BRITISH COUNCIL PROGRAMME UNIVERSITY OF NIZHNY NOVGOROD

INTERNATIONAL NETWORKING FOR YOUNG SCIENTISTS

# UK-RUSSIA WORKSHOP
# “PROACTIVE COMPUTING”

4th–6th February 2005

*University of Nizhny Novgorod*

# *PREPROCEEDINGS*

# *PROGRAMME*

| | | |
|---|---|---|
| 3rd Thursday | | Evening arrival. Participants from UK. Meeting at the airport of Nizhny Novgorod<br>Accommodation *Oktyabrskaya Hotel*<br>9a, Verkhne-Volzhskaya Naberezhnaya Str., Nizhny Novgorod, Russia<br>Phone: +7 (8312) 32-11-24, Fax: +7 (8312) 32-05-50 |
| 4th Friday | | Morning arrival. Accommodation Oktyabrskaya Hotel |
| | 11:00 – 12:00 | *Workshop opening* UNN, Building 2, Room 411<br>*Roman G. Strongin* (UNN) Welcome from rector of UNN<br>*Irina Kouklina* Welcome from the British Council |
| | 12:00 – 13:00 | *Peter Robinson* (Cambridge University, UK) Video User Interfaces (invited lecture) |
| | 13:00 – 14:00 | LUNCH |
| | 14:00 – 16:00 | Visiting Intel Russia Research Center<br>*Valery Kuriakin* (Intel) INNL and Intel Russia General Presentation<br>*Ludmila Nesterenko* (Intel) Intel Academic Activities<br>*Vadim Pisarevsky* (Intel) OpenCV and Performance Libraries<br>*Victor Eruhimov* (Intel) Proactive Computing in IRRC |
| | 16:30 – 19:00 | Getting acquainted round table and Welcome party |
| 5th Saturday | | Presentations. Room 411 |
| | 10:00 – 11:30 | *Mathew Chalmers* (Glasgow University) Equator: Mixing Media and Showing Seams<br>*Denis Ivanov, Victor Lempitsky, Anton Shokurov and Yevgeniy Kuzmin* (Moscow State University) Computer Vision in CGG MSU |
| | 11:30 – 12:00 | COFFEE-BREAK |

| | | |
|---|---|---|
| | 12:00 – 13:30 | *Maja Vukovich* (Cambridge University) Proactive Service Composition<br>*Nikolai Zolotykh* (University of Nizhni Novgorod) Exact Learning Geometrical Objects |
| | 13:30 – 14:30 | LUNCH |
| | 14:30 – 16:00 | *Quentin Stafford-Fraser* (Newnham Research, Cambridge) How Content Indexing affects User Interfaces<br>*Victor Eruhimov and Igor Chikalov* (Intel) Adaptive Algorithm for Anomaly Detection in Network Environment |
| | 16:00 – 16:30 | COFFEE-BREAK |
| | 16:30 – 18:00 | *Alastair Beresford* (Cambridge University) Towards Automated Computation Sharing for Ubiquitous Computing<br>*Sergey Belov and Victor Gergel* (University of Nizhni Novgorod) Parallel Algorithms for Probabilistic Expert Systems |
| | 19:00 – 20:00 | DINNER (Hotel Oktyabrskaya) |
| | 20:00 | Free time for discussions |
| 6th Sunday | 10:00 – 11:30 | Presentations<br>*Stephen Brewster* (Glasgow University) Multimodal Interaction and Proactive Computing<br>*Vyacheslav Shkodyrev* (Saint Petersburg Technical University) Information-Theoretical Approach to Self – Organizing Neural Network |
| | 11:30 – 12:00 | COFFEE-BREAK |
| | 12:00 – 13:30 | *Round table* “Science and IT Industry” Free discussion |
| | 13:30 – 14:30 | LUNCH |
| | 14:30 – 17:00 | Nizhniy Novgorod Sightseeing Tour |
| | 17:00 – 17:30 | COFFEE-BREAK |
| | 17:30 – 19:00 | *Round table* Free discussion of joint research projects |
| | 19:00 – 21:00 | FAREWELL DINNER (Hotel Oktyabrskaya) Free discussion |

|  |  |  |
|---|---|---|
|  |  | Evening departure from Nizhniy Novgorod |
| 7th Monday |  | Morning arrival in Moscow<br>Check in Novotel Address:<br>127055, Novoslobodskaya Str. 23, Moscow, Russia<br>Tel: +7 (095) 780-4000, Fax: +7(095) 780-4001<br>Free time. Sightseeing |
|  | 19:00 | UK participamts meet with Deputy Director British Council *Simon Kay* |

# *LIST OF PARTICIPANTS*

(*in alphabetical order*)

**Sergey Belov** PhD in Mathematics (University of Nizhny Novgood, 1992), PhD in Engineering (Osaka Sangyo University, Japan, 1999), associate professor, department of numerical and functional analysis, University of Nizhny Novgorod, Member of the Board of Nizhny Novgorod Mathematical Society. Areas of scientific interest are optimization, partial differential equations, functional analysis, probabilistic networks and nonlinear analysis, parallel computing.

**Alastair Beresford** is a Research Associate in the Computer Laboratory exploring how edge-networks, equipped with network processors, can be used to enhance low-power, resource-starved, ubiquitous computing devices.
http://www-lce.eng.cam.ac.uk/~arb33/

**Stephen Brewster** is a lecturer in the Department of Computing Science at the University of Glasgow, UK. Main research interest is in Multimodal Human-Computer Interaction, sound and haptics and gestures. Research into Earcons, a particular form of non-speech sounds.
http://www.dcs.gla.ac.uk/~stephen/aboutme.shtml

**Matthew Chalmers** is a Reader in Computer Science, University of Glasgow. Co-Director of Research, Kelvin Institute, His current research aims to take account of social and perceptual issues both in the design of computer systems, in visualization, recommender systems and ubiquitous computing, and in the theory of computer science, relating contemporary semiology/philosophy to computational representation. In practice that generally means tracking the hell out of systems that show a lot of information, and then feeding that tracked data back to people and into adaptive system infrastructure.
http://www.dcs.gla.ac.uk/~matthew/

**Victor P. Gergel** PhD in Engineering (1984), D.Sc. Advanced in Engineering (1994), Professor in Software (1998) Co-executor of project of Russian Foundation of Basic Research on developing methods and software for computer-aided decision making. Leader of project for developing computer software of decision making (this software has been presented at the World International Exhibition CeBIT 95, Hannover, Germany). Co-leader of the project of the Programme of Russian-Hungarian Research and Technics Cooperation (1995-98). Leader of a part of the research program of the Russian Ministry of Higher Education and Technologies on Perspective Information Technology in Computer-Aided Modelling and Researching (1999-2001). Professional experience in high performance parallel computations: transputer microprocessor systems (1989), SUN SMP systems (1995), supercomputer Cray (1995), MPP Parsytec PowerXplorer (1996), clusters (2000). Expert in parallel technologies: OCCAM (1989), PVM (1996), MPI (1998), OpenMP (2000).

**Denis V. Ivanov** received his master degree in Mathematics and Ph.D. in Computer Science from Moscow State University, where he currently leads and advises several research projects at the Laboratory of Computational Methods of the Department of Mathematics and Mechanics. Denis has more then 20 publications related to Image Processing, 3D Graphics, Computer Vision and other areas of computer graphics. His primary academic activities include scientific research, lecturing, advising students and serving in organizing and program committees of several conferences, including GraphiCon, which he is proud to be co-chair of in 2004. In 2001, Denis was appointed to be a President of RL Labs JSC, which is a private Russian company specializing in knowledge-intensive software development in such areas as geographic information systems, information technologies and technologies for mobile devices. In 2004, Denis joined Russian Systems Corporation in the position of Director for development. His responsibilities in the Corporation include organization of the complete cycle of product development based on the state-of-the-art software and hardware technologies.

**Yevgeniy Kuzmin** Ph.D. senior researcher at the Laboratory of Computational Methods of the Department of Mathematics and Mechanics of Moscow State University, Member of Editorial Board of "Journal of Fundamental and Applied Mathematics", Russia, Chair of Program Committee at Graphicon International Conferences. Owner and director in several SW and technology development companies. His scientific interests include such areas of Computer Graphics, Computational Geometry, Computer-Human Interfaces. Over 50 publications, 7 issued and pending patents.

**Victor S. Lempitsky** received his master diploma (with honors) in mathematics from Moscow State University, where he is now a PHD student at the Laboratory of Computational Methods of the Department of Mathematics and Mechanics. Victor has 10 publications related to Computer Vision and Computer Graphics. His scientific interests include such areas of Computer Vision as stereo reconstruction, image-based modeling and rendering, structure-and-motion.

**Peter Robinson** is Reader in Computer Technology and Deputy Head of Department at the University of Cambridge Computer Laboratory in England, where he leads the Rainbow Research Group working on computer graphics, interaction and electronic CAD.
http://www.cl.cam.ac.uk/~pr/

**Anton V. Shokurov** received his master diploma (with honor) in Mathematics from Moscow State University, where he is now a PHD student at the Laboratory of Computational Methods of the Department of Mathematics and Mechanics. He has 5 publications related to Computer Vision and Computer Graphics. His scientific interests in Computer Graphics include such areas as stereo reconstruction, computer vision and 3d rendering.

**Quentin Stafford-Fraser** has held research posts at the University of Cambridge, Xerox EuroPARC, Olivetti Research Ltd, and AT&T Labs, where he was a co-inventor of the webcam, and a co-developer of the VNC system before starting the AT&T Broadband Phone project. He co-founded Newnham Research in 2003.
http://www.qandr.org/quentin

**Alexander V. Sysoev** M.Sc. (1999) Microsoft Certified Professional (2003) Co-investigator of the research program of the Russian Ministry of Higher Education and Technologies Perspective Information Technology in Computer-Aided Modeling and Researching (1999–2001). Co-leader in the project of design and development of software system for parallel solving the problems of multidimensional multicriterial optimization. Professional experience in high performance parallel computations, parallel technologies.

**Maja Vukovic** is a second year Research Student in the Rainbow Research Group, working on application modeling to facilitate context awareness under the supervision of Peter Robinson.
http://www.cl.cam.ac.uk/~mv253/

**Nikolai Yu. Zolotykh** PhD in Mathematics (University of Nizhny Novgood, 1998), associate professor, department of mathematical logic and higher algebra, University of Nizhny Novgorod. Areas of scientific interest are discrete optimization, computational learning theory, computer algebra.
http://www.uic.nnov.ru/~zny/

# *PRELIMINARY PROCEEDINGS*

# Parallel Algorithms for Probabilistic Expert Systems[⋆]

S.A. Belov and V.P. Gergel

Nizhni Novgorod State University
Gagarin ave., 23, Nizhni Novgorod 603950, Russia
belov@vmk.unn.ru   gergel@unn.ru

**Abstract.** This paper is devoted to parallel algorithms of inference and learning for probabilistic networks. We introduce parallel algorithms for junction tree inference, loopy belief propagation, Gibbs sampling and EM Learning for systems with distributed memory. Balancing and messages flow optimizing issues are discussed throughout this paper. Scalability results for real networks are presented.

## 1. Introduction

Probabilistic (or Bayesian) networks are a highly active area of research now and the interest to it is still rapidly increasing. This interest is inspired by variety of applications of probabilistic expert systems in diagnostics, bio-informatics, proactive computing, etc, whenever the artificial intelligence is required.  As we know, the first model was first successfully used in 1961 for diagnostics of congenital heart decease [1]. Naive Bayesian inference has been realized for calculating the posterior probabilities on the basis of observations. These brute force calculations cannot be applicable, however, for more or less complicated expert probabilistic systems because the computation time grows exponentially with the size of the systems. For this reason very sophisticated methods of inference and learning have been developed. We would refer our reader to [2], [3] for further references. The developed algorithms and powerful computers have greatly extended the applicability of probabilistic expert systems for solving real problems with a significant impact on resources. However, there are still a lot of important applications where the time required for inference process makes Bayesian networks almost inapplicable. Parallel inference algorithms are viewed as a next step for extending applicability of probabilistic expert systems in highly important areas such as business processes control and proactive computing. In this paper we introduce parallel algorithms for junction tree inference, loopy belief propagation, Gibbs sampling and EM Learning

[⋆] Supported in part by the Intel Research Grant "Parallel Algorithms of Inference and Learning for Graph Probabilistic Models”

for systems with distributed memory and discuss the corresponding scalability results. The rich structure of learning and inference algorithms led us to developing new parallel algorithms in order to reach a good performance results. Throughout this paper we discuss how to balance the load and minimize the messages flow between processors. At first we introduce Gibbs sampling and EM Learning parallel algorithms for which good performance can be reached by means of samples database dividing. Then we consider much more complicated parallel algorithms of loopy belief propagation and, especially, junction tree inference, when we deal with all spectrum of problems which influence negatively on performance results.

## 2. Gibbs sampling and EM Learning parallel algorithms

We will start with a sampling algorithm known as Gibbs sampling. This algorithm, which is based on stochastic simulation technique, is used when the size of original network is too large and its structure is too complicated**.** Suppose we have a discrete set of random variables $X_v$ and full conditional distribution P $(X_v |X_{V\setminus\{v\}})$. Then we can start from admissible initial configuration $(x_1^0, \quad , x_n^0)$ and generate samples in some natural order for each variable upon the current instantiation of other variables and then replace the current instantiation of the variable with its sampled state. Then we can simply use the frequency of appearance to approximate marginal distribution of any variable instead of summing joint distribution over the rest variables. Since it is a recurrent scheme, samples generating cannot be produced in parallel. Therefore we realized the approach when different initial configurations have been assigned to different processors, samples has been generated independently as well as local frequency to be found. All we need after it is done is to calculate the mean value. Obviously we have a maximal possible speedup for such scheme.

**Results of experiments – MPI**

| Number of processors | Average speedup |
|---|---|
| 2 | 1.99 |
| 4 | 3.99 |
| 8 | 7.98 |

Tests are carried out on P4 1300, RAM: 256Mb, LAN 100Mbit

The most amazing thing is that increasing the number of starting points for samples generating is very consistent with the correctness of algorithms, because of high correlations of consecutive samples. Otherwise it takes a long time for the algorithm to forget the initial values

Similar idea has been successfully applied for creating a parallel version of so-called EM learning algorithm of maximum likelihood estimation from incomplete database with data missing at random [3]. Having a database of samples and graph structure of probabilistic networks we are supposed to find corresponding conditional probabilities in two steps. First we calculate the current expected marginal counts for each configuration based on observed data (E step) and then we calculate conditional probabilities during M step by dividing the obtained marginal count for the family, which contains the node by the corresponding count of its parents (M-step). To create a parallel version we simply divide the database equally between processors making both E-step and M –step in parallel. Again all we need is to calculate mean-values before the end of algorithms. The results of experiments are given below.

**Results of experiments – MPI**

| Number of processors | Average speedup |
|---|---|
| 2 | 2.00 |
| 4 | 3.99 |
| 8 | 7.99 |

Tests are carried out on P4 1300, RAM: 256Mb, LAN 100Mbit

## 3. Loopy Belief Propagation and Junction Tree parallel algorithms

This is a one of most popular algorithms of inference in probabilistic networks with loops based on special message passing process between the nodes [2], [4], [5]. The size of message to be processed depends on different factors and it strongly varies from one node to another. It is possible, however, to calculate theoretically the computational load for each messages depending on the architecture and it has been done. For the first approximation we have constructed a skeleton of the original networks and then we have divided the skeleton by parts with an approximately equal load. However, it does not allow reaching a good performance, due to messages passing between processors in original graph. The special skeleton of maximal weight, which has been built to minimize an additional messages flow (not the number of messages, of course), has helped us to improve the performance. The effect of asymmetric calculations between sender and recipient of the message has been taken into account. The new scheme of transfer of messages in which the quantity of transmitted messages tends to minimum has been also introduced. In this scheme the messages are incorporated in one package. Thus, each process forms data packages for other processes during processing the nodes. After processing nodes each process exchanged the data with other processes. Still the difference is essential between the general networks and regular networks, when special nodes dividing can be implemented to make good balancing and minimize messages flow between processors. The results are given below

**Results of experiments – MPI**

| Number of processes | Average speedup for a general networks | Average speedup on lattices |
|---|---|---|
| 2 | 1,88 | 1.96 |
| 4 | 2,88 | 3.89 |
| 8 | 4,57 | 7.69 |

Tests are carried out on P4 1300, RAM: 256Mb, LAN 100Mbit

It should be also noted that very interesting effect takes place if we use asynchrony. It turns out that, generally speaking, the number of iterations needed for convergence grows drastically if we use the available by the moment data for messages passing. However, if static balancing is near to perfect (lattices, for example), then such scheme performs very well.

Last parallel algorithm we would like to discuss here is so-called junction tree inference algorithm, which uses special tree of cliques (containing different number of nodes) with data structure called potentials, which represent functions over all possible variables in the clique. There are at least two challenges for anyone who tries to develop a scalable parallel version of such algorithm. First of all there is a special schedule of cliques processing in junction tree (from leaves to the root and vise versa). Therefore we cannot simply divide the tree by subtrees to reach a good performance. The idea of making a new root to balance the tree before processing is working well only for two processors. New schemes should be considered.

The next challenge is an existence of large cliques in junction tree after triangulation. In our database of quite general Bayesian networks the experimental average of weight of the whole tree to the weight of the biggest clique is 2.45, when standard one step look ahead triangulation procedure is realized. Developed heuristic algorithms of triangulation have allowed improving substantially this coefficient of maximal possible speedup (It is well known that finding optimal triangulation procedure is NP-hard problem.) The search for scalable parallel algorithm of junction tree inference is under investigation now.

## References

1. Warner H.R., Toronto A.F, Veasey L.G., Stephenson R. A mathematical approach to medical diagnosis – application to congenital heart disease. Journal of the American Medical Association, 177 (1961) 177–184
2. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan and Kaufman Publishers, San Mateo, California (1988)
3. Cowell R.G, Dawid A.P, Lauritzen S.L. and Spiegelhalter D.J. Probabilistic Networks and Expert Systems, Springer (1999)
4. Murphy K.P. Dynamic Bayesian networks: representation, inference and learning. PhD Thesis, University of California, Berkeley (2002)
5. Weiss Y. Correctness of local probability propagation in graphical models with loops. Neural Computation, 12 (2000) 1–41

# Towards automated computation sharing for ubiquitous computing

Alastair R. Beresford and Andrew C. Rice

Cambridge Laboratory,
University of Cambridge,
15 JJ Thomson Avenue,
Cambridge CB3 0FD. UK
{arb33,acr31}@cam.ac.uk

**Abstract.** Ubiquitous computing envisions an era when users own a hundred or a thousand computational devices. These devices will have a large variation in computational ability, network connectivity and mobility. In this paper we argue that applications running on a ubiquitous computing platform should be able to automatically distribute their execution over multiple devices in order to maximise their utility to the end user. We present a new algorithm for autonomously identifying shared computation between different distributed applications and demonstrate the utility of our model with reference to an existing location sensing system.

## 1 Introduction

Ubiquitous computing envisions an era when users own a hundred or a thousand computational devices. Manual supervision of such a large number of devices will be infeasible and therefore the applications running on these devices must automate the services they offer to users whenever possible. Context-aware computing aims to enable such automation by giving applications detailed information about their environment; applications can then use this context to adapt their operation automatically.

Environmental data are usually gathered from a set of sensors distributed throughout the environment. Raw sensor data are often of little direct use to applications and are therefore normally processed into higher-level context information. For example, the Bat system [1] and associated middleware [2] provides context-aware applications with location information about people and objects. The Bat system converts time-of-flight measurements of ultrasound pulses from special tags into estimates of physical tag location in three-dimensional space. Location information can be provided directly to applications or can be further refined by the middleware. For example, applications might define virtual containers, or regions, for tags or the environment and register interest in container intersection and separation [3].

In a ubiquitous computing environment, many applications will require access to the same higher-level context information; therefore program development and execution costs can be amortised over multiple applications. Several context-aware middleware systems have been developed to perform these functions, however they are usually centralised, with the functionality of any distributed computation manually allocated. In addition, data gathered from the sensor networks is decoupled from the applications, resulting in data collection activities even when the data produced have no application output; this can result in inefficient power usage in wireless sensor networks and sub-optimal scheduling of data collection in systems where there are bandwidth restrictions.

Work in related areas such as active networks, grid computing and ad-hoc sensor networks distribute computation in a much more dynamic way. We believe ubiquitous computing applications and services would benefit from greater distribution of context-processing because:

- sensors are increasingly attached to mobile devices, e.g. cameras on mobile telephones, and therefore these new sensors need to be incorporated into the available sensory inputs in a dynamic way;
- an automated scheme for decentralisation would allow new computing resource or new sensor hardware to be dynamically added to a running system (if components are not dynamically managed, the system will not scale to a thousand devices);
- many end-users who wish to deploy ubiquitous computing services do not have the technical skill to configure systems, so an automated, dynamic system could take advantage of resources when they become available;
- mobile devices are often carried by users, and these should be able to dynamically make use of resources and applications in the local network.
- Since the sensor, actuator and application numbers are dynamic, we wish to reallocate resource dynamically (it would be too expensive to have static machines allocated for each task).
- with large numbers of computers, failure of some components is inevitable, so the system should be dependable and cope with failure of components and adapt to unforeseen changes.

## 2 Sensor graphs

Sensors systems are used to measure physical properties of the environment. In ubiquitous computing, it is common for sensors to be paired with passive or active devices which augment the environment in a way suitable for measurement by the sensor system; this makes detection of context easier, and in many cases improves the accuracy of the data. For example, the TRIP vision tracking system [4] affixes circular, two-dimensional bar codes on items which can then be tracked with cheap cameras attached to computers; vision systems of this sort are discussed in greater detail in Section 3.

Middleware platforms usually provide applications with access to sensor data in two modes: (1) an event-driven style, where data of interest are streamed to applications when available, or (2) a query style, where applications make an explicit request for a particular piece of information.

Sensor data are often processed in multiple stages. Each stage produces information which might be used by a context-aware application, or undergo further processing by another stage. The processing of sensor data can be modelled as a directed *processing graph* $G = (V, E)$, where each vertex $v \in V$ represents a distinct stage of processing, and each edge $e \in E$ represents a communication between two processing stages. Inputs to the directed graph initially come from sensors, or as queries from applications; outputs from the graph send data to applications. Cycles in the graph are used to represent data processing through iterative refinement, e.g. simulated annealing, or to depict feedback from later stages of processing, which may be required by earlier processing steps.

