

The Self-Organizing Desk

Daniela Rus Peter de Santis

Department of Computer Science
Dartmouth College
Hanover, NH 03755

{rus,gwitto}@cs.dartmouth.edu

Abstract

The self-organizing desk is a system that enhances a physical desk-top with electronic information. It can remember, organize, update, and manipulate the information contained in the documents on a desk. The system consists of a simple robot eye that can survey the desk, a module for smart extraction of information from the images taken by the robot, a module for representing this information in multiple views, and a module that allows a user to interact with this information.

1 Introduction

We wish to create smart physical worlds, that can augment reality with electronic information. Such spaces will keep track of their own contents, indexing and organizing their objects in electronic views. We hope to achieve this vision by using sensors to extract information about the physical world. The self-organizing desk, a system that can keep track of its contents autonomously, is an example of such systems. Consider the flow of paper that arrives for processing at someone's desk. Many paper documents potentially relevant to planning and scheduling arrive every day. These documents are filtered and filed in filing cabinets, or in random piles. In this space one has to manage queries such as "where is the letter from John Hopcroft?", "where is the paper I received last week that has a red table in the upper right corner?", "where is all the information I need to complete this report?", etc. Answers to these queries should be of the form "the top, right pile, of the desk, about halfway down". Such query processing would be much easier if all the data on the desk-top were available electronically. The self-organizing desk described in this paper implements these ideas.

Systems capable of self-organization present their users with accurate summaries at various levels of de-

tail, even when the data changes dynamically. Self-organization is implemented by using sensors to observe and extract information about the objects in a physical space. The sensor-data is captured, indexed, and organized in electronic views. The views are aggregated as a database and stamped with spatial and temporal information. In selecting electronic views, our premise is that textual cues, visual cues, and topic-subtopic relationships are equally important for locating and manipulating documents.

In this paper we describe a system that implements the self-organization metaphor on the documents on a desk-top. The self-organizing desk uses a robot eye (a steerable, 3 degree of freedom computer control camera) to survey the documents on a desk-top, to capture and index their attributes (words, color, images, tables, location, time, etc.). The system interacts with users through a GUI to help them locate items on the piles on the desk top, or visualize conceptually the content of the desk. It supports operations to add a document to a random location on the desk, to remove a document from the desk top, and to shift a stack of documents from one desk-top location to another.

The paper is organized as follows. First, we present the system that implements the self-organization metaphor for a desk. Second, we describe our experiments with the desk. Finally, we discuss related work and future extensions.

2 The System Description

The architecture of the self-organizing desk system is shown in Figure 1. The system uses a camera that surveys a desk top by sweeping above its surface continuously, looking for changes. We assume that the only objects on the desk are standard size papers¹

¹The presence of other objects for example coffee cups, does not affect the performance of the system. At the moment such objects are simply not recognized or processed by the system.

and that these papers undergo changes one at a time. The change can be one of the following: *add a new paper, remove a paper, or shift a stack of papers from one location to another*. Change is detected by the camera via the segmentation module. Each time the system notices a change, the following set of operations are executed. Suppose, for simplicity, that the system has noticed a new paper that was added to the desk. The approximate coordinates of the paper are computed (by the segmentation module) and the camera is automatically positioned (by the camera control module) so as to capture a picture of maximum detail of the new document. The resulting image is passed through a variety of filters including OCR, color, and tables and the filtered data is indexed in a database. In addition to layout, the database also contains space, time, and history information for each document. History captures the order in which the documents arrive and is used for defining stacks (or piles) of documents. This database is indexed, updated, organized and summarized on-line (as its contents changes dynamically) by the information access module. The user can query this module by specifying full text, visual attributes, and/or requests for content summaries, content organization, and visualization. Queries are answered in the form of a GUI that points the user to the location of the relevant documents.

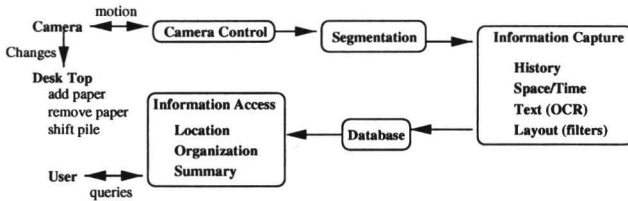


Figure 1: The system components of the self-organizing desk. The camera surveys a desk top for changes. The images taken by the camera are segmented and filtered for multiple representations in a database. The database is indexed, organized, and available for queries.

