event* to refer to a change in the state of the document stacks, and assume that there are three types of events: entry, exit and move. See Figure 4 for examples of each event type. The state of the

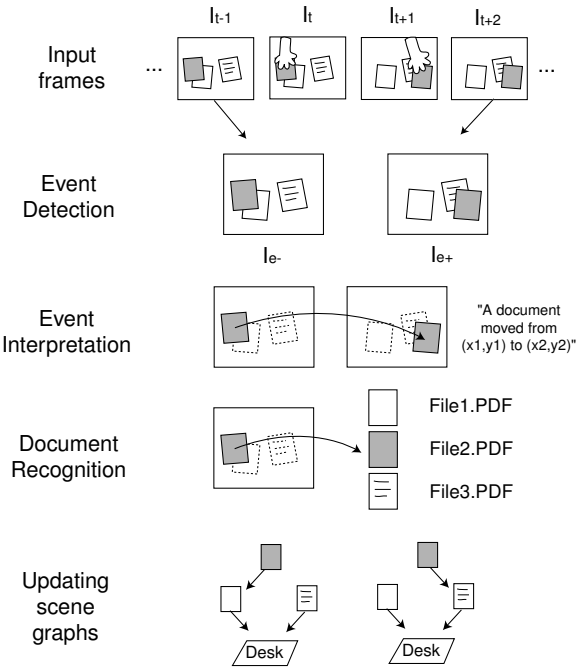


Figure 6: An overview of the document recognition and tracking algorithm.. For each event, we extract a pair of images I_{e-} and I_{e+}, before and after the event. Then, these images are analyzed to determine the type and motion of the event. Next, the document that moved is recognized by matching it with the electronic file on disk. Finally, the scene graph is updated accordingly.

desk is represented by a directed acyclic graph called a *scene graph*, where each node corresponds to a document and edges exist between pairs of documents where one document is directly on top of the other (Figure 5). It is also assumed that each document on the desk has a corresponding image on the computer that is used by the system to match and recognize the document. In the case of papers, images are manually extracted from the PDF file; for books, the JPEG image of the book cover is used; for digital photographs, the image file itself is used.

Assumptions

We make a few simplifying assumptions to make the tracking problem more tractable. All events are assumed to occur to a single document on the top of the stack structure, i.e., only one document may move at a time, and users cannot place or remove documents to or from the middle of a stack. We also assume that each document is unique, i.e., there is no duplicate copy of the same document on the desk. These assumptions somewhat restrict the range of possible user interactions, and generalizing the computer vision techniques to relax these assumptions is an important topic for future work. Nevertheless, these assumptions still allow many useful and natural interactions, which enable the paper tracking and photo sorting scenarios in this paper. Finally, it is important to note that we do not require the desk to be initially empty. Each document is discovered the first time it moves.

Algorithm

The recognition and tracking algorithm works in 4 steps: event detection, event interpretation, document recognition and updating scene graphs. An overview of the algorithm is provided in Figure 6.

Event Detection An event starts with the motion of a document and lasts until the motion ends. To detect events, we first compute frame differences between consecutive input frames. If the difference is large, we assume that an event is occurring. Let e denote an event, and I_{e-} and I_{e+} denote the frames immediately before and after the event, respectively.

Event Interpretation To interpret an event e, we analyze I_{e-}, I_{e+} and frames during the event to determine the type and motion of the event. We use the Scale Invariant Feature Transform (SIFT) [13] to accomplish this goal.

SIFT computes descriptive local features of an image based on histograms of edge orientation in a window around each point in the image. The following characteristics make it suitable for reliable matching and recognition.

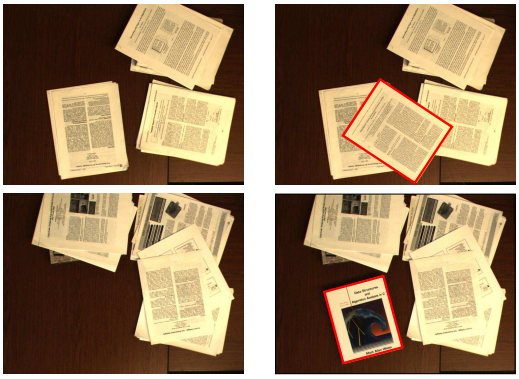
- **Distinctiveness**: its high-dimensional (128-D) descriptor enables accurate differentiation between a large number of features.
- **Invariance to 2D scale, rotation and translation**: features are reliably matched between images of the document in vastly different poses.
- **Robust matching**: detection and matching is robust with respect to partial occlusion and differences in contrast and illumination.