We define a tuple of resource requirements $r = \langle r_1, \ldots, r_i \rangle$ for each vertex and each edge in the graph. Types of resource include processing power, memory and disk space, latency and network bandwidth as well as more specialised hardware resources and sensors (e.g. FPU support, video capture hardware, etc.). In addition, achieving the same task with different types of resource may have different financial implications; for example connecting to other machines via GPRS means the user must pay a telephone operator to route their data, whereas using Bluetooth is free. We can view such monetary costs as an additional edge resource requirement.

A typical ubiquitous computing environment has a large number of devices which will have different available resources. The position and connectivity of networked computing resources can also be modelled as an undirected *resource graph*, where vertexes represent physical machines supporting applications, data processing and sensors; edges represent communication links. Since many of the devices in the network will be mobile, the resource graph will be constantly changing. Therefore a device discovery protocol, such as ZeroConf [5], is required to maintain the state of the current set of resources.

Context-aware applications can be specified by a processing graph, and since many applications may run simultaneously, we wish to composite all processing graphs onto the resource graph. This can be achieved in two stages: (1) compose all application processing graphs into a single, combined processing graph; and (2) map the combined processing graph onto the resource graph. The first stage is discussed in more detail in the next section.

Mapping a processing graph onto a set of computing resources is equivalent to partitioning the combined processing graph into $k$ distinct subgraphs, where each subgraph contains the set of tasks to be performed by a particular machine in the resource graph. Often, creating an optimal multi-way partition of a graph, such as minimising the maximum cost of cut edge weights between any two vertexes, is NP-hard. Approximations do exist, and solutions have been forthcoming in many areas. For example, in the field of peer-to-peer networks, Svitkina and Tardos approximate the optimal distribution of data in order to minimise the number of requests to any one server [6]. Multi-way partition approximations are also required for many grid applications, where a distributed computation should minimise both the load imbalance of the servers (resulting in shorter execution times) and the communications overhead between machines [7]. In this paper we

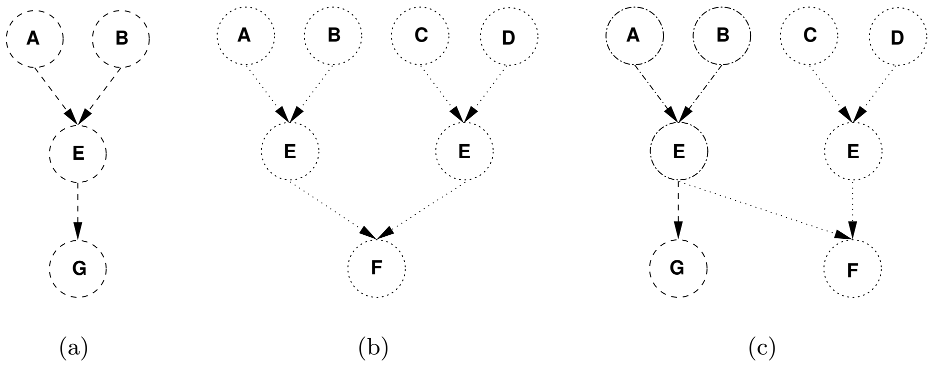


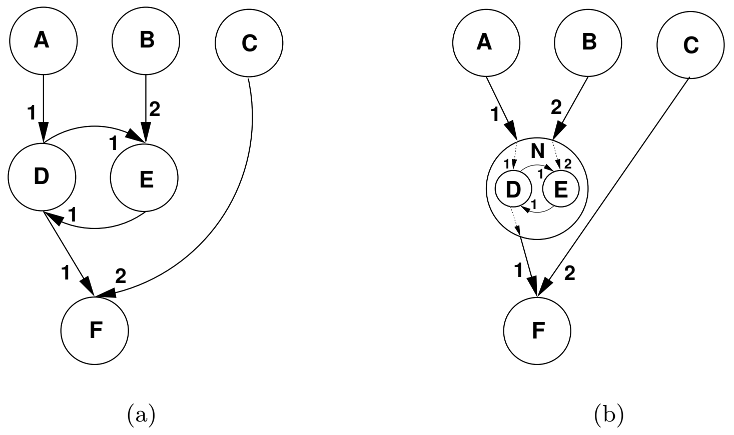
**Fig. 1.** Figures (a) and (b) show processing graphs for two separate applications. Figure (c) shows the merged processing graph, combining the required processing for the two applications. Note that the two vertexes labelled '*E*' in Figure (c) cannot be merged because their inputs come from different sources, so their outputs will differ.

concentrate on developing a method for combining multiple processing graphs, and leave the development of automatic partitioning of a processing graph onto a resource graph for future work.

### 2.1 Composing processing graphs

When composing multiple application processing graphs together, it is important to ensure that data processing is not repeated. Many applications will process the same sensor data using the same functions, and we would like to ensure that this is only done once whenever possible. Figure 1 provides example processing graphs for two applications; the two applications share common sensors and processing stages, and these steps can be shared when the two applications are executed together.

A vertex with no ingress edges represents a *sensor node*. If two graphs contain the same sensor nodes, the vertexes must be shared when the processing graphs are mapped to the physical resource graph, since the two sensor nodes represent the same underlying sensor. For this purpose, every vertex representing a sensor node is given a unique name by the device discovery protocol, and this is used by applications to specify their sensor requirements. Vertexes with at least one ingress edge and one egress edge represent *processing nodes* which perform some computation on their inputs and, possibly, produce an output. Vertexes representing processing nodes are given a name which uniquely defines their function. Vertexes with at least one ingress edge and no egress edges are *application nodes*. Each ingress and egress edge of a processing node or application node is labelled so that it can be distinguished from any another. (This is analogous to the ordering of parameters passed to a function in a programming language.)



(a) (b)

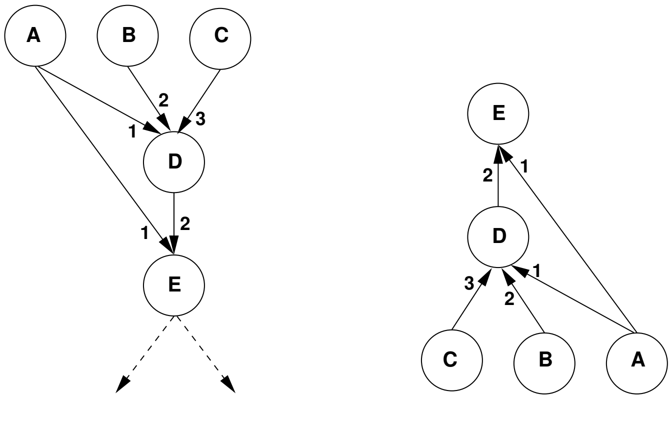
**Fig. 2.** Figure (a) shows the original processing graph $G$ for an application; Figure (b) provides an acyclic reduction of $G$ by introducing a meta-vertex $N$ to represent the strongly connected component $G' = (\{D, E\}, \{(D, E), (E, D)\})$.

In general, deciding whether there exists a subgraph of one graph which is isomorphic to another graph is NP-Complete [8]. Our processing graphs have more constraints than a general graph however, and there exists a polynomial time algorithm to produce a combined processing graph. Our solution is found by performing the merging process in two stages: (1) remove all cycles from the graph; and (2) merge the remaining acyclic graph.

A strongly connected subgraph of a directed graph is one in which every vertex is reachable from every other. Therefore a directed graph with no strongly connected subgraphs containing two or more vertexes is acyclic. Tarjan developed an algorithm for finding all strongly connected subgraphs of a graph with $v$ vertexes and $e$ edges in $O(v + e)$ time [9]. More recently, Nuutila and Soisalon-Soininen improved the performance of Tarjan's algorithm for sparse graphs and graphs with many trivial components [10].

All the cycles in the two processing graphs can be removed by (1) finding all strongly connected subgraphs; and (2) replacing all the vertexes of each strongly connected subgraph with a single meta-vertex. Each meta-vertex has an ingress edge from normal vertex if, and only if, the there was an ingress edge from the normal vertex to a vertex in the strongly connected subgraph; the same applies for an egress edge. More formally,

**Definition 1** *Given a graph* $G = (V, E)$ *with a strongly connected subgraph* $G' = (V', E')$*, we generate a new, acyclic graph* $G^- = (\{v_{meta}\} \cup (V \setminus V'), (E \setminus E') \cup \{(v, v_{meta}) | (v, v') \in E \wedge v \in (V \setminus V') \wedge v' \in V'\} \cup \{(v_{meta}, v) | (v', v) \in E \wedge v \in (V \setminus V') \wedge v' \in V'\})$



(a) Graph (b) Fullname

**Fig. 3.** Figure (a) depicts part of a processing tree. Figure (b) describes the fullname tree for vertex $E$, including depth-first search ordering for the fullname.

Any vertex which is created to replace a strongly connected subgraph is marked to denote that it replaces a subgraph. A graph which contains no strongly connected subgraphs with two or more vertexes is acyclic. Therefore the transformation above can be used to reduce a general process graph to an acyclic process graph. Figure 2 provides an example conversion of a general graph into an acyclic graph.

Two acyclic processing graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can then be merged into $G_m = (V_m, E_m)$. The merging process operates on a topologically sorted version of the acyclic graph. (A topologically sorted graph positions a vertex at row $i$ if at least one of the vertexes connected by an ingress edge is at row $i - 1$ and all the vertexes connected to it by an ingress edge have a row value $< i$; if a vertex has no ingress edges, it is positioned at row zero.) We use the notation $V[i]$ to denote all vertexes in the set $V$ which are positioned at row $i$.

The zeroth row of the acyclic graph consists of sensor nodes. Each sensor node has a unique name and therefore every vertex is distinct. A merged version of the graph for the zeroth row is simply the union of the sensor nodes in both graphs; in other words $V_m[0] = V_1[0] \cup V_2[0]$. Sensor nodes are *shared* in $G_1$ and $G_2$ if the vertexes are in both processor graphs; in other words, $V_1[0] \cap V_2[0]$ are shared between $G_1$ and $G_2$.

The $i$th row of the processing graph can contain application nodes and/or processing nodes. The function names for these nodes may not be unique, since their name relates to the function of the node. A *fullname* for each vertex in the $i$th row is represented as a tree of the reverse paths of all ingress nodes.

| Step | Description | Data passed to next stage |
|---|---|---|
| 1 | Acquire a grey-scale image from a CCD camera. | *Grey-scale image* |
| 2 | Convert this image into a 1-bit black and white image by pixel thresholding. | *Binary image* |
| 3 | Traverse the image extracting contours which might correspond to the concentric circles on the tag. | *Contour pixels* |
| 4 | Fit an ellipse to each contour. | *Ellipse formulae* |
| 5 | Derive the inverted projection transform. | *Transformation matrix* |
| 6 | Estimate suitable points on the image to sample in order to read the data payload. | *Tag information* |

**Table 1.** Six basic stages involved tag recognition. Note: The image itself is not required as an input to stages 4 or 5, but is required for reading the tag data payload (stage 6).

Recall that each ingress edge has a unique number (with respect to the destination vertex) assigned to it. The lexicographical ordering of the ingress edge numbering can be used to determine a unique ordering on a depth-first search of the fullname tree. Two processing vertexes will have the same matching vertex names and edge layout in a depth-first search of their fullnames if, and only if, all the processing nodes and sensor nodes above them in the processing graph are also the same. Therefore two vertexes $v_1 \in V_1[i]$ and $v_2 \in V_2[i]$ can be shared in the merged graph if and only if $\mathit{fullname}(v_1) = \mathit{fullname}(v_2)$. Figure 3 depicts an example graph and its related fullname.

If a meta-vertex $v$ represents a strongly connected subgraph, a name for it is constructed when vertex $v$ appears in the current row to be processed. A depth-first search of the subgraph can be used to construct a name. The search begins from the subgraph vertex which is connected to the main acyclic graph via the lowest edge number on vertex $v$. Within the subgraph, edge numbering is used to determine the depth-first search order. Once a name has been constructed for vertex $v$, a fullname for vertex $v$ can then be constructed as normal.

## 3 Example application

The TRIP vision system [4] is capable of determining the relative location and pose of a passive printed paper tag from a camera. The tags described in this paper use two concentric circles as a prominent fiducial which can be detected by image processing. The two concentric circles provide sufficient constraints to enable estimation of the projective transform from the tag to the camera; from this information, an estimate of location and pose of the tag can be extracted.

We have implemented a visual tag recognition system which is capable of recognising TRIP tags as well as other fiducial tags. It provides a generic framework for evaluating the automated allocation and distribution of network resources. A simple program can be written to take advantage of the generic framework and either run tag recognition entirely locally or distribute arbi-

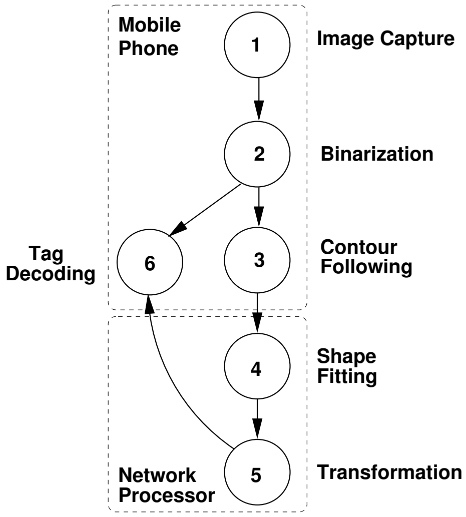


**Fig. 4.** A camera-enabled mobile telephone can off-load some of the processing to nearby network processors.

trary stages across multiple hosts. The six processing stages of the visual tag recognition algorithm are described in Table 1.

There are a many scenarios where it is necessary to distribute the image processing algorithm over a multiple computers. Three examples are outlined next.

- In order to monitor a whole room, several cameras may be required to ensure good coverage. If all of these are connected to the same machine for processing, then the total framerate could exceed 100 frames-per-second. In this case it is necessary to distribute processing of the raw image data to additional machines.
- A mobile telephone equipped with a camera can be used as a source of images for processing. Resource usage on the phone can be minimised by performing the relatively simple steps of binarisation and contour following before sending the edge pixel information (which is smaller than the original image) over a wireless network. Network processors can be used to perform the ellipse fitting and transformation stages; the resultant transformation matrix for each tag can be transferred back to the phone. The phone may then read the data payload from the tag image to complete the identification process. See Figure 4.
- Our vision system supports many different tag designs; each design provides a different tradeoff between data payload size, location accuracy, and reliability. Therefore two or more applications may use different tag designs and each request a different mode of processing. Many of the processing stages are unchanged between each application and therefore applications are able to share the results of some of the processing stages. See Figure 5.

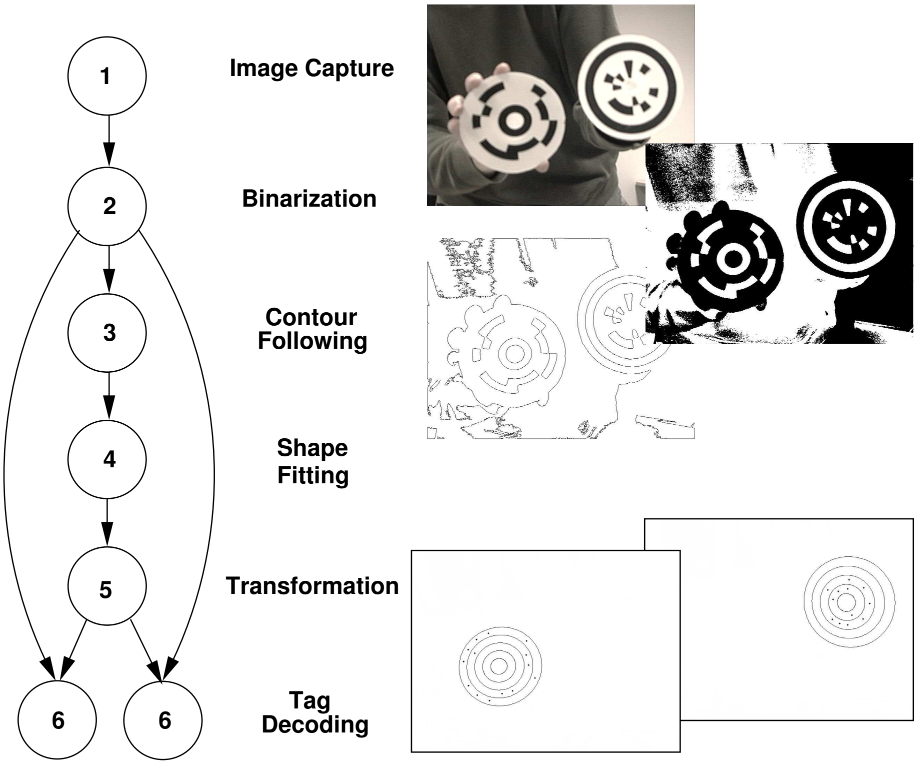


**Fig. 5.** Five out of the six processing stages of the location system can be shared between applications which use two different types of tag design.

The first scenario demonstrates the need to distribute the computation due to processing requirements. In the second case, a mobile phone without a floating point unit will have difficulty computing the results of stages 4 and 5 but also suffers from reduced network bandwidth; this scenario demonstrates the potential for minimising network bandwidth and utilising in-network floating point resources. Scenario three demonstrates the benefit of sharing elements of computation between applications in order to increase overall system efficiency.

## 4 Related work

Gu et al. describe an adaptive offloading mechanism for ubiquitous computing [11]. Their aim is to allow a resource-constrained mobile device to dynamically execute portions of an application on nearby computers with wired power and larger resources. The offloading mechanism is activated when the memory usage of the application approaches the memory capacity of the mobile device. The authors assume the application is written in an object-oriented language such as Java of C# and the environment has plenty of available wireless bandwidth. The offloading system dynamically partitions the code and automatically marshals access to remote objects over the network.

The Solar Project [12] developed a middleware system to enable distributed processing of context information. Applications combine distributed sensor data into higher-level contextual information using an acyclic graph. The data processing is performed at one or more nodes, which are fully connected via an overlay network. Routing between processing nodes is performed over a peer-to-peer substrate. Failure of hosts or processing components is managed by peer-wise monitoring of liveness. If a peer detects a host or component failure, the affected services are automatically restarted.

SpiderNet [13] is a decentralised multimedia service composition framework. The system supports multiple, statistical, quality-of-service constraints for an acyclic graph of processing steps. Processing occurs on media servers connected via an application-level overlay network. A processing graph is constructed with a peer-to-peer algorithm which determines the suitable processing nodes, gathers information on the resources available and then distributes the processing graph.

The Context Toolkit [14] is inspired by the development of graphical user-interface toolkits built for personal computers. The toolkit insulates the application developer from the details of the context-aware widgets and provides an abstract query or callback interface. Widgets talk via TCP/IP and can therefore be distributed over multiple machines.

## 5 Conclusions

Devices built for ubiquitous computing will exhibit many asymmetries in terms of computational resources, network connectivity and power requirements. These differences motivate the need for effective distribution of applications over a network. Many existing ubiquitous computing platforms are centralised or manually distributed. We have argued that automated allocation and distribution is vital due to the increasing complexity and short-timescale reconfiguration of ubiquitous systems. This has motivated the development of an efficient algorithm for autonomously sharing computational units between, possibly cyclic, process graphs.

The ability to distribute a computation and to share commonly used results between applications increases the efficiency and utility of a context-aware environment. In this paper we have described how our approach permits effective distribution of the elements within the our visual tag recognition system.

## Acknowledgements

This work was supported by EPSRC. Thanks to Alan Mycroft and Andy Hopper for their advice and support.

## References

1. Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.
2. Noha Adly, Pete Steggles, and Andy Harter. SPIRIT: A resource database for mobile users. In *Proceedings of ACM CHI'97 Workshop on Ubiquitous Computing*, March 1997.
3. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2/3):187–197, 2002.
4. Diego López de Ipiña, Paulo Mendonça, and Andy Hopper. TRIP: a low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, May 2002.
5. Erik Guttman. Autoconfiguration for IP networking: Enabling local communication. *IEEE Computer*, 5(3):81–86, May 2001.
6. Zoya Svitkina and va Tardos. Min-max multiway cut. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 3122, August 2004.
7. Chris Walshaw and M. Cross. Multilevel mesh partitioning for heterogeneous communications networks. *Future Generation Computer Systems*, 17, 2001.
8. Mikhail J. Atallah, editor. *Algorithms and Theory of Computation Handbook*. CRC Press, 1999.
9. Robert Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2):146–160, June 1972.
10. Esko Nuutila and Eljas Soisalon-Soininen. On finding the strongly connected components in a directed graph. *Information Processing Letters*, 49(1):9–14, 1994.
11. Xiaohui Gu, Alan Messer, Ira Greenberg, Dejan Milojicic, and Klara Nahrstedt. Adaptive offloading for pervasive computing. *IEEE Pervasive Computing*, pages 66–73, July 2004.
12. Guanling Chen. *Solar: Building a context fusion network for pervasive computing.* PhD thesis, Dartmouth College, August 2004.
13. Xiaohui Gu and Klara Nahrstedt. Distributed multimedia service composition with statistical qos assurances. *IEEE Transactions on Multimedia*, 2005. *To appear.*
14. Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 434–441. ACM Press, 1999.

# Multimodal Interaction and Proactive Computing

Stephen A Brewster

Glasgow Interactive Systems Group
Department of Computing Science
University of Glasgow, Glasgow, G12 8QQ, UK
E-mail: stephen@dcs.gla.ac.uk Web: www.dcs.gla.ac.uk/~stephen

**Abstract.** One important issue for proactive computing is how users control and interact with the systems they will carry and have access to when they are out in the field. One solution is to use multimodal interaction (interaction using different combinations of sensory modalities) to allow people to interact in a range of different ways. This paper discusses gestural interaction as an alternative for input. This is advantageous as it does not require users to look at a display. For output non-speech audio and tactile displays are presented as alternatives to visual displays. The advantages with these types of displays are that they can be unobtrusive and do not require a user's visual attention. The combination of these underutilised senses has much potential to create effective interfaces for proactive systems.