The following sections describe these modules in greater detail.

2.1 Surveying the Desk

The desk-top surveillance system consists of a Canon VC-C1 camera with zooming capabilities that is mounted on a pan-tilt computer steerable platform. The camera is connected to a Silicon Graphics Indy

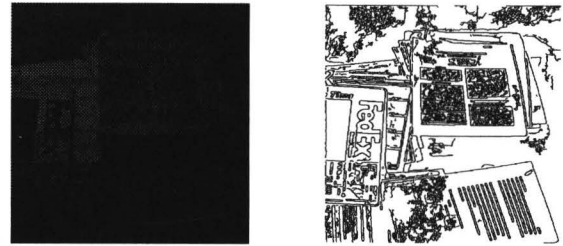


Figure 2: The left image shows a desk-top picture taken with an Indy camera. The right image shows the result of running a Canny edge detector of the left image. Indy cameras are very low-resolution and inexpensive. Yet, they provide enough detail to be useful for surveying certain features.

(SGI) that controls its operation. Our initial experiments established that the manufacturer's motion control for this camera is inadequate for two reasons: it is inaccurate and inflexible. Instead of using the commercially supplied controller we developed an expressive interface and accurate control. The following operations are currently supported: (1) pan or tilt along the entire sweeping range, (2) pan or tilt to a specific location, (3) zoom to a specific location, and (4) capture the image.

2.2 Extracting Documents from Images

Currently, it is near-impossible to extract conceptual information about arbitrary three dimensional objects that are subject to noise and occlusion. However, since we restrict the objects in our application to be standard paper documents, we can develop effective segmentation algorithms. Consider Figure 2. It represents the image of a desk-top taken with a very low-resolution camera and the corresponding edge-detected image. The edge-detected image has significant noise but enough detail that can be detected and parsed automatically.

One important feature of a segmentation algorithm for this application is robustness in the presence of noise. Noise arises naturally here in two ways. Every time a human touches a paper on the desk, its position may be shifted by a small amount. The segmentation algorithm should tolerate this. Second, when papers are stacked on top of one another as in Figure 2(right) the edges of the stack are not perfectly aligned. The segmentation algorithm should find one set of edges only to approximate the location of the stack.

We have developed a statistical segmentation algorithm that has noise tolerance built-in. This algorithm

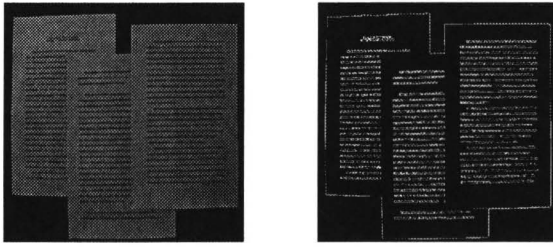


Figure 3: The left image shows a desk-top image taken with our Cannon VC-C1 camera in our experimental setup. The right image shows the result of running our statistical border-finder on the left image.

has four steps. Step 1 compares the base image taken when the last event was detected against the current image. When a change is detected, the area of change is extracted from the image. Step 2 generates a new image of maximal detail that contains the entire area identified in Step 1. This is accomplished by aiming and zooming the camera. The correct camera motion (pan/tilt/zoom parameters) is computed by using the absolute camera and desk space coordinates, and the relative coordinates of the area of interest in the space. Step 3 identifies the enclosing border of the area of interest. This algorithm relies on object features and uses statistics on pixels to recognize and enclose the page boundaries. The basic idea is to do a walk along pixels that identifies a simple polygon, called the border polygon. The greatest challenge of this algorithm is noise: for the case of a desk-top, any form of object manipulation is likely to displace the paper by a small amount. Our algorithm uses statistics to address the stop problem by fitting a line through the pixels in an incremental fashion. Figure 3 shows a snapshot from the execution of this algorithm. Finally, the fourth step of the algorithm takes the border polygon and parses it to identify the pages. For the self-organizing desk application, this reduces to finding a cover of this polygon by rectangles, that is consistent with the document history. In our current implementation we assume that the documents on the desk top can only be translated relative to each other. We use this assumption to determine the vertices of a new document when the difference between the current image and the previous image returns 1, 2, or 3 vertices only. This situation arises when the new document overlaps existing documents on the desk.