The event is first classified as a move event or otherwise, by looking for a valid motion of a document from I_{e-} to I_{e+}. This is done by matching features between I_{e-} and I_{e+} and clustering the pairs of matching features that have similar motion. If the largest cluster with a non-zero motion contains sufficiently many matches, it is considered a valid motion and the event is classified as a move. The remaining events are either entry or exit events, and we classify them later.

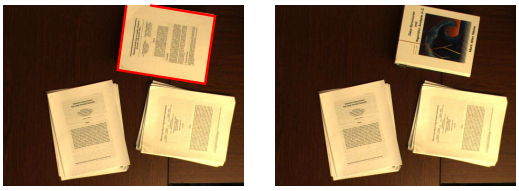
The SIFT features in I_{e-} and I_{e+} are split into two groups *foreground* and *background*, for use in the rest of the procedure. For a move event, features in the largest non-zero motion cluster are considered foreground, and the remaining features background. For remaining events, a feature is background if it does not move across the event, and foreground otherwise.

Distinguishing between an entry and an exit requires running three tests in sequence, described below. We run each test only if the previous test fails.

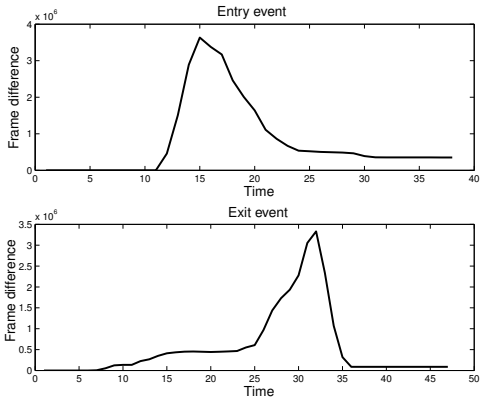
- **Test 1**: Foreground features of I_{e-} and I_{e+} are matched against the image database of electronic documents. For an entry event, if the entering document overlaps with multiple underlying documents or there is no underlying document (Figure 7 (a)), the foreground features of I_{e+} will yield a good match with one document, whereas those of I_{e-} will match either parts of multiple documents or no document (and vice versa for an exit event).



(a) Test 1



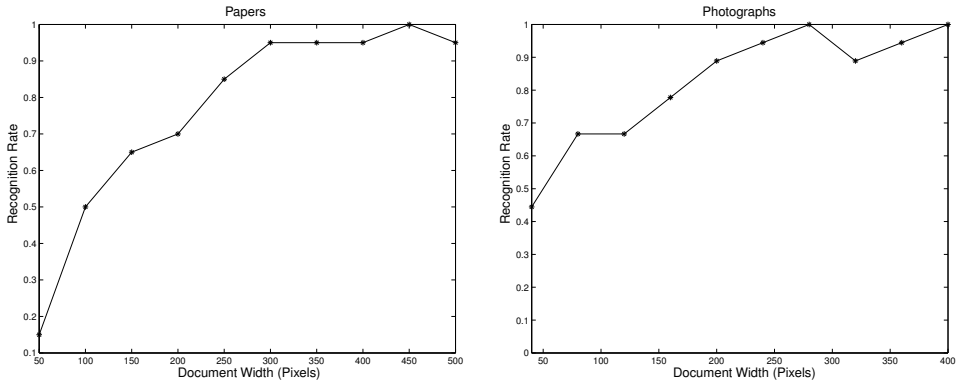
(b) Test 2



(c) Test 3

Figure 7: Three tests are performed in sequence to distinguish between an entry and an exit. (a) Test 1: The entering (or exiting) document overlaps with multiple underlying documents (top) or there is no underlying document (bottom). (b) Test 2: The entering (or exiting) document aligns fairly well with the underlying document, and the system has seen the document beneath that underlying document. (c) Test 3: The system has not seen the document beneath the underlying document, and looks for the peak in the function that measures the amount of motion during the event.

- **Test 2**: If the entering or exiting document aligns fairly well with the underlying document (Figure 7 (b)), Test 1 will fail to classify the event. However, if the system has previously seen what lies under the foreground region of I_{e-}, it can compare the new foreground region of I_{e+} with that underlying document. If they match, it is an exit event; otherwise, it is an entry.
- **Test 3**: Finally, if the system does not have sufficient knowledge about the current stack structure to perform



(a) Paper documents (b) Photographs

Figure 8: Plots of document recognition rate for (a) paper documents and (b) photographs under varying image resolution. The Y axis represents the percentage of correctly recognized documents, and the X axis represents the width of the document (i.e., the length of the longer side in pixels) in the captured image.