## 1. Introduction

As more and more devices incorporate some form of computation people will soon carry and be connected to a large number of systems and services all of the time. Users going about their everyday lives need effective ways of managing them otherwise the effort of controlling them will be too great and they will not be used. To avoid such problems we need flexible, efficient ways to interact with and monitor the systems and services. In a proactive computing world these devices will also be making decisions for users, who will need to be kept informed of status and outcomes without unnecessary disruption [20]. Designing user interfaces to support such activities is not well understood (nor is how we realistically evaluate their effectiveness).

Ljungstrand *et al.* [14] suggest some important questions that need to be answered to develop the area of human-computer interaction (HCI) within proactive computing, amongst these are:

- What are the best means for controlling proactive computers and agents?
- What kind of manipulation and feedback mechanisms do users need, at what levels, how often, and how should feedback be manifested?
- How can we design user interfaces that take advantage of all the human senses, as well as our inherent skills in moving about in the real world and manipulating real things?

This paper will begin to deal with some of these issues, but much more work will need to be done to really understand how to design good proactive computing interactions. A starting point in thinking about how interactions might be designed is to look at how people currently cope with complex situations. In the real world we deal with large amounts of data all the time. We do this using a range of different senses and the combination of senses avoids any one becoming overloaded. Multimodal human-computer interaction studies the use of multiple different sensory modalities to enable users to interact effectively with computers.

The interface designs of most mobile and wearable computers are based heavily on those of desktop graphical user interfaces. These were originally designed for users sitting at a computer to which they could give their full (visual) attention. Users of proactive or mobile systems are often in motion and performing other tasks when they use their devices. If they are interacting whilst walking, running or driving, they cannot easily devote all of their visual attention to the interface; it must remain with the main task for safety. It can be hard to design visual interfaces that work well under these circumstances.

Much of the interface work on wearable computers tends to focus on visual displays, often presented through head-mounted graphical displays [1]. These can be obtrusive and hard to use in bright daylight, plus they occupy the users' visual attention [11] when it may be needed elsewhere. Other solutions utilise nearby resources that your 'personal server' might connect to for display or input [21]. These again may be difficult to use when on the move. One of the foci of work at Glasgow is on how far we can push non-visual interaction so that we do not tie users to visual displays and conventional input devices.

Both input and output need to be considered when designing proactive interactions. This paper will discuss some of the possibilities of the different senses and give some examples of how they might be used to create an effective proactive interface.

## 2. Input Techniques

Making input when in the kinds of scenarios envisaged by proactive computing is problematic; users will be out in the real world doing tasks that may be supported by computers. They may be mobile or engaged in an activity that needs the focus of their attention so cannot give it all to the computer they are carrying.

Current mobile and wearable computers typically use a touch screen and stylus, or small keyboard. These are effective when stationary but can be difficult to use when mobile. Buttons and widgets on touch screens tend to be small due to the small screens required to make the devices portable. This makes the targets hard to hit and input error prone, because the device and stylus are both moving as the user moves around the environment, making accurate pointing difficult. Similar problems affect stylus input of characters when on the move. Brewster [4] showed that when a stylus based device was used whilst walking performance dropped by over 30% compared to sitting. Small keyboards tend to have similar difficulties as the keys must be small enough to allow the keyboard to be easily carried and so become hard to press.

In all of these cases much visual attention is required to make input. Users must look closely to see the small targets and the feedback to indicate they have been used correctly. Visual attention is, however, needed for navigating the environment around the user. If too much is required for the interface then users may have to stop what they are doing to interact with the system, which is undesirable.

Many of these techniques also require two hands, which can be problematic if the user is engaged in other activities. The 'Twiddler' [1], a small chord keyboard, requires only one hand but it can be hard to use and requires learning of the chords.

Speech recognition is often suggested as a future alternative input technique. This has great potential but at present is not good in fully mobile environments due to high processor and memory requirements and highly variable background noise levels. There are also issues of error recovery without visual displays. If great care is not taken, error recovery can become very time consuming.

## 2.1 Gestural interaction

One alternative technique gaining interest is gestural interaction. Gestures can be done with fingers on touch screens, or using head, hands or arms (or other body parts) with appropriate sensors attached. They can also be attached to devices such as handheld computers or mobile phones to allow them to be able to generate gestures for input. Harrison *et al.* [13] showed that simple, natural gestures can be successfully used for input in a range of different mobile situations.

Gestures are a good method for making input because they do not require visual attention; you can do a gesture with your hand, for example, without looking at it because of your powerful kinaesthetic sense – you know the positions, orientations and movements of your body parts because you sense them through your muscles, tendons and joints. This means that input can be made without the need for visual attention.

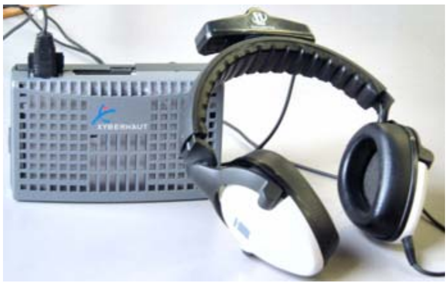


**Figure 1: A simple wearable computer system comprising a Xybernaut MAV wearable computer, a pair of standard headphones and an Intersense orientation tracker for detecting head movements (on top of the headphones) [7].**

The use of hands or arms may be problematic if users are carrying equipment, but there are still possibilities for input via the head. We have looked at using head nods for making selections whilst on the move [7]. Head pointing is more common for desktop users with physical disabilities [15], but has advantages for all users, as head movements are very expressive. There are many situations where hands are busy but the head is still free to be used for input. There are still important issues of gesture recognition to be dealt with as users nod and shake their heads as part of normal life and we need to be able to distinguish these nods, or nods that people might do when listening to music, from nods to control the interface.

Figure 1 shows an example of a simple audio-based wearable computer that used head gestures for input [7]. The sensor we used was an off-the-shelf model which could easily be made much smaller and integrated into the headphones. Figure 2 shows a Compaq iPAQ with an accelerometer attached (devices such as mobile phones are now also incorporating accelerometers). This can be used to detect movement and orientation of the device. We have also used this to allow tilting for input. In the simplest case this might be tilting to scroll (although this can be difficult as the more you tilt the harder the screen is to see) or more sophisticated interactions may use tilting for text entry. Gesturing with the whole device is also possible, for example to allow users to point at objects or draw simple characters in space in front of them.

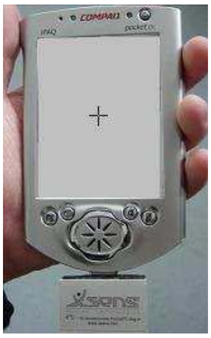


**Figure 2: A Compaq iPAQ handheld computer with an Xsens 3-axis accelerometer for detecting device movements (www.xsens.com).**

To assess the use of fingers on touch screens for input we developed a gesture driven mobile music player on a Compaq iPAQ [17]. Centred on the functions of the music player – such as play/stop, previous/next track – we designed a simple set of gestures that people could perform whilst walking. Users generated the gestures by dragging a finger across the whole of the touch screen of the device (which was attached to a belt around their waist) and received non-speech audio feedback upon completion of each gesture. Users did not need to look at the display of the player to be able use it. An experiment showed that the audio/gestural interface was significantly better than the standard, graphically based, media player on the iPAQ when users were operating the device whilst walking. One reason for this was that they could use their eyes to watch where they were going and their hands and ears to control the music player.

These kinds of interactions have many benefits for proactive systems. Users can make input with parts of their bodies that are not being used for the primary task in which they are involved. Certain types of input can be made without the need for a screen, or even a surface, which makes them very flexible and suitable for the wide range of interaction scenarios in which proactive computer users might find themselves.

### 2.2 Sensing additional information from accelerometers

One extra advantage of devices equipped with accelerometers (such as Figure 2) and other motion sensors is that other useful information can be gained about the context of the interaction in addition to data for gesture recognition. This is important for proactive systems as they must communicate with their users in subtle but effective ways and knowing something about the user's context will help this. There is much existing work in the area of context-aware computing which is beyond the scope of this paper, but data from accelerometers gives some other useful information that has not been considered so far. With instrumented devices we can collect information to provide input to allow a system to make decisions about how and when to present information to a user, and when to expect input.

Alongside gesture recognition, we can use the accelerometers to gather information about the user's movement. When users are walking, for example, we can extract gait information from the data stream. Real-time gait analysis allows the display to be changed to reduce its complexity if the user is walking or running, as the users attention will be elsewhere, or to compensate for input biases and errors that occur because of the movement. For example, we have found that users are significantly more accurate when tapping targets during particular parts of the gait cycle. So, any system we create must allow the user to interact appropriately when on the move or we may end up with a system that is unusable, or alternatively forces the user to stop what he/she is doing to operate the interface. The accelerometers also give us information about tremor from muscle movements that we can use to infer device location and use. We have used this, for example, to allow the user to squeeze the device to make selections; the tremor frequency changes when the user is squeezing and we can easily detect this change and use it as an input signal.

## 3. Output

Current mobile and wearable devices use small screens for displaying information and this makes interaction difficult. Screen size is limited as the devices must be small enough to be easily carried. As mentioned above, the user interfaces of many current mobile and wearable computers use interaction and display techniques based on desktop computer interfaces (for example, windows, icons, pull-down menus). This is not necessarily the best solution as users will not be devoting their full attention to the systems and devices they are carrying; they will need to keep some of their attention on the tasks they are performing and the environment through which they are moving. Head-mounted augmented-reality displays overcome some of these problems by allowing the user to see the world around them as well as the output from their wearable systems. However, there will always be problems with the competing demands on visual attention (and also the obtrusive technologies that users currently have to wear). Humans have other senses which are useful alternatives to the visual for information display, but they are often not considered. One aim of the research done at Glasgow is to create systems that use as little of the users' visual attention as possible by taking advantage of the other senses.

### 3.1 Non-speech audio display

There is much work in the area of speech output for interactive systems, but less on non-speech sounds. These sounds include music, sound from our everyday environment and sound effects. These are often neglected but can communicate much useful information to a listener. With non-speech sounds the messages can be shorter than speech and therefore more rapidly heard (although the user might have to learn the meaning of the non-speech sound whereas the meaning is contained within the speech – just like the visual case of icons and text). The combination of these two types of sounds makes it easy for a proactive system both to present status information on continuously monitored tasks in the background and to capture a user's attention with an important message.

There are two basic types of non-speech sounds commonly used: Earcons [2] and Auditory Icons [10] (for a full discussion of the topic see [3]). Earcons are highly structured sounds based around principles from music, encoding information using variations in timbre, rhythm and melody. Auditory icons use natural, everyday sounds that have an intuitive link to the thing they represent in the computer. The key advantages of non-speech sounds is that they are good for giving status information, trends, for representing simple hierarchical structures and grabbing the user's attention. This means that information that may normally be presented visually could be presented in sound, thus allowing users to keep visual attention on the world around them.

Sound can significantly improve interaction in mobile situations. Brewster [4] showed that the addition of simple non-speech sounds to aid targeting and selection in a stylus/touch screen interface significantly reduced subjective workload, increased tapping performance by 25% and allowed users to walk significantly further. This was because the user interface required less of the users' visual attention, which they could then use for navigating the environment. This suggests that information delivered in this way could be very beneficial for proactive computing environments.

Sawhney and Schmandt's Nomadic Radio [18] combined speech and auditory icons. The system used a context-based notification strategy that dynamically selected the appropriate notification method based on the user's attentional focus. Seven levels of auditory presentation were used from silent to full speech rendering. If the user was engaged in a task then the system was silent and no notification of an incoming call or message would be given (so as not to cause an interruption). The next level used 'ambient' cues (based on Auditory Icons) with sounds like running water indicating that the system was operational. These cues were designed to be easily habituated but to let the user know that the system was working. Other levels used speech, expanding from a simple message summary up to the full text of a voicemail message. The system attempted to work out the appropriate level to deliver the notifications by listening to the background audio level in the vicinity of the user (using the built-in microphone) and if the user was speaking or not. For example, if the user was speaking the system might use an ambient cue so as not to interrupt the conversation

One extension of basic sound design is to present sounds in three-dimensions (3D) around the listener. This gives an increased display space, avoiding the overload that can occur when only point source or stereo sounds are used. Humans are very good at detecting the direction of a sound source and we can use this to partition the audio space around the listener into a series of 'audio windows' [9]. To increase the accuracy of perception most 3D auditory interfaces just use a plane around the users head at the height of the ears. Audio sources can then be played in different segments of the circle around the head. The use of a head-tracker (see Figure 1) means that we can update the sound scene dynamically, allowing egocentric or exocentric sound sources.

Brewster *et al.* [7] used a 3D auditory display to create an 'eyes-free' interaction for use on the move. As mentioned above, this interface used head nods to allow users to interact: a nod in the appropriate directed selected a source. The idea behind the system was that a user might have a range of different sound sources around his/her head playing in the background but when required a nod would bring a source to the centre of attention. An evaluation of this interaction was undertaken whilst users were walking. A wide range of usability measures was taken, from time and error rates to subjective workload, percentage preferred walking speed and comfort. These showed that such an interaction was effective and users could easily make selections of auditory objects when on the move. It also showed that egocentric positioning of sound sources allowed faster interactions but with higher error rates than exocentric positioning. This shows that a proactive system that used sound (and gestures) in this way could be used whilst the user was mobile. Work is progressing on the development of more sophisticated interactions in a 3D audio space [16].

### 3.2 Vibrotactile displays

Vibrotactile displays are another possibility for non-visual output. They have been very effective in mobile telephones and personal digital assistants (PDAs), but their displays are crude, giving little more than an alert that someone is calling. The sense of touch can do much more. As Tan [19] says "In the general area of human-computer interfaces … the tactual sense is still underutilised compared with vision and audition". Our cutaneous (skin-based) sense is very powerful, but has been little studied in terms useful for proactive computing. This has begun to change as more sophisticated devices are now easily available that can be used on mobile devices (see Figure 3). Tactile displays have an advantage over audio ones in that they are private, so others around you cannot hear the information being presented,

Recent work has started to investigate the design of tactile icons, or Tactons. These are structured vibrotactile messages that can be used alongside audio or visual displays to extend the communication possibilities [5, 8]. The key parameters of touch that can be used to encode information are: waveform, rhythm and body location. Brown *et al.* [8] have shown that information can be encoded into Tactons in the same way as in Earcons, with the same levels of recognition. Brewster and King [6] have shown that Tactons can successfully provide information about the progress of tasks.

This is important as it means that progress and status information can be delivered using this modality without requiring the visual attention of the user.

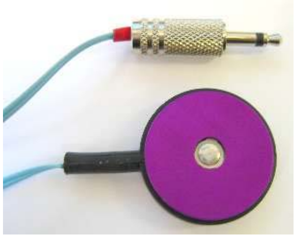


**Figure 3: An Engineering Acoustics Inc. C2 tactile display (www.eaiinfo.com).**

Tactile displays can be combined with audio and visual ones to create fully multimodal displays. There are interesting questions about what type of information in the interface should be presented to which sense. Tactons are similar to Braille in the same way that visual icons are similar to text, or Earcons are similar to synthetic speech. For example, visual icons can convey complex information in a very small amount of screen space, much smaller than for a textual description. Earcons convey information in a small amount of time as compared to synthetic speech. Tactons can convey information in a smaller amount of space and time than Braille. Research will show which form of iconic display is most suitable for which type of information. Crudely, visual icons are good for spatial information, Earcons for temporal. One property of Tactons is that they operate both spatially and temporally so they can complement both icons and Earcons. Further research is needed to understand fully how these different types of feedback work together.

## 4. Users with a range of abilities

Proactive computing using multimodal interaction offers many new possibilities for people with disabilities. These may be physical disabilities or disabilities caused by the environment or working conditions. For example, the multimodal displays described above are valuable for visually-impaired people as they do not use visual presentation. They can also be effective for older adults; Goodman *et al.* [12] showed that older users could perform as well as younger ones in a mobile navigation task when multimodal displays were used on a handheld computer. Another advantage of multimodal displays is that information can be switched between senses. So someone with hearing loss could use a tactile and visual display, whilst someone with poor eyesight could use a tactile and audio one to access the same systems and services. These advantages also apply to physically able users who are restricted by environment (for example, bright sun makes visual displays hard to use, loud background noise makes audio input and output impossible) or clothing (jobs requiring gloves or goggles make it hard to use keyboards or screens). Information can be switched to a different modality as appropriate to allow users to interact effectively.

## 5. Discussion and Conclusions

This paper has presented a range of input and output techniques using different sensory modalities. One of the key issues for interaction with proactive computer systems is that computing takes place away from the office and out in the field [20]. This causes problems for standard interaction techniques as they are not effective when users are on the move. Using different senses for input and output can avoid some of these problems. Our different senses are all capable of different things and interaction designers can take advantage of this to create suitable interactions. This is also dynamic as users out in the field will be subject to changing environments and tasks. Good proactive interface design will allow interaction to move between different techniques and senses as situations change.

Evaluating interfaces to proactive systems has had little attention. New techniques will need to be developed to allow us to test the sophisticated interactions we need to develop in realistic usage scenarios. At Glasgow we have begun to develop a battery of tests to allow us to evaluate mobile and wearable devices in mobile but controlled conditions so that we can discover if our new interaction designs are successful or not [4, 7, 17].

As Ljungstrand *et al.* suggest, there are many questions to be answered before we can construct effective user interfaces to proactive computing systems and much research is still needed. However, we can see that multimodal displays are a key part of these interactions. Using gestures, for example, is a good way to allow flexible, dynamic input whilst the user is involved in other tasks. Gestures do not need a visual display and many different parts of the body can be used to make gestures so they can be effective even if the hands are busy. Feedback through audio or tactile displays offer solutions when visual displays are not possible. The combination of all three types of display can be very powerful. We have also seen that when we deliver feedback and expect input can have significant effects on users in terms of selection accuracy and movement. If we force them to attend to information and make input when it is not suitable then there may be consequences for the primary task in which they are involved.

## Acknowledgements

This work was funded by EPSRC Advanced Research Fellowship GR/S53244.

## References

1. Barfield, W. and Caudell, T. (eds.). *Fundamentals of wearable computers and augmented reality*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2001.
2. Blattner, M., Sumikawa, D. and Greenberg, R. Earcons and icons: Their structure and common design principles. *Human Computer Interaction*, *4* (1). 11-44.
3. Brewster, S.A. Chapter 12: Non-speech auditory output. In Jacko, J. and Sears, A. eds. *The Human Computer Interaction Handbook*, Lawrence Erlbaum Associates, 2002, 220-239.

4. Brewster, S.A. Overcoming the Lack of Screen Space on Mobile Computers. *Personal and Ubiquitous Computing*, *6* (3). 188-205.
5. Brewster, S.A. and Brown, L.M., Tactons: Structured Tactile Messages for Non-Visual Information Display. In *Proceedings of Australasian User Interface Conference 2004*, (Dunedin, New Zealand, 2004), Austalian Computer Society, 15-23.
6. Brewster, S.A. and King, A.J., The Design and Evaluation of a Vibrotactile Progress Bar. In *Proceedings of WorldHaptics 2005*, (Pisa, Italy, 2005), IEEE Press.
7. Brewster, S.A., Lumsden, J., Bell, M., Hall, M. and Tasker, S., Multimodal 'Eyes-Free' Interaction Techniques for Wearable Devices. In *Proceedngs of ACM CHI 2003*, (Fort Lauderdale, FL, USA, 2003), ACM Press, Addison-Wesley, 463-480.
8. Brown, L., Brewster, S.A. and Purchase, H., A First Investigation into the Effectiveness of Tactons. In *To appear in Proceedings of World Haptics 2005*, (Pisa, Italy, 2005), IEEE Press.
9. Cohen, M. and Ludwig, L.F. Multidimensional audio window management. *International Journal of Man-Machine Studies*, *34*. 319-336.
10. Gaver, W. The SonicFinder: An interface that uses auditory icons. *Human Computer Interaction*, *4* (1). 67-94.
11. Geelhoed, E., Falahee, M. and Latham, K. Safety and comfort of eyeglass displays. In Thomas, P. and Gellersen, H.W. eds. *Handheld and Ubiquitous Computing*, Springer, Berlin, 2000, 236-247.
12. Goodman, J., Brewster, S.A. and Gray, P.D. How can we best use landmarks to support older people in navigation? *Behaviour and Information Technology*, *24* (1). 3-20.
13. Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C. and Want, R., Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In *Proceedings of ACM CHI'98*, (Los Angeles, CA, 1998), ACM Press Addison-Wesley, 17-24.
14. Ljungstrand, P., Oulasvirta, A. and Salovaara, A., Workshop Forward. In *Workshop 6: HCI Issues in Proactive Computing (Workshop at NordiCHI 2004)*, (Tampere, Finland, 2004), iv-v.
15. Malkewitz, R., Head pointing and speech control as a hands-free interface to desktop computing. In *Proceedings of ACM ASSETS 98*, (Marina del Rey, CA, 1998), ACM Press, 182-188.
16. Marentakis, G. and Brewster, S.A., A Study on Gestural Interaction with a 3D Audio Display. In *Proceedings of MobileHCI 2004*, (Glasgow, UK, 2004), Springer LNCS, 180-191.
17. Pirhonen, A., Brewster, S.A. and Holguin, C., Gestural and Audio Metaphors as a Means of Control for Mobile Devices. In *Proceedings of ACM CHI 2002*, (Minneapolis, MN, 2002), ACM Press, 291-298.
18. Sawhney, N. and Schmandt, C. Nomadic Radio: speech and audio interaction for contextual messaging in nomadic environments. *ACM Transactions on Human-Computer Interaction*, *7* (3). 353-383.
19. Tan, H.Z. and Pentland, A., Tactual Displays for Wearable Computing. In *Proceedings of the First International Symposium on Wearable Computers*, (1997), IEEE.
20. Tennenhouse, D. Proactive Computing. *Communications of the ACM*, *43* (5). 43-50.
21. Want, R., Pering, T. and Tennenhouse, D. Comparing autonomic computing and proactive computing. *IBM Systems Journal*, *42* (1). 129-135.

# Design Ideals in Proactive and Ubiquitous Computing

Matthew Chalmers

Computing Science, University of Glasgow, Glasgow, UK
matthew @ dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~matthew

**Abstract.** A substantial amount of design work in the area of ubiquitous computing and proactive systems is based on the ideal of 'transparency' in system use, as set out by Weiser in the early days of ubicomp. This paper points out ways that this ideal is incomplete or unachievable, by considering both the theory that Weiser drew from and the pragmatics of systems' use. It also presents some system designs that exemplify more realistic design ideals.