2.3 Capturing Electronic Information about the Desk

The output of the border detection and document extraction algorithm consists of the relative coordinates of the page within the physical space. These coordinates are used to compute a camera configuration that can capture a new image of the object of interest, of maximal detail. This new image is ready for information extraction. A multitude of features is useful for defining attributes users can query on: arrival time of the document, physical location of the document, textual content, figures, color, tabular data, and layout content (for instance, is the document a letter?). Visual and layout cues like tables, color, and figures, complement the textual content of documents and they play an equally important role in information access.

The self-organizing desk supports queries that combine textual with layout information through multiple representations, each representation corresponding to an attribute detectable by a filter. This information extraction is carried by a library of smart filters. Our current library consists of three filters: OCR, color, and tables, but the architecture is expandable and we will add new filters in the future.

OCR. We have investigated the use of text filters (OCR systems) on camera images. We found that commercial OCR systems do not work well on images because the resolution is not high enough. Our approach was to enhance the resolution by dividing the image into n subimages (n is determined by the desired OCR accuracy), doing a linear interpolation for each pixel in the resulting images, and feeding the resulting images to the OCR system. This method improved the performance of the OCR system from 60% to 95% in character recognition accuracy.

Color. The color filter works by building a color histogram annotated with layout information for each object. The filter determines the 24 most prevalent colors occurring in the document and the location of each color. Location is a layout attribute determined by placing a 3×3 grid on the paper.

Space, time, and history. Each paper gets assigned a location on the desk by using the coordinates computed by the camera. Each paper is also time stamped. In addition, each paper is associated with a list of papers above it and papers below it, to capture the history of the desk-top. This information is necessary to implement the stack-shift operation. It

is also necessary to estimate the location of a document on the desk if the document is covered by other documents.

2.4 Searching, Organizing, and Visualizing the Desk

The filters defined in Section 2.3 generate a web of representations for each document. We compile this multiplicity of representations in a database. In this database we also capture temporal information for each document (its arrival time on the desk), spatial information (its location on the physical desk), and history information (if the paper is in a pile, which documents are above it, and which documents are below it.) This database supports the following desk operations: adding a paper to a random location on the desk, removing a paper from the desk, and re-locating a paper or a stack on papers on the desk. The information in this database changes dynamically, as driven by these operations. In response to each event, the database is updated automatically.

The database comprises a collection of inverted indices, one for each attribute. An inverted index associates each attribute instance with a list of documents containing it. The advantage of this representation scheme is a speed-up in search: given a specific attribute (an word, a color, *etc.*), the list of documents containing this attribute is available in constant time.

We implemented several methods for searching, organizing, and visualizing the contents of this database.

Search. The self-organizing desk can be queried with keywords, with an entire document (full text), with color and layout information, with table information, and any boolean combination of these attributes. We use an augmented version of the *Smart System* [Sal91], which is a sophisticated text-retrieval system. We augmented Smart to also support color, layout, and table indices. Smart copes well with partially corrupted (by OCR) text. Its basic premise is that two documents are similar if they use the same words. Documents and queries are modeled as points in a *vector space* defined by the important words occurring in the corpus. When all texts and text queries are represented as weighted vectors, a similarity measure can be computed between pairs of vectors that captures the text similarity. We use this similarity measure as the basis for computing hyperlinks between documents that are similar to each other in this statistical framework.

Organization. The textual information contained in the documents on the desk is organized by topic

using our own clustering algorithm (called the star algorithm) on the vectors in the document space. Our implementation uses a modification of the Smart system and the underlying cosine metric. The star algorithm gives a hierarchical organization of a collection into clusters. Each level in the hierarchy is determined by a threshold for the minimum similarity between pairs of documents within a cluster at that particular level in the hierarchy. This method conveys the topic-subtopic structure of the corpus according to the similarity measure used. The star algorithm is accurate in that it produces dense clusters that approximate cliques with provable guarantees on the pairwise similarity between cluster documents, yet are computable in $O(E)$, where E is the number of edges above threshold in the document collection². The documents in each cluster are tightly inter-related and a minimum similarity distance between all the document pairs in the cluster is guaranteed. This resulting structure reflects the underlying topic structure of the data.

Summaries. A topic summary for each cluster computed by the star algorithm is provided by the center of the underlying star for the cluster.