Test 2, the input frames during the event are analyzed to determine the event type. There is an asymmetry in the amount of changes in the image between an entry and an exit. During an entry event, both the user's hand and the new document enters the scene in the beginning and only the hand exits in the end, therefore causing more changes in the beginning than the end, whereas the reverse is true in an exit event. Therefore, the system classifies the event based on the peak location in the function of the amount of motion over time, measured by differencing each frame with I_{e-} outside the region occupied by the entering/exiting document, as shown in Figure 7 (c).

Document Recognition Once the event is interpreted, the foreground SIFT features of I_{e-} (I_{e+} for an entry event) are matched against the features of each image of electronic documents on the computer and clustered according to the relative transformation. The matching score is defined as the ratio of the sum of matching scores for the features in the largest cluster to that of all matching features. The document with the best matching score is considered the matching document. We assume that all documents have enough features to perform reliable matching between the physical and electronic copy.

Updating Scene Graphs The interpreted event is used to update the current scene graph representing the structure of document stacks on the desk. Initially, the scene graph is empty, and new nodes are added as new documents are discovered. If the current event is the first event for the document, a new node representing that document is introduced into all scene graphs up to that point, and new edges are added to connect the new node to all scene graphs.

For an exit, all edges are disconnected from the node representing the exiting document. For an entry event, new edges are introduced between the entering document and all documents directly under it. For a move event, these two steps, i.e., exit and entry, are performed in sequence.

RESULTS AND DISCUSSION

In this section, we discuss our results and present a performance analysis on document recognition. See our web site (http://grail.cs.washington.edu/projects/office) for a video demonstrating the results.

Experimental Setup and Input Sequences

We used the Dragonfly video camera from PointGrey Research, Inc. that records 1024×768 images at 15 frames per second. We streamed the video frames to memory using the firewire port on a PC. The paper tracking sequence was recorded over approximately 40 minutes. It contained 49 events in total (27 moves, 9 entries and 13 exits). There were 20 printed paper documents and 2 books in the sequence. The photo sorting sequence was recorded over approximately 10 minutes, with 30 events in total (11 moves, 19 entries and no exits). There were 30 photographs in the sequence, all of which were printed on paper sheets of almost identical size (approximately 6×4 inches). Most of them contained a mixture of people and landscape. The user distributed photographs from two source stacks, one held in her hand and the other on the desk, into three target stacks. These input sequences were processed offline after the recording session was over.

Event Classification

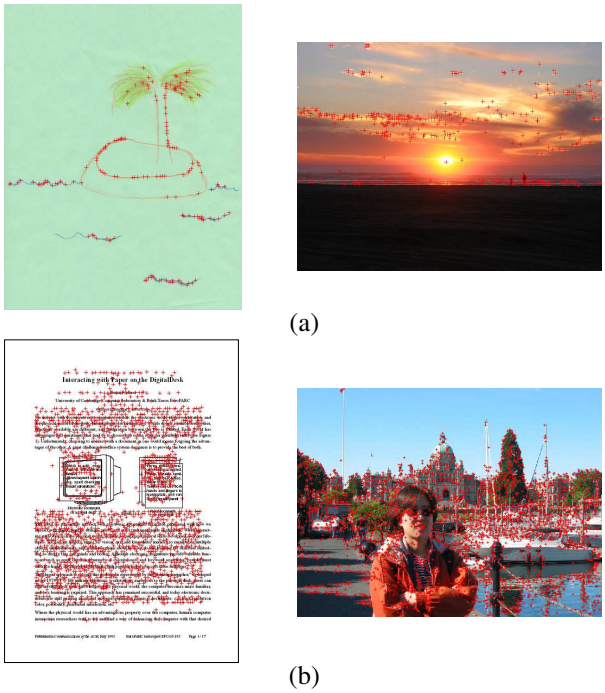
The event classification method described in the Algorithm section had a 100% success rate on the two input sequences. The move vs. entry/exit classification test worked in all cases. For entry and exit events, tests 1, 2 and 3 were conducted in sequence, and all of these events were classified correctly. Because each of the three tests handles different situations, all three are required for a perfect classification. Tests 1 and 2 succeeded on 14 out of 22 entry and exits in the paper tracking sequence and on all 19 entry and exits in the photo sorting sequence. Only the 8 remaining entry and exits in the paper tracking sequence required the use of test 3. To evaluate test 3, we performed this test on all entry and exit events in the two sequences. It failed on 1 out of 22 cases and 1 out of 19 cases, respectively, showing that by itself it is a fairly reliable method for distinguishing entry and exit events.