## 1 Introduction

Context and awareness are at the core of proactive, context–aware and ubiquitous computing. The design ideal of *transparency* means that a system should represent the context of the user and respond to it in a way that does not demand the conscious awareness of the user. Within ubicomp, this ideal has existed since Weiser's foundational work. As he put it in [26]:

> A good tool is an invisible tool. By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not the tool.

Weiser's ideal was inspiring, and has supported a powerful and productive shift away from system design that makes excessive demands on users. Nevertheless, as described in more detail in [9] and [10], this design ideal is unachievable or incomplete, does not fit well with the theory that it stems from, and is being extended to form more complex but more realistic design concepts.

Other areas of computing, such as human–computer interaction (HCI) and computer–supported cooperative work (CSCW), also treat context and awareness as central—even though they use different interpretations with regard to theoretical principles and design practice. CSCW focuses on intersubjective aspects of context, constructed in and through the dynamics of each individual's social interaction. It tends to defend against reductionism and objectification. In contrast, proactive and ubiquitous computing generally concentrate on computational representations of context that span and combine many senses and media—with little attention paid to the social construction of context in interaction. In the introductory article of a recent special journal issue on context–aware systems, Dey et al. describe context as "typically the location, identity and state of people, groups and computational and physical objects" [12].

Such definitions are common in proactive and ubiquitous computing, but they do tend to favour objective features that can be tracked and recorded relatively easily. They often ignore or avoid important aspects of the user experience, such as subjectively perceived features and the way that past experience of similar contexts may influence current activity—aspects that are central concerns of CSCW.

This kind of discussion and this kind of dichotomy have appeared before in HCI and CSCW, and it would seem appropriate for proactive computing to draw from that experience. There is a long–standing discourse on the conflict between the infinite and subjective detail of social interaction, and the finite and objective aspects of systems design. One key concern has been how systems can represent work and its context without over–formalising, over–simplifying and over–objectifying it. A canonical example in CSCW is the attempt by Winograd and Flores to make theoretical discussion and system design inform each other in the workflow design approach, as implemented in a system called The Coordinator [29]. The Coordinator was essentially an email tool in which the system supported one's work not only by presenting the content of each document for editing, but by presenting the document's context within a flow or temporal pattern of social interactions, such as a request from someone for its creation and the promise to deliver it to someone else once complete. Workflows were thus 'conversations for action' built from a pre–designed categorisation of work interactions. The system gave users an explicit representation of the process of work as well as the documents, spreadsheets, reports and other artifacts handled and constructed within the work process.

Winograd and Flores drew on a number of experiences and theories, but central among these was the hermeneutic philosophy of Martin Heidegger [19, 20]. In particular they focused on Heidegger's phenomenology and ontology, in which human activity is treated as an ongoing temporal process of language and interpretation, rather than a series of separable perceptions, each of which frames and fixes the world as a set of symbols or signs. He (and they) treated language as activity, and activity as language, i.e. language was seen not merely as a mode of representing things, but as a mode of doing things. By formalising and making explicit the temporal flow of such actions, Winograd and Flores aimed to make work 'present–at–hand', in that people in an organisation would use the workflow as a way to consciously focus on their work, rationalising it and making it more efficient.

Such workflow–like representations of activity are being brought into ubiquitous computing. In Activity-Based Computing [5,11], a direct connection is made between context–aware systems design and the formal models of activity in workflow and Activity Theory [21]. In their healthcare systems for hospitals, patient treatment is organised and managed through a set of defined tasks or activities that have been decided upon by all clinicians. Each clinician's work activity is represented as a heterogeneous collection of computational services, and such services are made available on various stationary and mobile computing devices. A related system design approach is the 'task driven computing' approach of the Aura system [18], in which tasks are 'explicit representations of user intent' constructed out of 'coalitions of abstract services' within the system.

However, such representations of activity have a potential danger, namely "that their design is predicated entirely by formal procedures—ignoring (and even damaging) the informal practice" [4], and this leads to a paradox or tension that Bardram summarises well:

> On the one hand, due to the contingencies of the concrete work situation, work has an ad hoc nature. Plans are not the generative mechanisms of work, but are 'merely' used to reflect on work, before or after. On the other hand, we find that plans, as more or less formal representations, play a fundamental role in almost any organisation by giving order to work and thereby they effectively help getting the work done.

Such pre–designed formal categorisations and representations of work can be useful as resources for action, and as resources for accounting for one's action, but tightly structured representations of work can be problematic. A good proportion of mainstream CSCW researchers have focused on revealing the same detail of socially–constructed situated action that is excluded from these representations. The designer of a proactive system's task models should consider whether they are designed with the intention or assumption of being carried out with script–like consistency, instead of being seen as flexible map–like resources for the situated action of users [23], how much work is needed to make their use fit with the use of other work media, beyond the proactive system, that may not be easily tracked or controlled by that system [6], and whether they fully represent the dynamics, detail and articulation of users' intents and priorities [3, 22].

At the root of such valid criticisms is a tension or paradox described clearly by Dourish [14]: "how can sensor technologies allow computational systems to be sensitive to the settings in which they are used, so that, as we move from one physical or social setting to another, our computational devices can be attuned to these variations?" He also points out that the design practices of proactive and ubiquitous computing rely on objective or positivist notions of context, even though their ideals are bound up with subjective, social and phenomenological notions. Dourish refers to the Weiser's Scientific American article [25], which uses the work of social anthropologists such as Lucy Suchman and Jean Lave, and hermeneutic philosophers such as Martin Heidegger and Hans–Georg Gadamer [16, 19, 24]. Similarly, he refers to [1], which cites activity theory, situated action, distributed cognition and ethnographic studies as important resources for ubiquitous computing.

In the following section, we focus on the theoretical ideals for ubiquitous and context–aware computing that Weiser presented—as well as later researchers such as Dourish and Abowd et al. We focus on the underlying theories, assumptions and priorities of ubicomp and proactive systems design, in particular the ideal of transparent or invisible system use. We increasingly concentrate on the way that, over time, that mode of use changes: it depends on and is interwoven with rationalising conscious activity. We will, admittedly, neglect an important topic: the future, as evinced in plans and expectations, as we concentrate on the past as a resource for the present. A final short section of the paper shifts our attention from theory to system design practice, drawing on the earlier theoretical section as well as the system design work of the author and his collaborators in Equator (www.equator.ac.uk).

## Transparency as an Unachievable or Incomplete Ideal

Objective and subjective are bound together by histories of use and activity, and this is central to Heidegger's concept of the transparency of a tool or technology. Weiser used this as a core concept when laying the foundations of ubiquitous computing, and his ideal was for the systems we design to be "literally visible, effectively invisible" or, we suggest, objectively visible but subjectively invisible.

An old example from Heidegger is the way that a skilled carpenter engaged in his work acts through the hammer, focusing on how it changes and is combined with other tools and materials, rather than focusing on the hammer in itself. Heidegger called this transparent, practically engaged and non–rationalising use 'ready–to–hand', in contrast to the rationalising, objectifying and analytical activity he categorised as 'present–at–hand'. He saw both modes of use as being set within the ongoing circular process of interpretation, in which one is influenced by understanding and past experience of tools and media when using any tool or medium. One's use of a new tool (or a new use of a tool) in the course of everyday, situated and social interaction, combining it with the heterogeneous others used in everyday life, adapts experience and understanding—that will affect how one acts and interprets in the future. In time, this process of accommodation and appropriation lets one focus on the use of the tool, and not on the tool in itself, thus making the tool 'disappear' as Weiser later discussed.

Weiser called for a move towards design of interactive systems that have a better fit with everyday human activity, understanding and interaction, and with the practically engaged and non–rationalising mode of activity characteristic of much of (but not all of) everyday activity. Weiser focused on raising our awareness of embodied interaction, i.e. the interpretation and use of a system by a user in a ready–to–hand way. However, in moving away from traditional systems design, Weiser focused almost entirely on design to support embodied or ready–to–hand interaction. Following writers such as Schutz, Garfinkel and Suchman, he did not fully address the relationship between the two modes. In particular, how does a tool become transparent and ready–to–hand?

Heidegger, and his successors such as Gadamer and Ricoeur, held that situations where a tool becomes present–at–hand are crucial to the individual's learning and to the differences between individuals. The ongoing 'feedback loop' of interpretation and understanding integrates these two modes, and social interaction affords variation in people's understanding as well as consistency in their behaviour. For example, creativity can be considered as the variation of an individual's subjective understanding from his or her prior understanding and from others'. The individual may be very conscious of his or her own activity, rationalising it and very aware of it, i.e. the system or tool is present–at–hand, for a while.

A most important situation here is the accommodation and appropriation of a new technology into a setting or community of use. As pointed out in [8] and [22], a system, like any formal and finite construct, necessarily involves under–specification of the situation of its use, and therefore openness to interpretation and variability of its normative effect. This allows the individual user to conform to a script–like pattern of actions, or to treat the system as flexibly interpreted, map–like resources for situated action. People accommodate the characteristic affordances of a new tool, but they may also appropriate it to suit and adapt the practices and priorities of their own contexts and communities of use i.e. other, older tools and media, and other people. With experience of its use, the tool may become understood and familiar to the individual, i.e. more ready–to–hand, embodied and transparent. Similarly, as people perceive one another's use, with each interpreting and reacting to the others, they can achieve intersubjective consistency of behaviour—consistent with each other, but not necessarily with the use expected by the designer. A use or activity that is new and present–at–hand for one of them can thus become transparent and ready–to–hand for all. The circular process of interpretation, whereby perception and activity are influenced by understanding and experience, but also feeding into and changing them, relies on the interplay between ready–to–hand and present–at–hand interpretation.

Transparent interaction, as Weiser made clear, is an aspect of human activity that was under–emphasised in HCI. Nevertheless, ready–to–hand transparent interaction and present–at–hand objectification are interdependent—and Weiser did not address this. In simpler terms, the context of use at any time is founded on both objective and subjective interpretation, with each influencing the other over time. We have to expect that a new technology will be to some degree present–at–hand, no matter how well the designer aims towards embodied or ready–to–hand interaction. This is just one of a number of modes of use and interpretations that Weiser never fully dealt with.

Dreyfus, summarising Heidegger, suggests three categories or modes of present–at–hand activity, which we label here as breakdown, analysis and contemplation [15, p. 124]. In the case of breakdown, one might continue with a different tool, use deliberation to eliminate the disturbance in the original tool, or stop because one can determine no way to continue with it. In breakdown, the affordances of even the most familiar tool may significantly differ from those of everyday ready–to–hand use e.g. when the head of the carpenter's hammer becomes loose, so that one becomes very aware and conscious of it, and the difficulty of progressing with normal use. Another example might be the breakdown that occurs with a mobile phone when it loses its network signal: one's attention may turn from a conversation 'through' the phone and its infrastructure to the tool itself.

A second mode of present–at–hand activity is analysis, for example skilled scientific activity, observation, experimentation, theoretical reflection or even wonder. Use of the tool or system is not transparent, and one cannot "focus on the task not the tool" because the task is to focus on the tool. The carpenter may work on the hammer, to fix it; the phone user may focus on the signal strength indicator, waiting or moving until he or she regains a signal; a researcher may study how a new mobile technology works in use; and an individual user may explicitly use a visualised representation of system structure so as to adapt it. A third form of present–at–hand activity is contemplation, which covers the cases in which one may be finished with a tool or resting from using it, and be engaged in less analytical reflection or curiosity towards it.

Dreyfus did not explicitly cover or address activity that was skilled and conscious, but social. In doing so, he accurately summarised Heidegger's tendency to narrowly focus on the individual. One of the ways that Heidegger's successors, particularly Gadamer and Ricoeur, advanced hermeneutics was to fit such social interaction into

Heidegger's framework [19, 24]. The ongoing 'feedback loop' of activity, interpretation and understanding also serves to integrate the different modes of use by different people. Social interaction affords consistency in people's understanding and behaviour, as well as inconsistency as they accommodate the characteristic affordances of a new tool, and appropriate it to suit the practices and priorities of their own contexts and communities of use i.e. other, older tools and media, and other people. One might then usefully define a fourth mode of present–at–hand activity, self–presentation, based on skilled, social activity involving conscious consideration of how a tool mediates one's activity i.e. presents one's activity to others. Examples include when one consciously considers how to demonstrate or teach the use of a tool to someone else; avoiding a web site or a surveillance camera, or turning off an active badge, because it may lead to one's activity being presented to others in embarrassing or invasive ways; and using GPS logs to spell out a name on a cartographic scale, as in GPS drawing (www.gpsdrawing.com), so that others can see where and how one moved.

Each of these categories of use and interpretation, i.e. transparency, breakdown, analysis, contemplation and self–presentation, is influenced by prior activity and experience, and also influences later activity. In other words, each affects and is affected by the others. This hermeneutic circle, whereby perception and activity are influenced by understanding and experience but also change understanding and extend experience, is thus an abstract description of the historical process that makes these different modes of use interdependent.

The most profound technologies may be those that disappear, as Weiser said at the start of his Scientific American article, but it may be clearer now that they do not weave themselves into the fabric of our everyday life. Instead, we weave them into our lives, in and through our use and activity. Disappearance happens through the process of coupling and contextualisation i.e. the circle of interpretation, action and experience that weaves together both ready–to–hand and present–at–hand uses of a tool by people over time. The objectifying use of tools and information is a constraint, influence and a resource for new forms of interaction, for sharing and learning, and is a precursor and foundation for transparent everyday use. Similarly, transparent use builds experience and understanding that are influences and resources for objectifying, rationalising use. It seems that a degree of care has to be taken when treating embodied interaction, disappearance and invisibility as an ideal for proactive and ubiquitous computing.

Weiser suggested that even a "glass TTY UI can be ubicomp" if its use is well woven into the fabric of people's collaboration and interaction [27]. Again, this may seem contradictory to the common notion of ubicomp and proactive systems, involving technologies such as location sensors, mobile displays and wireless communication, but Weiser was clear that it was not the technology in itself that made for ubicomp. Instead he suggested that we should aim for and support the accommodation and appropriation of computing into everyday life, so that its use is non–rationalised, intersubjective and interwoven with our use of other media. What he perhaps did not fully deal with was the way that rationalised, objectifying and focused activity is necessary to the process of achieving his ideal, and therefore that his ideal is unachievable or incomplete without complementary modes of activity. A challenge for system designers is, therefore, how to design systems that reflect this broader view of context and activity, and which use history and time to interweave different modes of use, media and people so as to support the accommodation and appropriation of computing into everyday life.

## From Transparency to Analysis and Back Again

Hermeneutics impresses upon us the interdependence of transparent or ready–to–hand use and analytic or present–at–hand use. It also makes clear that the significance of individuals' histories as a part of context. If one accepts this broader view of use, context and interpretation, what changes in system design principles arise? One issue for designers of proactive systems to consider is that long–term use of a system is likely to include focused, rationalising, present–at–hand use: breakdowns, analysis, reflection and self–presentation. Ideally, we might make a system in which ongoing system execution—including any system adaptation—is so well–coupled with use that its users never have to rationalise about it, focus on it, explain its use to others, or explicitly approve any adaptations. However, let us take a realistic view of this ideal situation: it will never happen.

Accepting that users will sometimes focus on and rationalise about a tool should not be taken as a reason or excuse to make a tool that they *always* have to focus on, in order to use it at all. Instead, we suggest that systems support rationalising present–at–hand use in ways that feed into and aid the process of understanding and experience that supports later ready–to–hand transparent use. We should treat system design that affords episodes of objectification of use as supporting the ideal of ubiquitous computing, rather than contrary to it—but only if the effect of those episodes is to make the system better woven into everyday life and embodied interaction.

For example, at Glasgow we are working with histories of system use, often generated by combining user activity logs (e.g. web proxy logs, GPS/location logs from PDAs) and instrumentation of software (e.g. print statements for debugging, system logs of components being loaded and methods being called). We began by building a tool to combine logs from PDAs and the server in a 'seamful' game [7], and overlaying the street map with data on game events and system log data so as to visualise or 'replay' the game (Figure 1). Depending on the data selected, one can present past games more from a player's perspective or from a system designer's perspective.

We are pursuing design that works with the way that the histories and patterns of use of a system are important for the programmer in understanding how to change or adapt the code structure, and vice versa: the code structure affects and constrains what histories and patterns of use can arise. Similarly, the histories and patterns of use of ubicomp systems are important for the user in understanding how to change or adapt his or her use. Such analytical use can help players pick up good tactics, see how badly certain others play, and redesign their tactics. This work is intended to blur the boundary between use and design, or between use and redesign of system components.

We would like to make the work of debugging and instrumenting more part of the interaction design for users too, so that they can control more of who, where, when and what is logged and analysed—because logging affects system performance, and is a

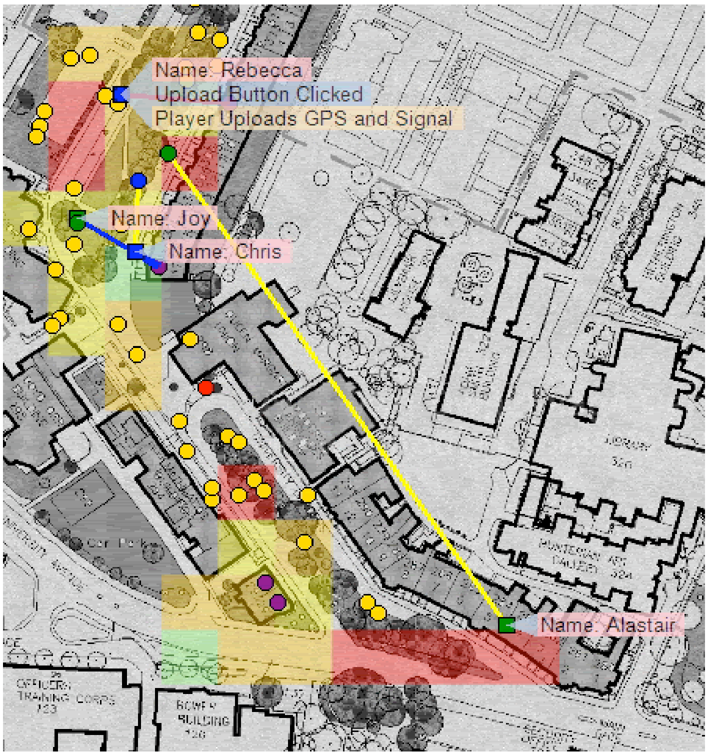


*Figure 1.* Log data from PDAs and servers in a mobile game is combined in the Replayer tool so as to reveal system use to users. Players and developers can analyse game tactics and system performance. Replaying often reveals discrepancies between the server's model of activity, and players' activity as tracked via their PDAs, e.g. the current GPS position (bottom right) of the player *Alistair* is far from his position shown on players' maps (top left), i.e. the last the server received. Such differences and delays can be a resource for game tactics and for new design.

means of self–presentation to others. By blurring the distinction between system use and system design, we also reflect and support the way that an increasing number of large collaborative systems are designed, redesigned and modified by people who use them, in particular the 'modding' community of game players. It also reflects the way that games can be designed to have useful side effects for non–players [2].

In summary, we suggest a pragmatic design response to the inevitability and importance of present–at–hand use, informed by an understanding of the effect of history in context and use, and the interdependence of objective and subjective modes of interpretation. We should design for such use rather than ignoring it. We suggest that there are practical ways to support people's interweaving of present–at–hand use and ready–to–hand embodied interaction. Temporal patterns and structure in embodied interaction can feed into and be resources for later use. A ubicomp or proactive system should support people in occasionally rationalising, focusing on and abstracting over the system 'in itself' along with its past use, so that they might adapt our systems so as to better feed into and be a resource for their use in their contexts for their purposes.

## Acknowledgements

This paper is a modified version of sections of [9]. This work was carried out within Equator, funded by the UK EPSRC (GR/N15986/01).

## References

1. Abowd, G., Mynatt, E., and Rodden, T. (2002): The Human Experience, IEEE Pervasive Computing, Jan-Mar, 48-57.
2. von Ahn, L. & L. Dabbish (2004): Labeling Images with a Computer Game, Proc. ACM CHI 2004, pp. 319–326.
3. Bannon, L. & S. Bødker (1997): Constructing Common Information Spaces, Proc. ECSCW, Lancaster, pp. 81–96.
4. Bardram, J. (1997): Plans as Situated Action: An Activity Theory Approach to Workflow Systems. Proc. ECSCW, Lancaster, pp. 17–32.
5. Bardram, J., Kjær, R. & Pedersen M. (2003): Context-Aware User Authentication—Supporting Proximity-Based Login in Pervasive Computing, Proc. Ubicomp 2003, pp. 107-123.
6. Bowers, J et al. (1995): Workflow from Within and Without: Technology and Cooperative Work on the Print Industry Shopfloor, Proc. ECSCW, 51-66 1995.
7. Chalmers, M. et al. (2003): Seamful Design: Showing the Seams in Wearable Computing, Proc. IEE Eurowearable, Birmingham, pp. 11-17.
8. Chalmers M. (2003): Awareness, Representation and Interpretation, J. CSCW 11:389–409.
9. Chalmers, M. A Historical View of Context, J. CSCW 13(3) (2004) 223–247.
10. Chalmers M., Galani, A. Seamful Interweaving: Heterogeneity in the Design and Theory of Interactive Systems. Proc. ACM Designing Interactive Systems (DIS2004), Boston, (2004) 243–252.
11. Christensen, H., & J. Bardram (2002): Supporting Human Activities—Exploring Activity-Centered Computing, Proc. Ubicomp 2002, Göteborg, pp. 107–116.
12. Dey, A., G. Abowd & D. Salber, (2001): A conceptual framework and a toolkit for supporting the rapid prototyping of context–aware applications, Human Computer Interaction, pp. 97-167.
13. Dourish, P. (1995): Developing a Reflective Model of Collaborative Systems, ACM Trans. CHI, 2(1), 40–63.
14. Dourish, P. (2004): What We Talk About When We Talk About Context, Personal and Ubiquitous Computing 8(1), pp. 19–30.
15. Dreyfus, H. (1991): Being–in–the–World: A Commentary on Heidegger's Being and Time, Division I, MIT Press.
16. Gadamer, H.–G., (1989): Truth and Method, 2nd edn., trans. J. Weinsheimer & D. Marshall, Crossroad. (Original published in 1960.)
17. Garfinkel, H. (1967): Studies in Ethnomethodology. Prentice Hall.
18. Garlan, D. et al. (2002): Project Aura: Toward Distraction-Free Pervasive Computing, IEEE Pervasive Computing, April-June 2002, pp. 22–31.
19. Grondin, J. (1994): Introduction to Philosophical Hermeneutics, trans. J. Weinsheimer, Yale University Press.
20. Heidegger, M. (1962): Being and Time, Harper & Row. (Original published in 1927.)