Visualization. We developed a visualization method for organized data that presents users with three views of the data: a Euclidean projection of the documents on the planar representation of the desk, a graph that shows the similarity relationship between all the documents, and a graph that shows the topic structure of the desk. The views allow users to select objects with a mouse and they are connected in that when an object is selected in one view, it gets highlighted in the other views. For instance, the user may select a cluster, which will highlight the areas of the desk where the documents in the cluster are located.

3 Experiments

We implemented all the modules described in Section 2 and constructed the following experiment. The VC-C1 camera is mounted on the tripod and connected to an SGI. The OCR package we use is EasyRead (which is a version of ScanWorks ported to SGIs.) The desk-top is set up vertically, on a wall. Documents are added/removed by pasting/removing them on the wall. Users interact with this system by physically moving papers, by entering queries in the GUI (see Figure 4), and by observing the visualization of the results returned by the search engine.

² E is at most N^2 , where N is the number of documents.

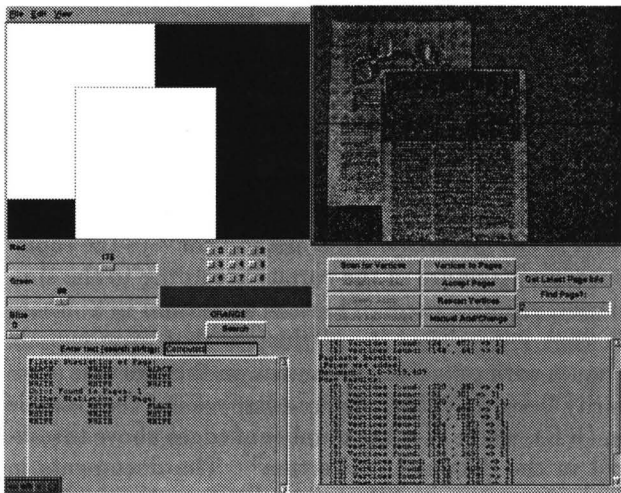


Figure 4: This figure shows the GUI to the desk. The top right section contains the actual image captured by the camera. The top left section contains the representation of the desk-top computed by the system. The bottom half portion of the GUI contains buttons for specifying attributes and queries and for displaying results.

We have repeated the following two experiments over fifty times recently. In the first experiment, three papers were added sequentially to the desk. The second paper was placed to overlap the first paper and the third paper was placed to overlap the previous two. In the second experiment we created a desk top by adding three papers sequentially. The first paper was placed randomly. The second paper was placed as in the first experiment. The third paper was placed approximately above the first paper so as to make up a stack. The addition of each paper was noticed by the camera, which triggered the main loop through the self-organization system. The location of each paper was computed by the segmentation module, the camera was then zoomed to capture the document location, the extracted images was passed through the OCR filter and the color filter. The results of these operations were entered in the database along with the space, time, and history of the document and the database was updated with this information. The database was then organized by content and summarized using the topic clustering and summarization algorithms. The experimental data for the segmentation and OCR aspects of the system is presented in Figure 5. Our human user input the following queries: "orange in upper right corner and distributed mobile robotics", "green

Task	Tries	Success	Reliability
Segmentation	50	45	90 %
OCR	50	50	90 % avg.

Figure 5: This table contains reliability data for the self-organizing desk operations. The first line shows accuracy of the segmentation algorithm whose task is to identify the papers. OCR is measured as average character recognition rate per run.

anywhere", and "manipulation". For all these queries the relevant documents were identified and the user was given pointers to their physical location of the desk. The only query failures we observed are for keyword searches on words corrupted by OCR.

The system takes approximately 30 seconds to determine a change on the desk. The enhanced OCR scheme (that processes and then merges 9 individual images per page) takes approximately 15 minutes, the initialization and update of the database takes negligible time (less than 1 second), and the queries take negligible time (less than 1 second.)

The failures of the system are due to the camera and the OCR. The current camera resolution and the current implementation of the segmentation algorithm make it impossible to distinguish corners that are closer than 1cm to each other. OCR is too slow. The character recognition is poor even with our enhancements. This problem could be solved by using a higher-resolution camera and a better OCR package.