The paper tracking sequence contained 22 documents including printed paper documents and books, and all of

them were recognized correctly against a database of 50 documents. The database included not only the cover page of a document, which usually has a more distinct text layout than the rest of the document, but also internal pages, some of which contained only text. The images of electronic documents in the database were approximately 400×500 pixels (width×height), and the captured images of the documents were approximately 300×400 pixels.

The photo sorting sequence contained 30 photographs. In the input video, the photographs were approximately 300×400 pixels. There were 50 image files in the database, with resolutions varying between 640x480 and 901x676. Many of them had people posing in front of a background landscape, and some of them contained only scenery. All 30 photographs were recognized correctly against the database.

We conducted a simple test to further analyze the performance of document recognition based on SIFT feature matching. We took pictures of approximately 20 documents and 20 photographs with varying number of detected features, and tried to match them against a database of 162 paper documents and 82 photographs, respectively. We also varied the resolution of the captured image, to examine the effect of the image resolution on the recognition performance.



(a)

(b)

Figure 9: (a) Documents with too little texture that our recognition technique could not handle reliably: a simple drawing (left, 660x773 pixels, 248 features) and a picture of sunset (right, 800x600 pixels, 508 features). SIFT features are overlaid as red cross marks. (b) Documents with average numbers of features for comparison: a research paper (left, 435x574 pixels, 2509 features) and a picture of a person in front of complex scenery (right, 819x614 pixels, 4198 features).

Document Recognition

The recognition rate increased in proportion to the image resolution, as shown in Figure 8. It can be seen that papers must be at least 230×300 (all papers were letter size) and photographs 150×200 pixels (all photographs had 4:3 aspect ratio) in the captured image to achieve a recognition rate of 90%. The recognition rate does not reach 100% even at fairly high resolutions, because a couple of documents had too few features to be reliably recognized (Figure 9 (a)).

The images in the database also had a varying number of SIFT features, ranging from 248 to 8409 for papers and from 35 to 9526 for photographs. We found that the recognition performance is not significantly affected by the number of features, except for the two cases with extremely few features. This is an expected result because the matching score is normalized with respect to the total number of features on the document, as described in the Algorithm section. It shows that our document recognition method can be successfully applied to a wide range of document resolution and numbers of features.

CONCLUSION AND FUTURE WORK

We have presented a novel computer vision system for recognizing and tracking documents on the desk with a user interface that allows the user to browse the document stacks and query documents of interest. We demonstrated how our system enables two scenarios, paper tracking and photo sorting. Our system provides a seamless unification of physical and electronic desktops, without requiring a new physical infrastructure except for a video camera.

We envision several directions to extend the current work in the future. First, we believe that some of the simplifying assumptions can be relaxed to handle more realistic desktops. For instance, people often shift stacks of documents together rather than moving individual documents one at a time. Also, it is common to have duplicate copies of the same document or documents of similar appearance, such as different versions of a document undergoing revision. Such situations are challenging because of the uncertainties they incur in reasoning about the events. To deal with these uncertain situations, it would be desirable to have a multiple-hypotheses tracking mechanism. Allowing the system to solicit feedback from users can also help the system to correct mistakes. Another natural extension is handling documents without sufficient features on the surface and documents that do not have matching electronic versions in the database. Also, we believe there is room for speeding up the computation, possibly making the system perform in real-time. A more thorough analysis of the system's performance and failure modes under various situations would be worthwhile.

There are other useful types of interactions that can be added to the current user interface. For example, one easy way to look up information related to a physical document is simply "showing" the document to the camera. The system can then recognize the document and display relevant information. Also, if the system can detect changes on the document surface as users make written annotations on documents, the written annotation may be automatically "lifted" by the computer vision system and incorporated into the electronic

version of the document. A user study on how people actually interact with documents on the desk can help us determine the types of user tasks that can benefit from our system.

Finally, we believe that our framework can be applied to other domains that may also benefit from a video-based tracking and recognition system. Some examples are bookshelves (e.g., library, bookstore), CD/DVD racks, bulletin boards, laboratories, warehouses, and kitchen counters.

ACKNOWLEDGMENTS

We would like to thank Li Zhang for his help in preparing the companion video. This work was supported in part by National Science Foundation grant IIS-0049095 and Intel Corporation.