21.Nardi B. (ed.) (1996): Context and Consciousness: Activity Theory and Human–Computer Interaction, MIT Press.

22. Schmidt, K. (1997): Of Maps and Scripts: The Status of Formal Constructs in Cooperative Work, Proc. ACM Group 97, Phoenix, pp. 138–147.

23.Suchman, L. (1987): Plans and Situated Actions: The Problem of Human Machine Communication, Cambridge University Press.

24. Warnke, G. (1987) Gadamer: Hermeneutics, Tradition and Reason, Stanford University Press.

25.Weiser, M. (1991): The Computer for the Twenty-First Century, Scientific American, 94-110, Sept. 1991.

26. Weiser, M. (1994a): The world is not a desktop. Interactions; January 1994; pp. 7-8. ACM Press.

27. Weiser M. (1994b): Creating the invisible interface: (invited talk). ACM Conf on User Interface Software and Technology (UIST94), p.1.

28. Wilson, T. (1983): Qualitative "versus" quantitative methods in social research, Dept of Sociology, U. California at Santa Barbara.

29.Winograd, T. & F. Flores (1986): Understanding Computers and Cognition, Addison Wesley.

# Computer Vision at CGG MSU

D. Ivanov, V. Lempitsky, A. Khropov, A. Shokurov, and Ye. Kuzmin

Computer Graphics Group
Department of Mathematics and Mechanics
Moscow State University
Vorobyovy gory, Moscow, 119 992 Russia
<denis,vitya,akhropov,anton,yevgeniy>@fit.com.ru

**Abstract.** This paper reviews the research activities in computer vision at Computer Graphics Group of Moscow State University. In particular, the results of our work on acquisition of 3D human head virtual models are presented. Also described is our experience with stereo reconstruction from photographs and registration of video sequences.

## 1 Introduction

Computer vision receives more and more attention in recent years as its application become more and more numerous. Computer Graphics Group has been carrying research in computer vision for several years. Most of our efforts were emphasized on elaborating techniques and methods suitable for concrete applications. Below we present the results achieved during three research projects.

The first one is concerned with the creation of virtual models of a head for a particular person. During the project we developed and implemented a bunch of techniques allowing for head modeling from digital photos or video sequences. Produced personalized head models are appropriate for animation, and this fact significantly augment the range of possible applications. Our project developed through several stages. Initially, as little as two photgraphs served as modeling input. The photographs were taken from front and profile angles. To produce the model, some user interaction was nescessary. On the second stage, we worked out a method basing on multiple images. Such modification significantly improved the visual consistency of the models. At the same time, the novel method demanded more user activity. On the final stage, we worked on head model acquisition from video sequences. In close cooperation with Intel Nizhny Novgorod Labs we developed a method for fully-automatic modeling. Created models can be used for object-based video compression and various virtual reality application.

Reconstruction from digital photographs is the topic of the second project. Produced highly-realistic models can be employed in such computer-vision applications as virtual reality (games, virtual tourism), computer-aided design, and non-intrusive metrology. Typical input for the algorithm constists of several images taken with an off-the-shelves digital camera. As our method requires significant user interaction, we investigated the possibilities of efficient assistance, based on image processing and visual geometry. Among objects amenable for reconstruction are architectural buildings and intérieurs, industrial objects, and many others.

The third project is dedicated to the registration of video sequences. Sophisticated tracking techniques as well as robust structure-and-motion methods were considered to achieve efficient registration. Among applications, which can be based on video registration, are augmented reality and video-based modeling.

# 2 Head Modeling

## 2.1 Problem Overview

The problem of the creation of realistic high-quality head models is one of the most complex problems of computer vision, which is not completely solved up to now. On the one hand, being a solid body located in 3D space a model of a particular head can be acquired by means of range scanners [2] or similar technologies. Such approach makes it possible to develop an accurate geometrical model and the corresponding texture; however, without additional processing, the model cannot be used for animation, and the subsequent adaptation of these data is a rather time-consuming procedure.

On the other hand, the use of a priori knowledge of the object structure (in the given case, a head) allows one to improve the model. This idea is usually implemented in the calibration strategies that use a generic head model and adjust it to the input data [3–6]. Alternatively, a set of various available models is considered to be a formal basis in some linear vector space [7]. The model obtained in this way is usually more appropriate for the animation purposes, since its construction is based on the knowledge of the head structure.

The head model calibration methods can be classified in terms of the type of data they operate on. Since the information about the scene depth is very expensive to obtain, digital images are usually used as the input data. There exist calibration technologies based on one image [7], two orthogonal images [4, 8], and a sequence of images or video [6, 9, 10]. Some methods require certain locations of cameras (viewpoints), certain lighting conditions, and other constraints, which considerably restrict the applicability domain of the corresponding algorithms.

An important feature of any calibration method is the degree of user participation in the model development. Some methods are completely automated [3, 7]; others require selection of several feature points on the image [9, 10]; however, the majority of the technologies suggest considerable user labor inputs [6, 8, 11]. The manual input of certain parameters often improves the model; however, the automated methods are more practical from the users standpoint.

### 2.2 Our Implementation

In this section our solution for the problem of personalized head model acquisition is briefly presented. More details can be found in [18, 1]. Our method has the following properties:

- a set of images or video sequence is used as the input data.
- the creation of a model relies on the knowledge of the model structure (a generic head model). In other words, the personalized model is obtained via *calibration* of the generic model.
- the model is constructed either automatically or with a minimal participation of the user.

The calibration of a head model under the above assumptions proceeds in several stages is organized in a pipeline, as shown on Fig. 1. At each stage, the data obtained at the previous stages of the pipeline are used.

Below, each stage is discussed in more details.

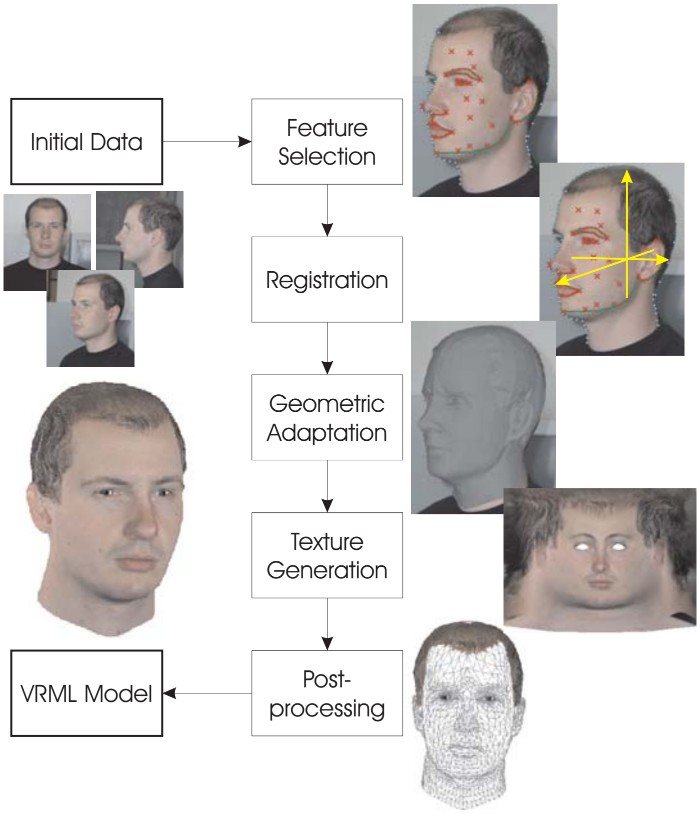


**Fig. 1.** Structure of the head model calibration pipeline

**Feature Selection.** Feature selection is the first and most important stage of the calibration pipeline, since the quality of the model greatly depends on the quality of data obtained as a result of the selection. The feature selection for an image or a video sequence identifies and selects certain elements of the head image for their subsequent use. These elements include the following objects:

- Feature points, such as corners of eyes and lips, lobes, etc., standardized in MPEG-4 for the animation of a head model in the framework of the synthetic video. [4]. Can be selected manually, or automatically in conjunction with feature contours.
- Feature contours, such as eye, lip, and face contours, profile nose contour, etc. Can be selected manually or automatically, using deformable templates [12–14]. Automatic selection was implemented in close cooperation with Intel Nizhny Novgorod Labs.
- Silhouette lines, such as a profile outline (if such is available on the image). Can be selected manually by user.
- Head masks (an image is segmented into a background and the head itself). Can be selected automatically using histogram methods.

In addition to the features listed above, for the case of video input a number of points on the skin are found and tracked through the video [15]. Since the data obtained in this way inevitably contain errors due to noises, model inaccuracies, and other similar effects, the subsequent use of these data for registration requires the use of statistically stable methods.

**Registration.** The objective of this stage consists in registering images or video frames and, in fact, estimating locations of cameras that were used to obtain these pictures in the global coordinate system. The elements selected at the previous stage are regarded to be the projections of the corresponding 3D objects on the picture planes with a pinhole camera. The problem consists in determining the orientation and position of each camera and its internal characteristics such as, for example, the focal length at the shooting moment.

The estimation of the parameters for each camera with simultaneous determination of the locations of three-dimensional points by their known projections is a wellstudied problem (*structure-from-motion*) [16].

In the given case, the solution of this problem is facilitated through the use of a priori knowledge of the camerawork (for example, it is known that the pictures were first taken almost full-face and then in profile) or the nature of some points (for example, certain points are known to correspond to points belonging to the human face, and, thus, their configuration in threedimensional space is approximately known).

Thus, each image is associated with a position and orientation of the camera in the three-dimensional space related to the model being reconstructed. Then it becomes possible to determine 3D coordinates of points of other, more complicated, objects. For example, two projections allow us to recover the 3D shape and the location of a contour.

If there are several registered images with head masks, it is possible to construct the intersection of mask cones in the three-dimensional space to get a *visual hull* of the head [17]. If there are many (several dozens) images with provided head masks, the resulting visual hull approximates the head shape everywhere except for concave regions, such as, for example, the region of the alae of the nose.

**Geometric Adaptation.** At this stage, a generic head model is placed into the global coordinate system and is subjected to deformations that adjust it to the given data. After this adaptation the projections of the required 3D elements of the model should be as close to the corresponding elements selected on the images as possible.

In our implementation, generic head model is supplied in the form of triangular mesh. We employ deformation with small number of DOF (e.g. affine deformation) to adjust the model globally.

To perform local adaptation, we formulate all geometric constraints in a form of linear equations on the displacement vectors of the vertices. Non-linear constraints are linearized. As a result, we get a system of linear equations on the coordinates of displacements. To enforce smoothness of tuning deformation and to propagate geometric knowledge to the regions of the mesh where no geometric constraints are available we augment our system with the linear equations that equate the displacement of each vertex to the weighted sum of displacements of the neighbour vertices. Being sparce and large resulting system of equation is solved in the least-quadratic sense by the conjugate gradients method. Resulting displacements are used to move the initial vertices (Fig. 2).

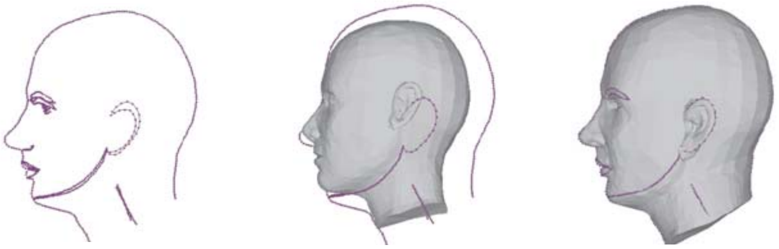


**Fig. 2.** Example of our deformation method. Left – 3D reconstruction of feature contours, middle – generic mesh after low-DOF deformation, right – adapted mesh

**Texture Generation.** The aim of this stage is to generate a texture map for the adapted (personalized) model. High quality texture map is vital for the realism of the model. Therefore, we investigated the problem of texture generation very thoroughly.

To generate a texture map we employ a texture mapping supplied with the generic head model. It defines a correspondence between surface points on the generic model and points in the unit square. Using inverse texture mapping we produce texture fragments that correspond to the frontal and the two profile views. (Fig. 3).

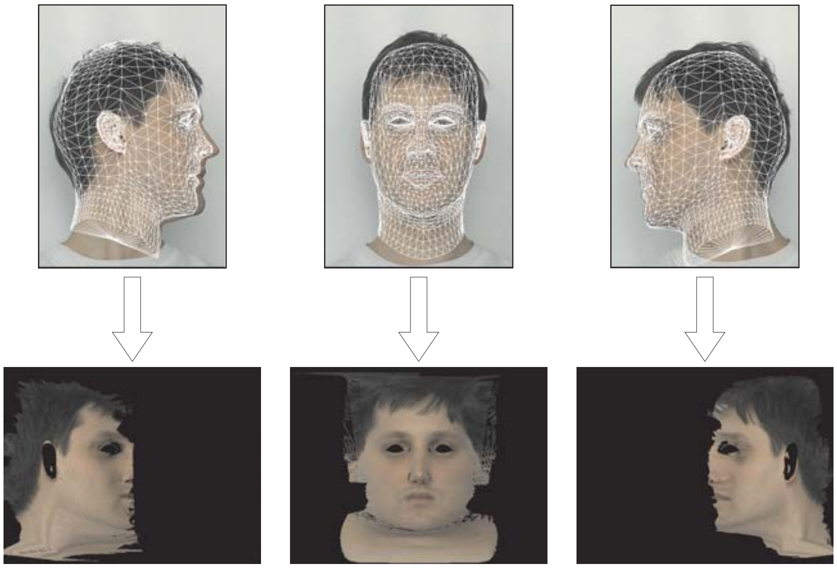


**Fig. 3.** Extraction of texture fragments using inverse texture mapping.

One possible way to combine the obtained fragments into a single map is to use the laplacian pyramid merging, which was initially proposed for image mosaicing [19]. This method decomposes each fragment into a set of images (pyramid levels), corresponding to different spatial frequences. Then weighted summation is applied to each level of the pyramyd. Resulting pyramid is recomposed into a combined map. This method is very efficient in coping with differences in lower frequencies of the fragments; however, it introduces significant smoothing when dealing with high frequencies, which reduces the texture resolution. Besides that, it is not local in the sense that every pixel of the image is affected by corresponding pixels in all initial fragments. Thus, if one of the fragment contains some artifacts, these artifacts will more or less corrupt the resulting map.

Another method for combining fragments is based on color balancing along optimal merging lines [20]. In this method the region of fragments' overlap are considered. Optimal merging lines passing from top to bottom of these regions are found using dynamic programming. The final texture is composed from three blocks with the shapes defined by the merging lines (Fig. 4). To eliminate the difference in colors along the lines the colors are balanced by adding to the left and the right fragments smooth color terms.

This method is local and do not introduce any smoothing. However, it doesn't deal well with low frequences. Therefore we developed a hybrid method, which efficiently employs strengths of both methods, producing texture maps superior to those of both previous methods (Fig. 5). For more details see [18].

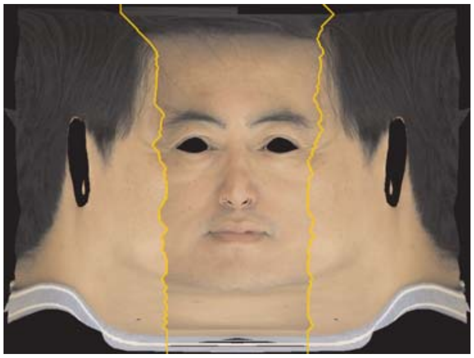


**Fig. 4.** Optimal merging lines for combining fragments divides the map into three parts. Each of the parts are taken from different fragments.

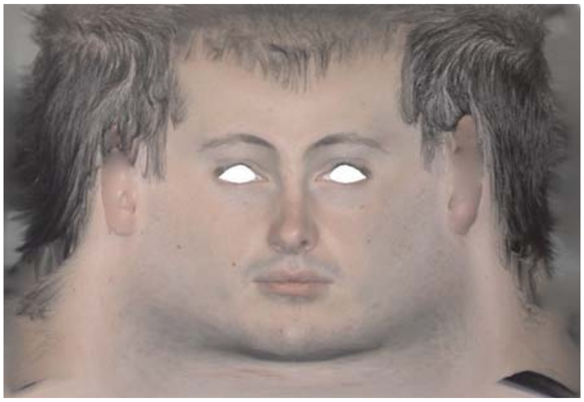


**Fig. 5.** Texture map produced by the hybrid method.

**Postprocessing.** Modeling of eyeballs as separate objects is necessary for realistic animation. Eyeball geometry and texture are processed on this stage.

Another head part that is impotant for model realism is hair. Since hairdos differ very much for different people, it is not reasonable to model the geometry of one by the calibration of a hypothetic generic hairdo using geometric data. Instead, hair model is created basing on the obtained texture map and then modified according to the observed geometry.

Postprocessing stage concludes the pipeline. Resulting personalized head model is shown on Fig. 6.

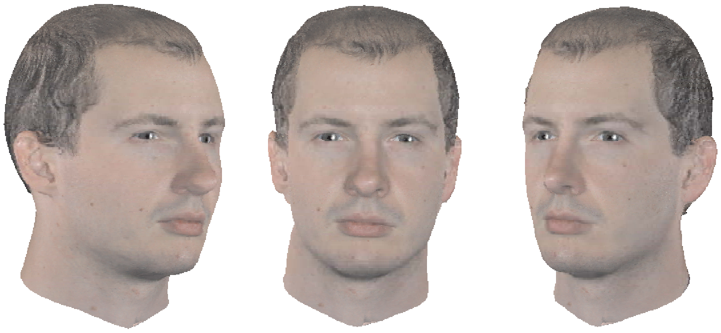


**Fig. 6.** Resulting head model, rendered from different angles.

### 2.3 Applications

The personalized head models produced with our method can be employed in a large number of applications.

While adapting generic model to the observed geometry our algorithm simply displaces vertices, thus preserving the whole structure of the generic triangular mesh. This fact allows for efficient animation, since the animation rules and algorithms developed for the generic model remain valid for the personalized ones.

In our case, the generic head model meets the requirements defined by MPEG-4 standard. This implies the correspondence of some of mesh vertices to the semantically meaningfull points on the human head (e.g. nose tip, lips corners, etc.). Therefore, any animation algorithm compliant with MPEG-4 standard can use our personalized models.

Our project was carried on in parallel with the project on MPEG-4 animation in Intel Nizhny Novgorod Labs [21]. As a result, we concluded with the full MPEG-4 pipeline [1]. Given a video camera, we can automatically acquire and animate the model of a person (alternatively, a digital still camera and a manual feature selection can be used). This strategy has the following applications:

- **Teleconferencing and videophones.** Consider a talking head sequence, that should be coded and transmitted through a narrow channel. Assume that the client side already has the personalized model of the speaker. Then only the animation parameters should be extracted from the input sequence and transmitted to the client side. The client side recieves the stream of animation parameters and animate the talking head according to it (Fig. 7).

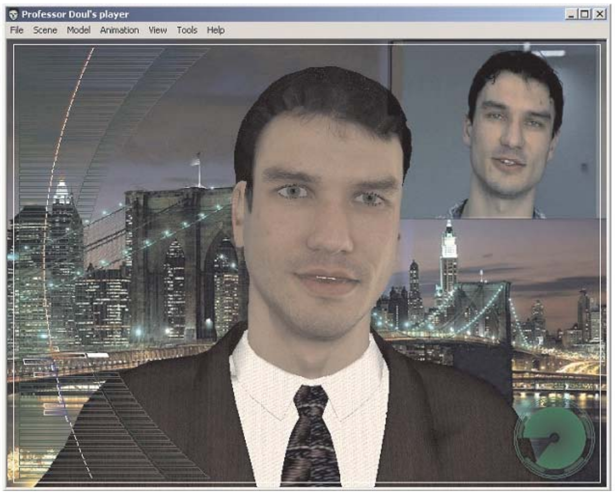


**Fig. 7.** Teleconferencing application prototype developed by Intel Nizhny Novgorod Labs. In the top right corner the input video stream is displayed. Client side animation based on the personalized model and a small number of transmitted animation parameters is shown in the main window.

- **Text- and speech-driven animation.** If the text- or speech-driven animation rules are given for the generic model then they could be used to animate a personalized model. This can be applied in internet messengers, computer games, and other virtual reality applications.

## 3 Stereo Reconstruction

### 3.1 Overview

Realistic image-based modeling has recently become one of the most important computer vision applications. In particular, the problem of interactive modeling from 2D photographs have been investigated in details (see e.g. [22, 23]). Several commercial systems capable of such interactive modeling are now available [24–26] (for review and comparison see [23]).

From the algorithmical standpoint, most commercial and non-commercial systems of the kind employ the recent multiview geometry results, collected and thoroughly discussed in [16]. Besides geometric issues, to produce good results such systems must perform an efficient optimization of a non-linear functional (so-called *bundle adjustment*), and this topic is covered by [27].

Below we describe our image-based modeling system, called ImagiCAD.

### 3.2 Implementation and Results

The input for our system consists of several photo images, taken with an off-the-shelves digital still camera. The only principal limitation is low amount of lens distirsion. Alternatively, one may preprocess the images with some distortion correction software.

The user select corresponding points on the images using a user-friendly interface. Common lines can be selected as well. Moreover, the user can point out the relations of incidence, parallelism or coplanarity for the selected points and lines.

The first and the most essential part of our system is point-based reconstruction, recovering structure (3D points' positions) and motion (camera parameters). There are two principal workflow schemes for the reconstruction; one for the case of uncalibrated cameras (i.e. cameras with unknown internal parameters) and the other for the case of calibrated cameras.