4 Related Work

Efforts to enhance physical environments with electronic information include The Intelligent Room project at the AI Lab at MIT and the ALIVE project at the Media Lab at MIT. The goal of the Intelligent Room project is to create a room surveyed by cameras that can recognize and understand physical gestures. Progress on this project has been reported in [Tor95]. The ALIVE project allows users to interact with animated electronic characters and has been described in [Mae95]. Other related projects include efforts from Euro Xerox and Hitachi to create interactive desks, where the user can write with a stylus pen on the desk top. The desk-top consists of a display that can capture the user's input. A camera mounted on the desk top is used to project on the desk top rather than extract information [AM+95].

The self-organizing desk draws from progress made in several areas: self-organizing systems [Koh90, CKP93], information retrieval and organization [Sal91, RA95], robotics and vision [MRR96,

HKR93], automated document structuring [TA92, RS95b, NSV92], and user interfaces [CAC93].

5 Discussion

We have described a system that implements the self-organization metaphor to enhance a physical space with electronic information. This system uses a number of key technologies in robotics, computer vision, OCR, information retrieval, filtering, and organization, and integrates them into a system that solves a new application. Our experiments demonstrate the feasibility and power of enhancing a desk-top with the electronic information extracted from the objects on the desk top. Applications of the technology demonstrated by this work reach beyond the self-organizing desk, into the area of using smart sensors for information processing in augmented reality. Candidates are dynamically changing physical space whose objects are simple enough that segmentation and feature extraction is possible. The smart sensors will support indexing, searching, and organizing the physical space. One immediate extension is a self-organizing bookcase.

The biggest computation sink in this system is the image segmentation module and the OCR. OCR performs poorly on camera-extracted images. We bypassed this problem by scanning images. We are currently building a self-organizing filing cabinet where documents are actively scanned in before being filed away. Our preliminary tests show that this system is fast and more reliable. In parallel with this effort, we are also developing better image segmentation tools for desk tops, smarter filters, and better GUIs.

Acknowledgements

The camera control package was implemented by Scott Silver. The first version of the image segmentation package was implemented by Michael Taylor. The color filter and GUI was implemented by Hans Kieserman and the information organization system was implemented by Katya Pelekhov. We thank Carlo Tomasi for the initial discussion on image segmentation in the presence of slop.

References

- [AM+95] T. Arai, K. Machii, S. Kuzunuki, and H. Shojima, InteractiveDESK: A computer augmented desk which responds to operations on real objects, in *Proceedings of Computer Human Interactions*, 1994.
- [CKP93] D. Cutting, D. Karger, and J. Pedersen. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th SIGIR*, 1993.
- [HKR93] D. Huttenlocher, G. Klanderman, and W. Rucklidge, Comparing images using the Hausdorff distance, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [Koh90] T. Kohonen, The self organizing map, in *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [Mae95] P. Maes, Artificial Life meets Entertainment: Interacting with Lifelike Autonomous Agents, Special Issue on New Horizons of Commercial and Industrial AI, Vol. 38, No. 11, pp. 108–114, *Communications of the ACM*, ACM Press, November 1995.
- [MRR96] R. Manmatha, S. Ravela, and E. M. Riseman, Retrieval from Image Databases Using Scale Space Matching, in *Proceedings of the ECCV '96*.
- [NSV92] G. Nagy, S. Seth, and M. Vishwanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7), 1992.
- [RA95] D. Rus and J. Allan. Structural queries in electronic corpora. In *Proceedings of DAGS95: Electronic Publishing and the Information Superhighway*, May 1995.
- [RS95b] D. Rus and K. Summers. Using whitespace for automated document structuring. *Advances in digital libraries*, eds. N. Adam, B. Bhargava, and Y. Yesha, Springer-Verlag, LNCS 916, 1995.
- [Sal91] G. Salton. The Smart document retrieval project. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 356–358.
- [Tor95] Mark C. Torrance, Advances in Human-Computer Interaction: The Intelligent Room, in *Working Notes of the CHI 95 Research Symposium*, 1995.
- [TA92] S. Tsujimoto and H. Asada. Major components of a complete text reading system. In *Proceedings of the IEEE*, vol. 80, no. 7, 1992.
- [Tur90] H. Turtle. Inference networks for document retrieval. PhD thesis. University of Massachusetts, Amherst, 1990.
- [CAC93] Special issue on visualization. *Communications of the ACM*, 36(4), 1993.