REFERENCES

1. T. Arai, D. Aust, and S.E. Hudson. Paperlink: a technique for hyperlinking from real paper to electronic content. In *Proc. of CHI*, pages 327–334, 1997.
2. T. Arai, K. Machii, S. Kuzunuki, and H. Shojima. Interactivedesk: A computer augmented desk which responds to operations on real objects. In *Proc. of CHI*, pages 141–142, 1995.
3. M. Dymet and M. Copperman. Intelligent paper. In *Proc. of Electronic Publishing*, pages 392–406, 1998.
4. K. Fujii, J. Shimamura, K. Arakawa, and T. Arikawa. Tangible search for stacked objects. In *Proc. of CHI*, pages 848–849, 2003.
5. F. Guimbretière. Paper augmented digital documents. In *Proc. of UIST*, pages 51–60, 2003.
6. J.M. Heiner, S.E. Hudson, and K. Tanaka. Linking and messaging from real paper in the paper pda. In *Proc. of UIST*, pages 179–186, 1999.
7. H. Hile, J. Kim, and G. Borriello. Microbiology tray and pipette tracking as a proactive tangible user interface. In *Proc. of the 2nd Int. Conf. on Pervasive Computing*, pages 323–339, 2004.
8. W. Johnson, H. Jellinek, J. Leigh Klotz, R. Rao, and S.K. Card. Bridging the paper and electronic worlds: the paper user interface. In *Proc. of CHI*, pages 507–512, 1993.
9. A. Kidd. The marks are on the knowledge worker. In *Proc. of CHI*, pages 186–191, 1994.
10. J. Kim, S.M. Seitz, and M. Agrawala. The office of the past: Document discovery and tracking from video. *IEEE Workshop on Real-Time Vision for Human-Computer Interaction*, 2004.
11. S.R. Klemmer, M.W. Newman, R. Farrell, M. Bilezikjian, and J.A. Landay. The designer's outpost: A tangible interface for collaborative web site design. In *Proc. of UIST*, pages 1–10, 2001.
12. H. Koike, Y. Sato, and Y. Kobayashi. Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system. In *ACM Trans. on Computer-Human Interaction*, pages 307–322, 2001.
13. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Jour. of Computer Vision*, 60(2):91–110, 2003.
14. W.E. Mackay and D. Pagani. Video mosaic: Laying out time in a physical space. In *ACM Multimedia*, pages 165–172, 1994.
15. D. Moore, I. Essa, and M. Hayes. Object spaces: Context management for human activity recognition. In *Proc. of the 2nd Annual Conf. on Audio-Visual Biometric Person Authentification*, 1999.
16. L. Nelson, S. Ichimura, E.R. Pedersen, and L. Adams. Palette: A paper interface for giving presentations. In *Proc. of CHI*, pages 354–361, 1999.
17. R. Nelson and I. Green. Tracking objects using recognition. In *Int. Conf. on Pattern Recogntion*, pages 1025–1030, 2002.
18. M.C. Norrie and B. Signer. Switching over to paper: A new web channel. In *Proc. of 4th Int. Conf. on Web Information Systems Engineering*, pages 209–220, 2003.
19. R.E. Peters, R. Pak, G.D. Abowd, A.D. Fisk, and W.A. Rogers. Finding lost objects: Informing the design of ubiquitous computing services for the home. Technical Report GIT-GVU-04-01, Georgia Institute of Technology, College of Computing, GVU Center, Jan 2004.
20. J. Rekimoto and Y. Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. In *Proc. of Designing Augmented Reality Environments (DARE)*, pages 1–10, 2000.
21. D. Rus and P. deSantis. The self-organizing desk. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, pages 758–763, 1997.
22. A.J. Sellen and R.H.R. Harper. *The Myth of the Paperless Office*. The MIT Press, Cambridge, Massachusettes, 2002.
23. L. Stifelman, B. Arons, and C. Schmandt. The audio notebook: paper and pen interaction with structured speech. In *Proc. of CHI*, pages 182–189. ACM Press, 2001.
24. N. Takao, J. Shi, and S. Baker. Tele-graffiti: A camera-projector based remote sketching system with hand-based user interface and automatic session summarization. *Int. Jour. of Computer Vision*, 53(2):115–133, 2003.
25. R. Want, K.P. Fishkin, A. Gujar, and B.L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proc. of CHI*, pages 370–377, 1999.
26. P. Wellner. Interacting with paper on the DigitalDesk. *Comm. of the ACM*, 36(7):86–97, 1993.