In the former case we first perform a projective reconstruction (i.e. reconstruction up to global projective transformation). This is done in a traditional manner: initializion from two view via fundamental matrix estimation is followed by sequential addition of cameras and points to the reconstruction. In the end bundle-adjustment is performed.

Resulting projective reconstruction is upgraded to near-euclidean reconstruction by autocalibration. Finally near-euclidean reconstruction undergoes another bundle adjustment process. Special term in the bundle adjustment functional forces near-euclidean reconstruction to drift towards euclidean reconstruction.

For the case of calibrated cameras, we perform reconstruction basing on essential matrices for different view pairs. Euclidean reconstruction is obtained directly here. Euclidean bundle adjustment is used to improve the result. Essential matrix computation is known to be unstable in the presence of significant noice. To solve this problem we again start with projective reconstruction and then perform noise reduction basing on the obtained projective reconstruction. This trick significantly increases the robustness of essential-matrix-based reconstruction.

In our system, the reconstructed 3D elements can be employed either for model construction or for metrology. To construct the model of the object the user interactively selects the set of points on any image. This set is triangulated and the obtained triangles textured with the corresponding image regions are added to the model. The model can be rendered with the internal OpenGL-based viewer [28]. Export in VRML format [29] is also available, making it possible to use the models in a multitude of applications (Fig. 8).

The second important application is image-based metrology. The user can interactively measure the distances and angles between reconstructed 3D ele-

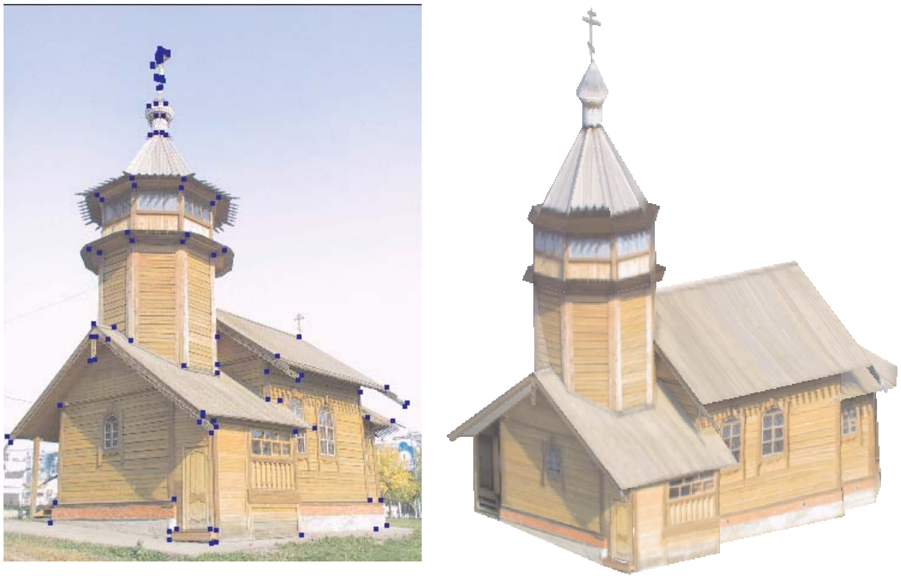


**Fig. 8.** A model produced with ImagiCAD. Left – one of the initial photographs with selected points. Right – the model.

ments. Very high metrology precision with tenths of percent order of relative errors can be attained.

## 4 Video Registration

### 4.1 Overview

In the previous sections some methods for 3D reconstruction have been described. In the case of head modeling we had a special generic model that was adapted to fit with the measured data. In the ImagiCAD system a user has to provide point and line correspondences. Fully automatic reconstruction of a general scene is a more challenging task especially taking into consideration the problem of detecting and establishing accurate correspondences between features (points and lines) in different images. We consider a continuously moving uncalibrated pinhole camera observing a static scene.

A huge amount of literature on this subject exists. One of the most important books that summarize the knowledge on this topic is [16].

There is an ongoing effort to create products of commercial quality in this sphere. One recent successful project for camera trajectory determination is Boujou [30] developed by 2d3 and used in production of some recent major film titles.

The algorithms being developed during the course of the project are incorporated in the open-source library O3RLib.

### 4.2 Overall pipeline

The overall pipeline (Fig. 9) consists of the following stages:

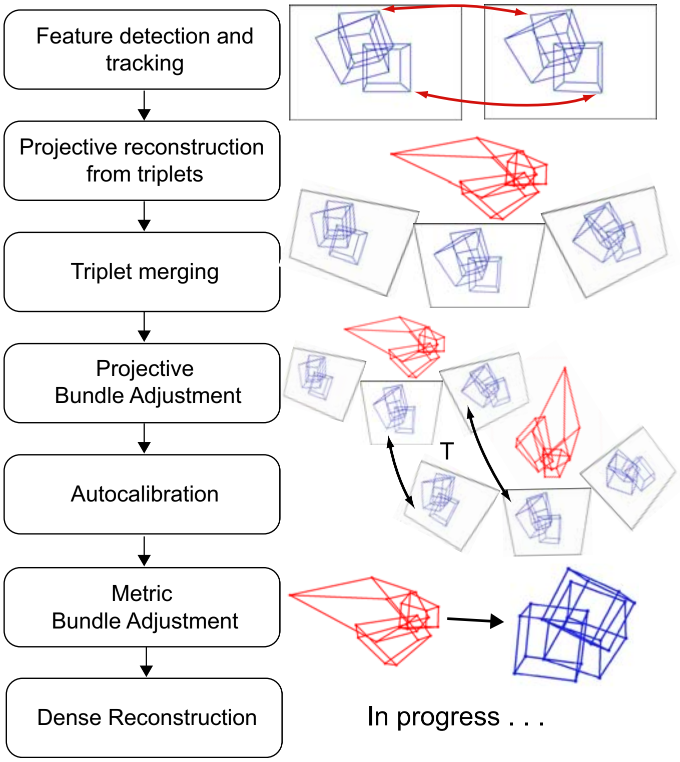

Feature detection and tracking
Projective reconstruction from triplets
Triplet merging
Projective Bundle Adjustment
T
Autocalibration
Metric Bundle Adjustment
Dense Reconstruction
In progress . . .

**Fig. 9.** Overall pipeline of Video registration.

- **Feature detection and tracking**
  Points are detected using the slightly modified Harris corner detector [31], lines are detected using a Canny-like algorithm [32]. In the case of video sequences we can use a helpful assumption that 2d projections of particular features do not differ too much for subsequent frames. This allows us to restrict the search area for a matching correspondence to the neighbourhood of the current feature position translated to the next frame. We use a pure image-based similarity measure (intensity cross-correlation) for points and a combination of image-based and geometric properties (length, orientation) for lines.

It is necessary to use some technique to detect mismatches. Geometric constraints for point projections in two images (epipolar constraint) and for point and line projections in three images (trilinear constraints) provide means not only to detect mismatches via robust RANSAC procedure but also allow to use guided matching by predicting feature positions from the estimated geometry (a fundamental matrix in case of two images and a trifocal tensor in case of three images). More information on our implementation can be found in [15].

- **Projective reconstruction from triplets**
  If a triple of images with established feature correspondences (projections of 3d features) is available, we can estimate a so-called trifocal tensor - a set of numbers that describe relative projective geometry of three uncalibrated cameras. A linear algorithm for the estimation of tensor components is followed by a non-linear reprojection error minimizing iterative Levenberg-Marquardt [33] algorithm. In fact this algorithm is a variant of a bundle adjustment for three cameras.
- **Triplet merging**
  In order to obtain one basis projective reconstruction for the whole sequence we use a triplet merging scheme. Every triplet has a projective reconstruction (based on an estimated trifocal tensor) in its own basis. Our goal is to retrieve parameters of all cameras in a common projective basis. This is done by "glueing" adjacent triplets to the existing projective reconstruction. Every two adjacent triples have two common cameras, hence we can build a projective transformation that transforms the second basis to the first one. First this transformation is estimated algebraically, then an optimal solution is found using the Levenberg-Marquardt iterative algorithm that minimizes reprojection error for the features the the projections of which are available in these four frames.
- **Projective Bundle Adjustment**
  The projective reconstruction that was built at the previous stage serves as an initial point for a globally-optimizing projective *bundle adjustment* algorithm which minimizes the sum of reprojection errors for all available projections in all images. This problem tends to be high-dimensional since typically we have many cameras and many features. That is why a sparse block-based variant of the Levenberg-Marquardt algorithm is used here [16, 27]. Robust cost function is also of great importance and can make a significant change in the behavior of the minimization algorithm especially if some outliers are still present in the data.
- **Autocalibration**
  Autocalibration means a procedure that retrieves internal parameters of the cameras and transforms the reconstruction of the cameras and the structure from projective to metric basis if certain restrictions on the cameras are assumed to be valid. For example, a zero skew, a centered principal point, square pixels, the constancy of internal parameters for all cameras.
  This part of the pipeline is still in progress.
- **Metric Bundle Adjustment**

The success at the previous stage means that we have an initial metric reconstruction available. As it was in the case of projective reconstruction a bundle adjustment procedure is necessary to improve the quality of the reconstruction. The algorithm is much the same as a projective bundle adjustment but additional constraints on the cameras (that assure that the cameras remain metric during the minimization) must be imposed.
This part of the pipeline is still in progress.

- **Dense reconstruction**
  Dense reconstruction deals with methods to reconstruct the shape and texture of the observed objects based on images and already estimated metric cameras and features.
  This will be the subject of our consideration in the nearest future.

### 4.3 Conclusion

The first four stages of the pipeline have already been implemented. This allows us to obtain the projective reconstruction of cameras and point and line features from video sequences. The details on the algorithms are published in [15, 34]. The implementations of these techniques can be used not only together but it is also possible to integrate them partially in another system. The library is based on object-oriented principles and separate stages have standard interfaces therefore different parts can be replaced and combined independently.

## 5 Aknowledgements

The presented projects were carried on under research agreements with Intel Russian Research Center (IRRC), Russian Fund for Basic Research (RFBR), and other organizations.

## References

1. V.G.Zhislina, D.V.Ivanov, V.F.Kuriakin, V.S.Lempitsky, E.M.Martinova, K.V.Rodyushkin, T.V.Firsova, A.A.Khropov, and A.V.Shokurov. Creating and Animating Personalized Head Models from Digital Photographs and Video. Programming and Computer Software, Vol. 30, No. 5, 2004, pp. 242v257. (in English) / Translated from Programmirovanie, Vol. 30, No. 5, 2004. (in Russian)
2. Lee, Y., Terzopoulos, D., and Waters, K., Realistic Modeling for Facial Animation, Proc. of SIGGRAPH95, 1995, pp. 55–62.
3. Goto, T., Kshirsagar, S., and Magnenat-Thalmann, N., Automatic Face Cloning and Animation, IEEE Signal Processing Magazine, 2001, vol. 18, no. 3, pp. 1725.
4. Lavagetto, F., Pockaj, R., and Costa, M., Smooth Surface Interpolation and Texture Adaptation for MPEG-4 Compliant Calibration of 3D Head Models, Image Vision Computing J., 2000, vol. 18, no. 4, pp. 345354.
5. Lee, W., Kalra, P., and Magnenat-Thalmann, N., Model Based Face Reconstruction for Animation, Proc. of MMM97, World Sci., 1997, pp. 323338.

6. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., and Salesin, D., Synthesizing Realistic Facial Expressions from Photographs, Proc. of SIGGRAPH98, 1998, pp. 7584.
7. Blanz, V. and Vetter, T., A Morphable Model for the Synthesis of 3D Faces, Proc. of SIGGRAPH99, 1999, pp. 187194.
8. Lee, W. and Magnenat-Thalmann, N., Fast Head Modeling for Animation, Image Vision Computing J., 2000, vol. 18, no. 4, pp. 355364.
9. Cohen, M., Jacobs, C., Liu, Z., and Zhang, Z., Rapid Modeling of Animated Faces from Video, Microsoft Tech. Report MSR-TR-2000-11, 2000.
10. Fua, P., Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data, Int. J. Comput. Vision, 2000, vol. 38, no. 2, pp. 153171.
11. Lee, W., Escher, M., Sannier, G., and Magnenat-Thalmann, N., MPEG-4 Compatible Faces from Orthogonal Photos, Proc. of CA99, 1999, pp. 186194.
12. Bovyrin, A.B. and Rodyushkin, K.V., Statistical Estimation of Color Components of Lips and Face to Determine the Mouth Contour by the Deformable Template Method, Abstracts of the 11th Conf. Mathematical Methods of Pattern Recognition, Moscow, 2003, pp. 245247.
13. Rodyushkin, K.V., Eye Features Tracking by Deformable Template to Estimate Face Animation Parameters, Proc. of Advanced Concepts for Intelligent Vision Systems, Ghent, Belgium, 2003, pp. 267274.
14. Yuille, A.L., Hallinan, P.W., and Cohen, D.S., Feature Extraction from Faces Using Deformable Templates, Int. J. Comput. Vision, 1992, vol. 8, no. 2, pp. 99111.
15. Shokurov, A., Khropov, A., and Ivanov, D., Feature Tracking in Image and Video, Proc. of GraphiCon-2003, 2003, pp. 177179.
16. R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2000.
17. Laurentini, A., The Visual Hull Concept for Silhouettesbased Image Understanding, IEEE Trans. Pattern Anal. Machine Intelligence, 1994, vol. 16, no. 2, pp. 150162.
18. Ivanov, D., Lempitsky, V., Shokurov, A., Khropov, A., and Kuzmin, Ye., Creating Personalized Head Models from Image Series, Proc. of the 13th Int. Conf. on Comput. Graphics GraphiCon2003, Moscow, 2003.
19. P. Burt, and E. Andelson. A Multiresolution Spline With Application to Image Mosaics. ACM Transactions on Graphics, 2(4), pp. 217-236, 1983.
20. Lempitsky, V., Ivanov, D., and Kuzmin, Ye., Texturing Calibrated Head Model from Images, Proc. of Euro- Graphics2002, 2002, pp. 281288.
21. Fedorov, A., Firsova, T., Kuriakin, V., Martinova, E., Mindlina, O., Rodyushkin, K., and Zhislina, V., Talking Head: Synthetic Video Facial Animation in MPEG-4, 13th Int. Conf. on Comput. Graphics and Vision GraphiCon 2003, Moscow, 2003, pp. 3741.
22. P. Debevec, C. Taylor, and J. Malik, Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach, SIGGRAPH96 Conference Proceedings, Annual Conference Series, 1120, July, 1996.
23. Sebastien Dedieu, Pascal Guitton, Christophe Schlick, and Patrick Reuter. Reality: an interactive reconstruction tool of 3d objects from photographs. In Proceedings of Vision Modeling and Visualization'2001, pages 195-202, 2001. Held in Stuttgart, Germany.
24. Canoma web site, http://www.canoma.com.
25. PhotoModeler(EOS Systems) web site, http://www.photomodeler.com.
26. RealViz web site, http://www.realviz.com.

27. Triggs, B. and McLauchlan, P. and Hartley, R. and Fitzgibbon, A.: Bundle Adjustment - A Modern Synthesis, Vision Algorithms: Theory and Practice, Springer-Verlag, LNCS 1883, pp. 298–372, 2000.
28. OpenGL web site, http://www.opengl.org/
29. Rikk Carey and Gavin Bell. The Annotated VRML 2.0 Reference Manual. Addison-Wesley, 1997.
30. Boujou software by 2d3, http://www.2d3.com/jsp/products/product-overview.jsp?product=10
31. C.J. Harris and M. Stephens. A combined corner and edge detector. In Proc. 4th Alvey Vision Conference, Manchester, pp.147-151, 1988.
32. J. Canny, A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679–698, 1986.
33. W.Press, B.Flannery, S.Teukolsky and W. Vetterling. Numerical recipes in C, 2nd Ed. Cambridge, UK: Cambridge University Press, 1992.
34. A. Khropov, A. Shokurov, V. Lempitsky, and D. Ivanov. Reconstruction of projective and metric cameras for image triplets. GraphiCon-2004 Proc., pp. 143-146.

# Video user interfaces

Peter Robinson

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
pr@cl.cam.ac.uk

**Abstract.** The increasing power and falling cost of computers, combined with improvements in digital projectors and cameras, are making the use of video interaction in human-computer interfaces more popular. This paper presents a review of video interface projects in the Computer Laboratory at the University of Cambridge over the past 15 years. These encompass early work on augmented environments, applications in publishing, personal projected displays, and emotionally aware interfaces.

## 1 Introduction

The increasing power and falling cost of computers, combined with improvements in digital projectors and cameras, are making the use of video interaction in human-computer interfaces more popular. This paper reviews work on video interfaces at the University of Cambridge over the past 15 years, and presents two recent projects in more detail.

With support from the Rank Xerox Research Centre in Cambridge, we laid the foundations for a new model of interaction based on video interfaces in the early 1990s. We built a user interface based on video projection and digital cameras (the *DigitalDesk*), extended this for remote collaboration (the *DoubleDigitalDesk*), and investigated the use of a camera for input alone (*BrightBoard*). The result is an augmented environment in which everyday objects acquire computational properties, rather than virtual environments where the user is obliged to inhabit a synthetic world.

The research continued with support from the EPSRC in the later 1990s to investigate combinations of electronic and conventional publishing, with applications in education. The *Origami* project combined electronic and printed documents to give a richer presentation than that afforded by either separate medium.

People manage large amounts of information on a physical desk, using the space to arrange different documents to facilitate their work. The 'desk top' on a computer screen only offers a poor approximation. Thales Research & Technology have supported work on the *Escritoire*, a desk-based interface for a personal workstation that uses two overlapping projectors to create a foveal display: a large display surface with a central, high resolution region to allow detailed work. Multiple pen input devices are calibrated to the display to allow input with both hands. A server holds the documents and programs while multiple clients connect to collaborate on them.

Facial displays are an important channel for the expression of emotions, and are often thought of as projections of a person's mental state. Computer systems generally ignore this information. *Mind-reading* interfaces infer users' mental states from facial expressions, giving the computer a degree of emotional intelligence. Video processing is used to track two dozen features on the user's face. These are then interpreted as basic action units, which are interpreted using statistical techniques as complex mental states.

## 2 Video augmented environments

The availability of digital video projection and digital video capture in the early 1990s led us to conceive the *DigitalDesk* – an ordinary desk augmented with a computer display using projection television and a video camera to monitor inputs [22][23]. Figure 1 shows the desk with a projector (made from an overhead projector and an early liquid crystal display) and two cameras.

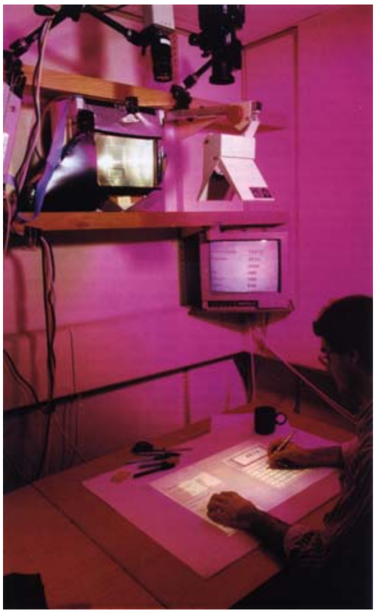


**Fig. 1.** The DigitalDesk

A number of prototype systems were implemented to demonstrate its feasibility. Figure 2 shows a sketching application called *PaperPaint*. The darker lines have been drawn with a pen. Some of these have then been copied electronically, and appear as grey lines in the projected image. Figure 3 shows the *DoubleDigitalDesk* where two

DigitalDesks are being used to support collaborative work [8]. The inset image at the top right shows the other participant.

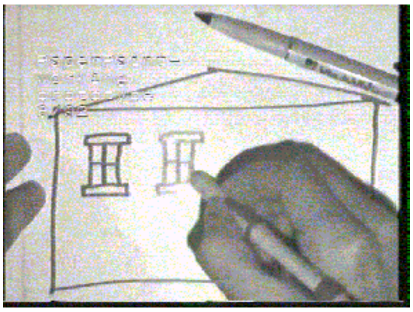


**Fig. 2.** PaperPaint on the DigitalDesk

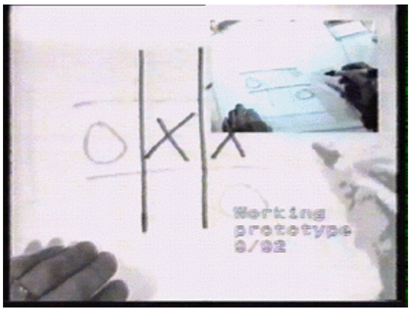


**Fig. 3.** The DoubleDigitalDesk

*BrightBoard* dispensed with the projector, and just used a camera to enable any part of the user's environment to be used to control a computer [20][21]. Figure 4 shows an ordinary whiteboard being monitored by a camera. The user could write commands on the board, for example to print a hard copy of its contents.

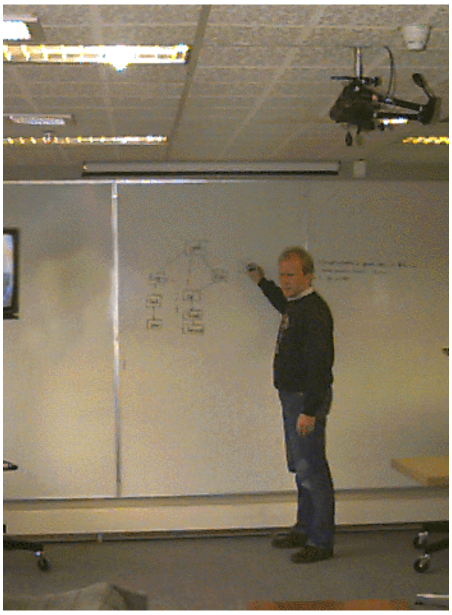


**Fig. 4.** BrightBoard

These early experiments established the value of *augmented environments* in which everyday objects such as paper and whiteboards acquired computational properties. This contrasts with virtual environments, where the user is obliged to inhabit a synthetic world.

## 3 Animated paper documents

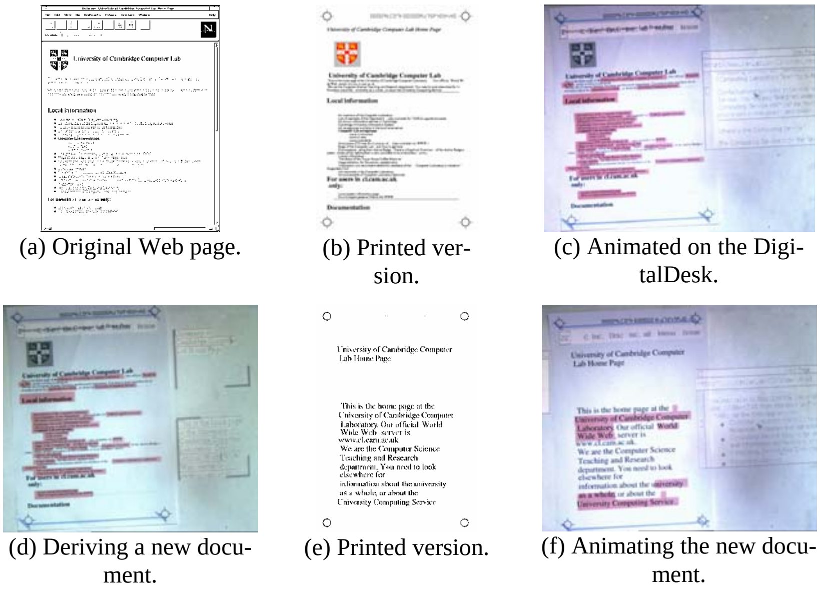
Electronic, multi-media publishing is becoming established as an alternative to conventional publishing on paper. CD-ROM and on-line versions of reference books and fiction can augment their conventional counterparts in a number of ways:

- They offer elaborate indexing, glossaries and cross-referencing.
- They allow non-linear progression through the text.
- Sound and moving images can be added.
- Sections can be copied into new documents.

However, screen-based documents have a number of disadvantages:

- People find screens harder to read than paper.
- Electronic bookmarks are less convenient than bits of paper or flicking through a book.
- Adding personal notes to electronic documents is more complicated than jotting in the margin of a book.
- Writing, editing and proof-reading a non-linear, multi-media document is still a specialised and difficult task.

Our solution is to publish material as an ordinary, printed document that can be read in the normal way, enjoying the usual benefits of readability, accessibility and portability. However, when observed by a camera connected to a computer, the material acquires the properties of an electronic document, blurring the distinction between the two modes of operation [16][17][19].



(a) Original Web page.

(b) Printed version.

(c) Animated on the DigitalDesk.

(d) Deriving a new document.

(e) Printed version.

(f) Animating the new document.

**Fig. 5.** Paper access to the World-Wide Web

A simple demonstration of this principle is a system enabling interaction with printed versions of Web pages [18]. Figure 5 shows a conventional WWW page at (a). This is imported into the system and reprinted with additional coding to assist recognition (b). When this is placed on a DigitalDesk it is recognised and active areas of the document illuminated by projected highlights. When these are selected, links are followed or programs executed and the results projected into a further window on the work surface (c).

Moreover, fragments can be copied from the paper document into new electronic documents also projected onto the desk (d). The new document can be printed to give a new paper document (e) which can be animated on the desk in just the same way (f).

Two further applications explored the use of this technology for educational material. The first is a course book for teaching mathematics [9]. The software which accompanies the course book is automatically launched when the book is first placed on the desk. Figure 6 shows a section on curve-sketching for polynomials. The generic equation of a quadratic polynomial is given with spaces for the values of the coefficients and an empty box underneath for plotting the graph. The software projects default values and draws the graph into the box. However, it also projects controls alongside the coefficients to allow the reader to change these values while observing the corresponding change in the graph.

Further down the page of the course book there is an assessment exercise. This time the polynomials are fixed and the student must draw the curve into the box (the active pen also has a real nib for writing). Clicking a projected button asks the computer to assess the sketch. The image is captured and analysed for features such as maxima, minima and axis crossings, and marked accordingly.

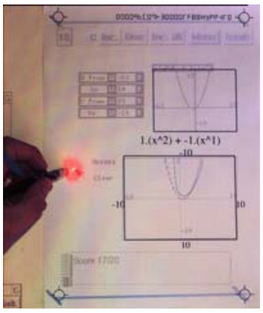


**Fig. 6.** A maths tutor

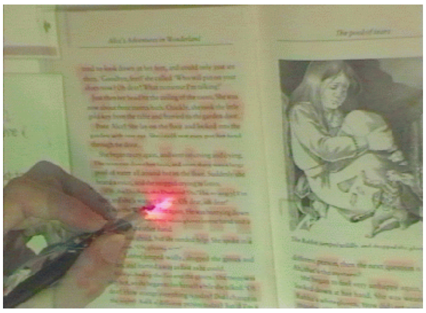


**Fig. 7.** A grammar tutor

Figure 7 shows a second educational application that teaches elementary grammar by animating a standard printed book [6][7]. This uses additional information from an SGML edition of the book distributed as part of the Text Encoding Initiative.

## 4 Personal projected displays

Since the inception of the personal computer, the interface presented to users has been defined by the monitor screen, keyboard, and mouse, and by the framework of the desktop metaphor. It is very different from a physical desktop which has a large horizontal surface, allows paper documents to be arranged, browsed, and annotated, and is controlled via continuous movements with both hands. The desktop metaphor will not scale to such a large display; the continuing profusion of paper, which is used as much as ever, attests to its unsurpassed affordances as a medium for manipulating documents; and despite its proven benefits, two-handed input is still not used in computer interfaces [14][15].

The *Escritoire* [1] uses a novel configuration of overlapping projectors to create a large desk display that fills the area of a conventional desk and also has a high resolution region in front of the user for precise work. The projectors need not be positioned exactly—the projected imagery is warped using standard 3D video hardware to compensate for rough projector positioning and oblique projection. Calibration involves computing planar homographies between the 2D co-ordinate spaces of the warped textures, projector framebuffers, desk, and input devices. The video hardware can easily perform the necessary warping and achieves 30 frames per second for the dual-projector display. Oblique projection has proved to be a solution to the problem of occlusion common to front-projection systems. The combination of an electromagnetic digitizer and an ultrasonic pen allows simultaneous input with two hands. The pen for the non-dominant hand is simpler and coarser than that for the dominant hand, reflecting the differing roles of the hands in bimanual manipulation. We use a new algorithm for calibrating a pen, that uses piecewise linear interpolation between control points. We can also calibrate a wall display at distance using a device whose position and orientation are tracked in three dimensions.

The Escritoire software is divided into a client that exploits the video hardware and handles the input devices, and a server that processes events and stores all of the system state. Multiple clients can connect to a single server to support collaboration. Sheets of virtual paper on the Escritoire can be put in piles which can be browsed and reordered. As with physical paper this allows items to be arranged quickly and informally, avoiding the premature work required to add an item to a hierarchical file system. Another interface feature is pen traces, which allow remote users to gesture to each other. We report the results of tests with individuals and with pairs collaborating remotely. Collaborating participants found an audio channel and the shared desk surface much more useful than a video channel showing their faces.

The Escritoire is constructed from commodity components, and unlike multi-projector display walls its cost is feasible for an individual user and it fits into a normal office setting. It demonstrates a hardware configuration, calibration algorithm, graphics warping process, set of interface features, and distributed architecture that can make personal projected displays a reality.

### 4.1 Foveal display

To create a display that fills an entire desk but also allows life-sized documents to be displayed and manipulated we have created what we call a *foveal* display. One projector fills the desk with a low-resolution display, while a second overlapping projector displays a high-resolution area in front of the user. The optical path of the first projector is folded using a mirror above the desk to enable it to generate a display of the desired size without being mounted at an inconveniently high position above the desk surface. Figure 8 shows the general arrangement. Baudisch et al. have combined an LCD monitor and a projector to get a dual-resolution display [2], although they do not address calibration, have used only a conventional keyboard and mouse for input, and get a display with different affordances because of its vertical rather than horizontal placement.

The user can arrange items on the desk, identify them at a glance, reach out and grab them, and quickly move them to the high-resolution region where the text becomes legible and they can be worked on in detail. Figure 9 shows a document being moved from the periphery into the fovea.

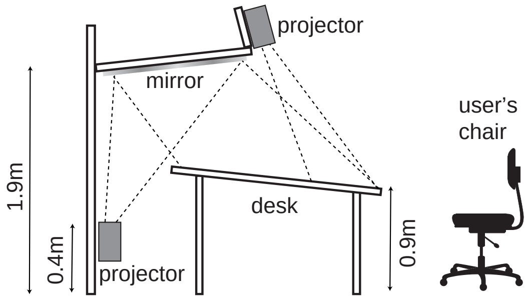


**Fig. 8.** The two-projector configuration of the Escritoire

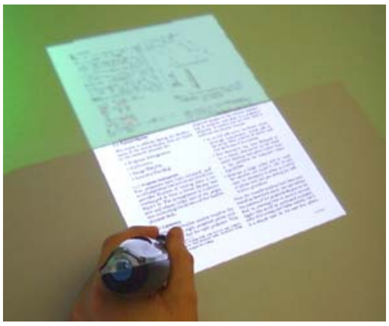


**Fig. 9.** Moving a sheet of virtual paper to the high-resolution region

### 4.2 Two-handed interaction

Bimanual input—using two hands—has manual benefits from increased time-motion efficiency due to twice as many degrees of freedom being simultaneously available to the user, and also cognitive benefits which arise as a result of reducing the load of mentally composing and visualizing a task at an unnaturally low level imposed by traditional single-handed techniques.

We have combined a desk-sized digitizer and stylus that provide accurate input for the user's dominant hand, with an ultrasonic whiteboard pen that provides simple and less accurate tracking for the user's non-dominant hand. The non-dominant hand is used to move items around on the desk, setting up a frame of reference for the dominant hand to do its more detailed work such as writing and drawing.

### 4.3 Collaboration

We have implemented the Escritoire in two parts: a server written in Java that stores the details of the items on the desk , and a client written in C++ that handles the input and output devices. This allows multiple desks to connect to the same server over the Internet allowing geographically separated users to share the desk contents.

We have conducted tests in which pairs of participants converse over a standard videoconference while using Escritoire desks whose contents are shared in a What You See Is What I See fashion. Figure 10 shows a videoconference being conducted on an ordinary computer, but where both participants are also using a pair of Escritoires driven from the same server. As they talk they can work together to read and annotate documents, gesturing in the shared graphical space as they do so. Systems for remote collaboration often concentrate on optimizing the talking heads model of a standard videoconference but we have found that a shared task space is often more useful. The shared space provided by the Escritoire is much larger than a monitor screen and supports fast and natural interaction over the whole area, so users share a large visual context while being able to easily refer to and collaborate on specific items.

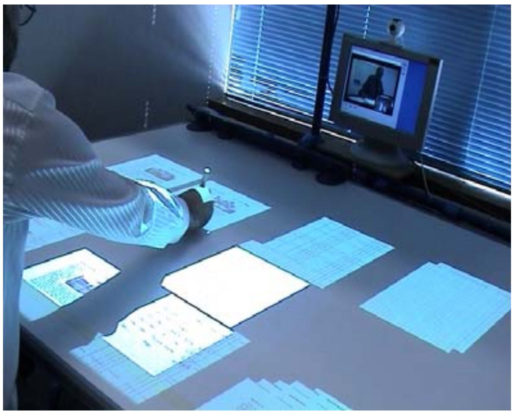


**Fig. 10.** Augmenting a videoconference with a desk surface that is shared between collaborators

## 5 Mind-reading interfaces

People routinely express their emotions and mental states through their facial expressions. Other people are used to this, and read their minds accordingly. This non-verbal communication is a vital part of human society, and those who lack the ability to read facial expressions are at a disadvantage. All computers suffer this disadvantage by failing to read their users' minds. In effect, computers are autistic. We have developed an automated system to remedy this problem [11][12][13].

In order to support intelligent man-machine interaction the system is designed to meet three important criteria. These are full automation so that it requires no human intervention, the ability to execute in real-time, and the categorization of mental states early enough after their onset to ensure that the resulting knowledge is current and actionable. Other aspects include being user-independent and dealing with substantial rigid head motion. The experimental evaluation shows promising results for 24 classes of complex mental states (sampled from 6 groups) in different interaction scenarios.

### 5.1 Multi-level representation

A person’s mental state is not directly available to an observer (the machine in this case) and as a result has to be inferred from observable behaviour such as facial signals. The process of reading a person's mental state in the face is inherently uncertain. Different people with the same mental state may exhibit very different facial expressions, with varying intensities and durations. In addition, the recognition of head and facial displays is a noisy process.

To account for this uncertainty, we use a multi-level representation of the video input, combined in a Bayesian inference framework. Our system abstracts raw video input into three levels, each conveying face-based events at different granularities of spatial and temporal abstraction. Each level captures a different degree of temporal detail depicted by the physical property of the events at that level. As shown in Figure 11, the observation (input) at any one level is a temporal sequence of the output of lower layers. At the bottom level, 24 facial feature points are tracked in each new frame every 33ms. Figure 12 shows hierarchy of the spatial analysis consisting of:

- *actions* which are explicitly coded being detected every 166ms,
- *displays* recognised by Hidden Markov Models (HMMs) every second,
- *mental states* assigned probabilities by Dynamic Bayesian Networks (DBNs) every two seconds.

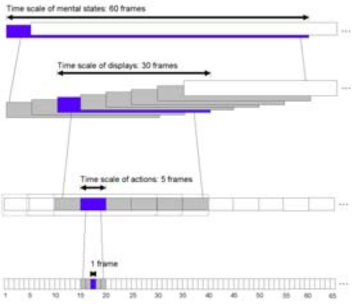

Time scale of mental states: 60 frames
Time scale of displays: 30 frames
Time scale of actions: 5 frames
1 frame
1
5
10
15
20
25
30
35
40
45
50
55
60
65

**Fig. 11.** Temporal abstraction in the mind-reading machine

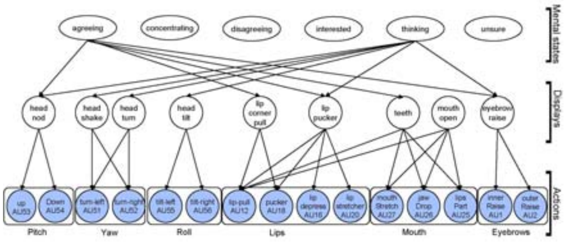


**Fig. 12.** Classification hierarchy

This approach has a number of advantages. First, higher-level classifiers are less sensitive to variations in the environment because their observations are the outputs of the middle classifiers. Second, with each of the layers being trained independently, the system is easier to interpret and improve at different levels. Third, the Bayesian framework provides a principled approach to combine multiple sources of information. Finally, by combining dynamic modelling with multi-level temporal abstraction, the model fully accounts for the dynamics inherent in facial behaviour. In terms of implementation, the system is user-independent, unobtrusive, and accounts for rigid head motion while recognizing meaningful head gestures.

### 5.2 Training

A great deal of data was necessary to determine the window sizes in the temporal abstraction and to train the statistical classifiers in the inference system. We have used the Mind Reading DVD [5], a computer-based guide to emotions, developed by a team of psychologists led by Professor Simon Baron-Cohen at the Autism Research Centre in the University of Cambridge. The DVD was designed to help individuals diagnosed along the autism spectrum recognize facial expressions of emotions.

The DVD is based on a taxonomy of emotion by Baron-Cohen *et al.* [4] that covers a wide range of affective and cognitive mental states. The taxonomy lists 412 mental state concepts, each assigned to one (and only one) of 24 mental state classes. The 24 classes were chosen such that the semantic distinctiveness of the emotion concepts within one class is preserved. The number of concepts within a mental state class that one is able to identify reflect one's empathizing ability [3].

Out of the 24 classes, we focus on the automated recognition of 6 classes that are particularly relevant in a human-computer interaction context, and that are not in the basic emotion set. The 6 classes are: *agreeing*, *concentrating*, *disagreeing*, *interested*, *thinking* and *unsure*. The classes include affective states such as *interested*, and cognitive ones such as *thinking*, and encompass 29 mental state concepts, or fine shades, of the 6 mental states. For instance, *brooding*, *calculating*, and *fantasizing* are different shades of the *thinking* class; likewise, *baffled*, *confused* and *puzzled* are concepts within the *unsure* class.

Each of the 29 mental states is captured through six video clips. The resulting 174 videos were recorded at 30 frames per second, and last between 5 to 8 seconds at a resolution of 320×240. The videos were acted by 30 actors of varying age ranges and ethnic origins. All the videos were frontal with a uniform white background. The process of labelling the videos involved a panel of 10 judges who were asked "could this be *the emotion name*?" When 8 out of 10 judges agreed, a statistically significant majority, the video was included. To the best of our knowledge, the Mind Reading DVD is the only available, labelled resource with such a rich collection of mental states, even if they are posed.

### 5.3 Operation

Figure 13 shows the system in operation. Seven frames from a six second performance of the *undecided* emotion are shown. These are followed by the outputs from the HMMs during the video for five displays – *head nod*, *head shake*, *head tilt*, *head turn*, and *lip pull*. Finally, the outputs from the DBNs are shown giving the probabilities of the six mental state classes during the clip.

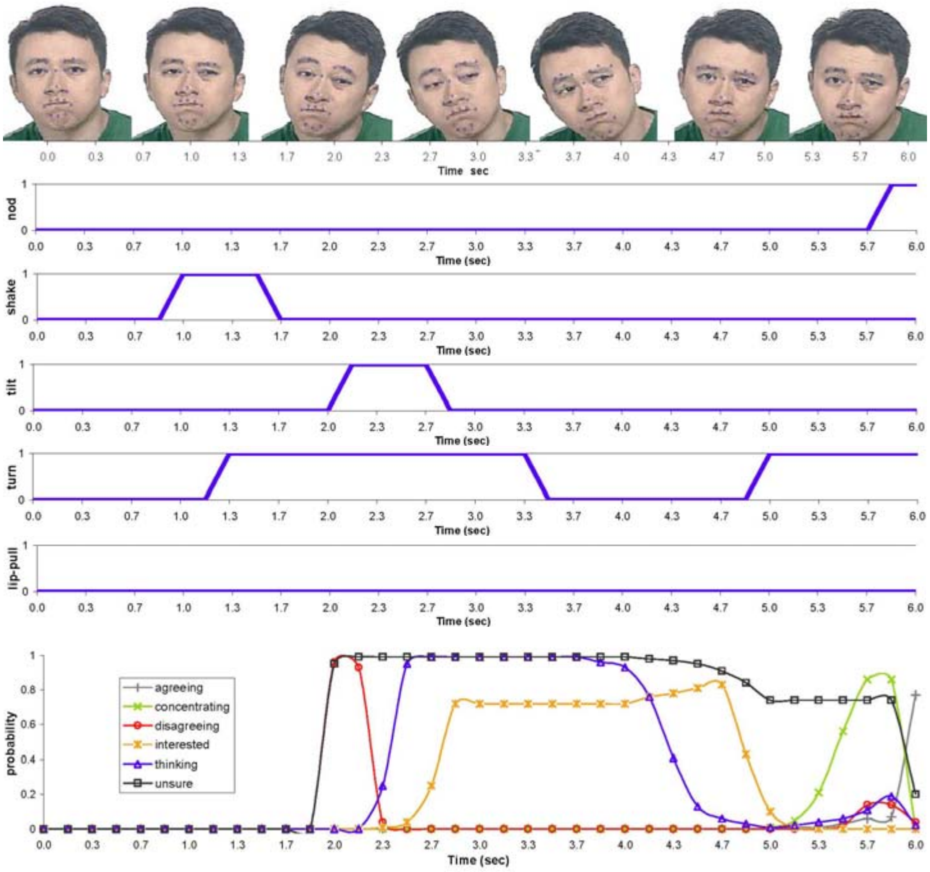


**Fig. 13.** Selected frames, traces of display recognition, and probabilities of mental state inference in a video labeled as *undecided*

The probabilities of the different mental state classes vary during the course of the video, and there are several plausible interpretations. This reflects the position with recognition of emotions by humans. A principal state can be inferred by measuring the area under the six graphs, and selecting the largest. In this case, *unsure* is correctly selected as the class within which *undecided* falls.

The overall accuracy of the system was evaluated by testing the inference results of 164 videos representing the six mental state classes. The videos span 25645 frames, or approximately 855 seconds. Using a leave-one-out methodology, 164 runs were carried out, where for each run the system was trained on all but one video, and then tested with that video. The classification rule that is used to deem whether a classification result is correct is defined as follows: compare the overall probability of each of the mental states over the course of a video. If the video’s label matches that of the most likely mental state or the overall probability of the mental state exceeds 0.6, then it is a correct classification.

The results are summarized as a 3D bar chart in Figure 14. The horizontal axis represents the classification results of each mental state class. The percentage of recognition of a certain mental state is represented along the z-axis.

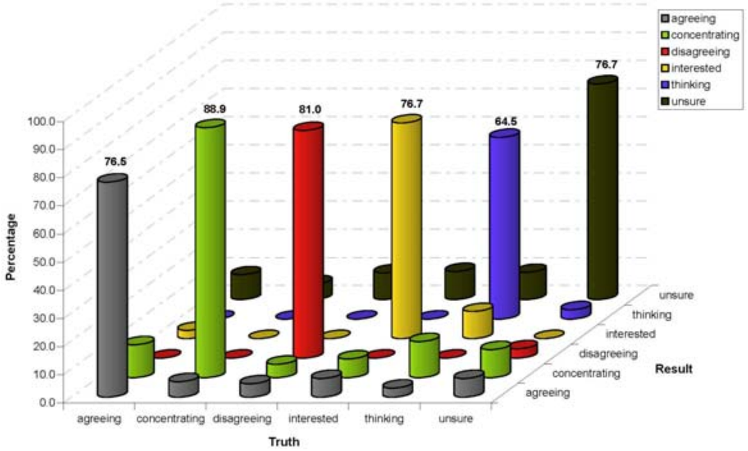


**Fig. 14.** Confusion matrix for the six classes of mental state used in the trials

For a mean false positive rate of 4.7%, the overall accuracy of the system is 77.4%. These results compare favourably with an earlier experiment in which the performance of a group of people in recognizing complex mental states in a similar set of videos from the Mind-reading DVD was tested []. In that experiment, human recognition rate reached an upper bound of 71.0 %. Thus, the accuracy of the automated mind-reading system in classifying complex mental states from videos of the Mind-reading DVD compares favourably to that of humans. Moreover, the system operates in real time on a standard computer workstation.

We are currently evaluating the performance of the automated mind-reading system in a more general context. The idea is to train the system on videos from the Mind-reading DVD, and test its performance on a previously unseen corpus with different recording conditions and subjects than those used in training the system. The generalization performance of a system is an important indicator to how well the system does outside of laboratory settings.

## 6 Conclusions

This paper has reviewed work on video user interfaces over 15 years at the University of Cambridge Computer Laboratory. The initial view that using cameras and projectors as part of the human-computer interface has proved extremely fruitful. Indeed, the steady improvements in technology over this period mean that computers are now 1000 times faster and have 1000 times the memory. Cameras have fallen in price by a similar factor. Projectors have also improved in brightness and resolution, and fallen in price, albeit by a rather smaller factor.

Many of the technical challenges remain the same. Projection systems require non-linear transformations to accommodate oblique projection and to correlate the coordinate systems of the different input and output devices. Analysing video input is expensive in terms of both processing and memory. However, the hardware of modern graphics cards can be exploited to offload much of this processing, and the systems now run comfortably on commodity hardware.

The experimental systems and applications investigated over the past 15 years in Cambridge are now entering the main stream. The Escritoire is being used to support distributed command and control systems for crisis management. Mind-reading interfaces are being used to augment teleconferencing systems and to control figures in computer animations. Video input and output are focal in the movement towards improved availability and usability of computer systems.

## References

1. Ashdown, M.: *Personal Projected Displays.* PhD Dissertation, University of Cambridge Computer Laboratory Technical report 585, September 2003.
2. Baudisch P., Good N., Stewart P.: Focus Plus Context Screens: Combining Display Technology with Visualization Techniques. *Proceedings of UIST 2001*, pages 31–40.
3. Baron-Cohen,S.: *Mindblindness: An Essay on Autism and Theory of Mind*. MIT Press, 1995.
4. Baron-Cohen, S., Golan, O., Wheelwright, S., Hill, J.: *A New Taxonomy of Human Emotions*. 2004.
5. Baron-Cohen, S., Golan, O., Wheelwright, S., Hill, J.: *Mind Reading: The Interactive Guide to Emotions*. London: Jessica Kingsley Publishers, 2004.
6. Brown H., Harding R.D., Lay S.W., Robinson P., Sheppard D.P., Watts R.R.: *Active paper for active learning.* Proceedings 4th annual conference Association for Learning Technology, Telford, September 1997, reprinted in Association for Learning Technology Journal, 1998.
7. Brown H., Harding R.D., Lay S.W., Robinson P., Sheppard D.P., Watts R.R.: *Active Alice - using real paper to interact with electronic text*. Proceedings 7th International Conference on Electronic Publishing, Saint Malo, April 1998, ISBN 3 540 64298 6, pp 407-419.
8. Freeman, S.M.G. *An architecture of distributed user interfaces*. PhD Dissertation, University of Cambridge Computer Laboratory Technical Report 342, July 1994.
9. Harding R.D., Lay S.W., Robinson P., Sheppard D.P., Watts R.R.: *New technology for interactive CAL - the Origami project*. Proceedings 3rd annual conference Association for Learning Technology, September 1996, reprinted in Association for Learning Technology Journal 5(1), 1997, ISSN 0968 7769, pp 6-12.
10. el Kaliouby R., Robinson P., Keates S.: Temporal Context and the Recognition of Emotion from Facial Expression. Proceedings of the *HCI International Conference*, June 2003.
11. el Kaliouby R., Robinson P.: Mind-reading Machines: Automated Inference of Cognitive Mental States from Video. Proceedings of IEEE International Conference on Systems, Machines and Cybernetics, 2004.
12. el Kaliouby R., Robinson P.: Real-time inference of complex mental states from facial expressions and head gestures. Workshop on Real-Time Vision for Human-Computer Interaction at the IEEE CVPR Conference, 2004.
13. el Kaliouby R., Robinson P.: The emotional hearing aid: an assistive tool for children with Asperger’s Syndrome. International Workshop on Universal Access and Assistive Technology, pages 244–246, 2004.

14.Leganchuk A., Zhai S,, Buxton W.: Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study. *Trans. on HCI 5(4)*, pages 326–359, 1998.

15.Norman D.: *The Psychology of Everyday Things*. Basic Books, 1988.

16.Robinson P., Sheppard D.P., Watts R.R., Harding R.D., Lay S.W.: *Animated paper documents.* Proceedings HCI '97, San Fransisco, August 1997, reprinted in Design of computing systems: social and ergonomic considerations 21B, Elsevier 1997, ISBN 0 444 82183 X, pp 655-658.

17.Robinson P., Sheppard D.P., Watts R.R., Harding R.D., Lay S.W.: *A framework for interacting with paper.* Proceedings Eurographics '97, Computer Graphics Forum 16(3), September 1997, ISSN 0167 7055, pp 339-324.

18.Robinson P., Sheppard D.P., Watts R.R., Harding R.D., Lay S.W.: *Paper interfaces to the World-wide.* Procedings WebNet ’97. Toronto, November 1997, ISBN 1 880094 27 4, pp 426-431.

19.Robinson P.: *Digital manuscripts and electronic publishing.* International Congress on Production and Context, Constantijn Huygens Institute, The Hague, March 1998; reprinted in *Editio* 13, Autumn 1999, pp 337-346.

20.Stafford-Fraser, J.Q.: *Video augmented environments*. PhD Dissertation, University of Cambridge Computer Laboratory Technical Report 419, February 1996.

21.Stafford-Fraser J.Q., Robinson P.: *BrightBoard - a video augmented environment.* Proceedings ACM Conference on Computer-Human Interaction, April 1996, pp 134-141.

22.Wellner P.D.: *Interacting with paper on the DigitalDesk.* Communications of the ACM 36(7), July 1993, pp 87-96.

23.Wellner P.D.: *Interacting with paper on the DigitalDesk*. PhD Dissertation, University of Cambridge Computer Laboratory Technical Report 330, October 1993.

## Acknowledgements

The systems described in this paper are the results of work by many people. Pierre Wellner, Steve Freeman and Quentin Stafford-Fraser undertook early experiments on the DigitalDesk, DoubleDigitalDesk and BrightBoard while working as research students in the Computer Laboratory with sponsorship from the Rank Xerox Research Centre in Cambridge. Steve Lay, Dan Sheppard and Richard Watts investigated animated paper documents under a grant from the UK Engineering and Physical Sciences Research Council. Robert Harding and Heather Brown also contributed to the work. Mark Ashdown designed and built the Escritoire while working as a research student in the Computer Laboratory with sponsorship from Thales Research and Technology. Rana el Kaliouby designed and built the mindreading system as a research student. Simon Baron-Cohen and his colleagues in the Autism Research Centre gave valuable advice on emotional intelligence, and supplied test data on DVD.

# How Content Indexing may affect User Interfaces: Some Thoughts on the Search Engine Revolution

Quentin Stafford-Fraser

Exbiblio
c/o Newnham Consulting, 20 Marlowe Road,
Cambridge CB3 9JW, United Kingdom
quentin@pobox.com

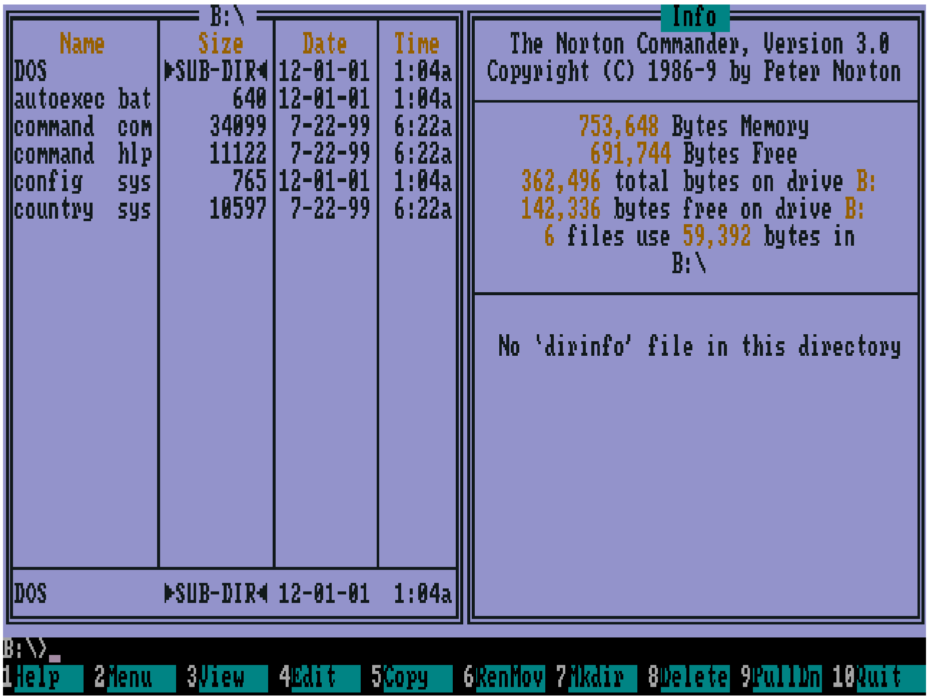
**Abstract.** The way we find information is changing. In the past, information was located primarily by attributes which were external to the information itself; titles, authors, filenames, dates. In the future, it will be located to a much greater degree using previously-created full-text indexes of the contents of documents. We examine the factors that have led to this change, and the likely implications for the future, especially in some less-typical user-interface scenarios. We look in particular at how it might affect our interaction with paper documents.

## Introduction – Finding information

If the last five years of the last millennium were the years of the web, it seems likely that the first five years of this one may become known as the Age of the Search Engine.

The way we locate information is changing. In the past, information was located primarily by attributes which were external to the information itself. In the case of a book, it was often the author’s surname or the ISBN number. In the case of a computer, it was the directory and filename. In an academic's filing cabinet, perhaps the name of a publication or conference from which a paper came, or the title of a project to which it refers.

Filing information, however, is a very personal process. The categories that you use in your filing cabinet, or the directory structure of your hard disk, represent the way you view the world, and in fact, they usually represent the way you *viewed* the world at the time the filing system was set up. In some circumstances, they may represent the way your organisation views the world, in which case other people in your group may be able to navigate your filing system, but anybody from outside that group might still have difficulty. In the past, though, files were typically stored on *personal* computers, the amount of content was relatively small, and it was often only the creator of the content who ever had to find it again.



```
                        B:\                                     Info
   Name          Size      Date     Time    The Norton Commander, Version 3.0
DOS            ►SUB-DIR◄ 12-01-01  1:04a   Copyright (C) 1986-9 by Peter Norton
autoexec bat        640  12-01-01  1:04a
command  com      34099   7-22-99  6:22a           753,648 Bytes Memory
command  hlp      11122   7-22-99  6:22a            691,744 Bytes Free
config   sys        765  12-01-01  1:04a       362,496 total bytes on drive B:
country  sys      10597   7-22-99  6:22a        142,336 bytes free on drive B:
                                                  6 files use 59,392 bytes in
                                                             B:\

                                             No 'dirinfo' file in this directory




DOS            ►SUB-DIR◄ 12-01-01  1:04a
B:\>_
```

**Fig. 1.** Finding information in the past: The Norton Commander, a popular and much-copied interface for navigating a DOS filesystem

As the world becomes more networked, we spend more and more time looking *for* and looking *at* information created by others. As the amount of available information and the number of associated authors and editors increases, the less easy it becomes to navigate the filing systems of others.

In the early days of the web, it was practical to remember the URLs for the pages you found useful. As the amount of content grew, bookmarks appeared as a way to create your own filing system in which to put other people's information. Structured indexes like Yahoo soon followed and attempted to provide an index to the web, a taxonomy of web pages as a single place where you could start looking for information on a particular topic. Such directories also had the advantage of 'symbolic links' - web sites could be listed in more than one category at a time. But the directories provided an index at a fairly coarse granularity, and they still suffered from the problem of personal perspectives: the person creating the content was not, often, the person categorising it for the index, and the person searching for it might have had yet another view about how it should be listed. The job was made even more difficult by the need to create new categories for all the new things happening on the internet which could not easily be put in the pigeonholes of the past.

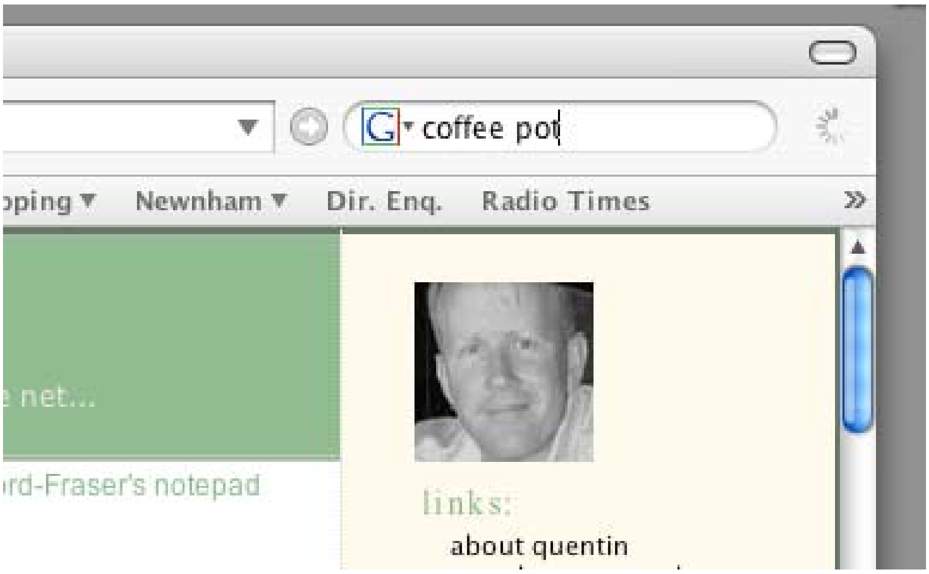


**Fig. 2.** Finding information today: the search engine built in to the Firefox web browser

## The Rise of the Search Engine

Search engines eventually provided the solution, locating information by the content, and not just by the location, author or title. It was what the information was actually about, not what the label said, that was important. Their success has been such that many browsers now offer two fields into which you can type things: one to locate a page by URL, the other for typing a search query. The one for the URL is often the least-used.

We can see some reflections of these trends in the commercial world, too. As more and more of other people's information becomes available to most of us, the proportion of computer time that most people spend creating information is smaller. For most users 15 years ago, the most important program was the word-processor - it was hard to imagine a piece of software absorbing more of your time than the one which helped you to write things. But then the web browser came along and changed the way we *read* things, which proved even more fundamental. So we have moved from a situation where the most influential technology businesses were those who allowed you to create content - Microsoft, Lotus - to those who allowed you to view it - Netscape, AOL, Real - and now to those which allow you to find it in the first place! (Google/Yahoo).

## Enabling the Revolution

We have discussed some of the driving forces for the search engine revolution. Several recent factors have also contributed to making it viable. The first was the availability of sufficient processing power and storage to ensure that the cost of creating and maintaining indexes was not high. Another was the availability of high-bandwidth links to servers which were always on and dedicated to indexing, meaning that it was not a task that had to be carried out on the user's PC. But perhaps the most important factor was that the web, on which everybody suddenly wanted to make their information available, used simple, open protocols and an open, text-based format for the vast majority of documents. (Before that, the best way to ensure that somebody else could read your electronic document was to print it out and send them a piece of paper.) The sudden transfer of so much of the world's knowledge into HTML meant that even when it had been created by one piece of software, it could still be read by another. A web based on proprietary, binary technologies would have been much less flexible, more difficult to search, and all information would probably now be much harder to find.

For many people, querying a web search engine is now a much more frequent operation than finding a file on their local filing system, and the development of the technology has been such that there is a reversal of the earlier situation: finding information on other people's machines is often easier than finding it on your own. Operating System and application vendors have realised this and have been working to incorporate the 'search box' into their products, and the underlying metadata directly into their filing systems, so that indexing is done as part of the process of saving a document, rather than as a separate activity. Most PCs now have enough processing and storage to perform such tasks in the background without significantly damaging their interactive responsiveness. Proactive indexing will continue to be important, but may be focussed more on data which requires substantial computation, such as that involving audio or image processing, rather than on simple indexing of textual data.

For the author, then, the search engine has taken an ever more central role in the computing experience:

1. A web site I knew about
2. A web site I bookmarked and used frequently
3. My default home page
4. Integration in the browser toolbar
5. Integration in various other applications (Mail, iTunes, Address Book)
6. (soon) Integration in Operating System taskbar and filing system

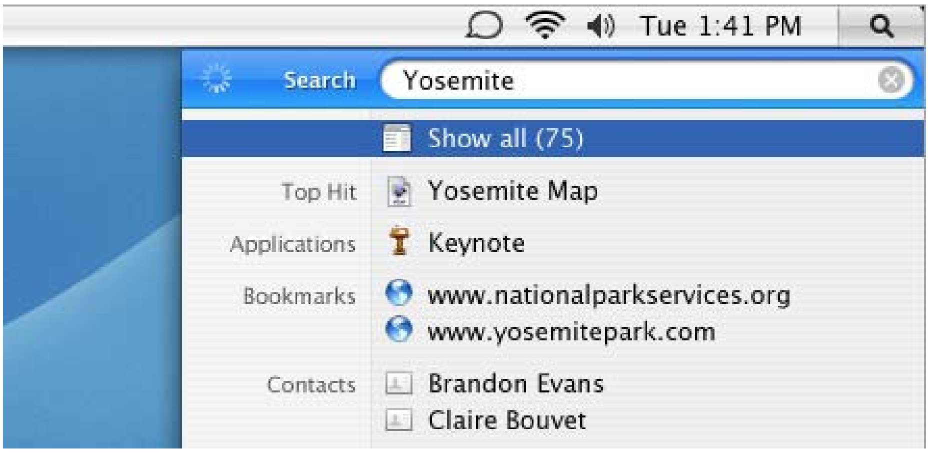


**Fig. 3.** Finding information tomorrow: the search facility built in to the toolbar of the upcoming version of Mac OS X

It is interesting to consider what step 7 might be. Could a search box augment, or even replace, the application menus themselves, providing access to functionality rather than content? Might we, in future, press a key and type 'page margins' to take us to the appropriate settings, rather than having to search menus for 'format document', or 'page setup', or 'format page', or 'document settings'? We're starting to see the first hints of this in the System Preferences dialog of the upcoming Mac OS X 10.4, where a user who doesn't know whether 'screen saver' settings can be found under 'Appearance', 'Displays', 'Energy Saver' or 'Desktop' can find them quickly by typing a few characters into a search box.

Because locating information is such an important part of everybody's daily life, from such simple tasks as finding the phone number of the local chemist to researching a topic for a book, it is important to consider carefully the implications of such a fundamental change in the way we manage our information. As users become more familiar with typing a search query to find a document, are there important analogies for other forms of input, and for other types of data?

## Known-item Searches

As more and more content is available in digital form, and an increasing amount of it is easily available on our hard disks or via the internet, a subtle change is occurring in the use of search engines. In the past, many searches were conducted to find out *whether* some information on a particular topic was available. Increasingly, we can assume that the desired item is available, that the user has a reasonably good understanding of what it is that they are looking for, and the motivation for the search is simply that searching is the most convenient way of accessing it. The search process can therefore be tailored to be very selective – relative ranking becomes less important - and thresholds can be set such that a strong correspondence between search and possible target can be taken as a definite success, perhaps resulting in a document being automatically retrieved and opened, while more ambiguous matches may be taken as a negative result.

The more certain the system can be about the user's intentions and the availability of the target within a known corpus of data, the more easily it can deal with uncertainties in another part of the system. While a large and valuable body of academic research has built up over the years regarding document clustering, new query languages, stemming and natural-language processing, user relevance feedback and so forth, it has always proved difficult to turn many of these into interfaces that the average user can understand. The key to widespread adoption of search technologies by ordinary users has often been a careful restriction of the scope of the domain to which it applies, hence the recent prominence of search boxes in particular applications which only search the data handled by that application. It has been easier to guide the user towards a specific corpus in this way than getting him or her to construct more subtle queries.

A final important example of how the 'search box' is transforming applications can be seen in Google's popular Gmail service. This web-based email system differs from traditional email programs in a couple of significant ways. It has almost no facility for deleting messages, and almost no facility for filing them in separate folders. The amount of storage provided is such that the former should not be needed, and the search facilities provided make it easy to locate messages and conversation threads even in large, unstructured archives.

## Beyond the Keyboard

Some of the most interesting commercial implementations of these ideas make use of other input methods than the traditional 'typing into a search box using a keyboard'.

And so we come to a key focus of this discussion: that input methods which are too unreliable to be generally useful for most people may still be useful when applied to a search engine in a restricted domain. As an extreme example, we would expect to be able to create a very reliable handwriting recognition system if it were known that the only phrases that would ever be written by users were the titles of songs by Simon and Garfunkel. The statement can be true, however, even when the domain in question is as broad as 'documents available on the internet', as we shall explore later.

One successful commercial example is Shazam Entertainment Ltd's music recognition service[1]. A user wishing to identify a piece of music, perhaps heard in a club or bar, can call an easily-memorable number on their mobile phone and let the music play through the mobile’s microphone. When the Shazam service has recognized the piece, it disconnects the call and sends the user a text message with the details of the song. It would be exceedingly difficult to create a system which did general transcription of music into symbolic form, and probably impossible using the low-quality audio connection provided by the phone system and considering the

[1] See http://www.shazam.com

typical environments in which the music is being sampled. But because it is known that the user is sampling audio which can probably be found somewhere in the company's database, the problem becomes tractable.

In some systems, the uncertainty comes the other way around: input from the user is reliable, but the indexing method is fuzzy. An example is the Video Mail Retrieval system developed by the University of Cambridge in partnership with the Olivetti Research Lab [Brown, Foote et al, 1994]. This used speech recognition technology to provide users with a way to search their video-mail messages. The audio in the messages would be processed off-line to provide an index of possible phonemes. The user could then type words as search queries and jump straight to the video segments where that word was spoken. Though the reliability of the speech recognition system was not sufficient to provide a useful transcript of the messages, it was able to provide a very useful service within this restricted domain. Searching video messages by any manual method is exceedingly time-consuming.

## Paper-Digital Integration

Let us now examine in more detail a particular application of the combination of unreliable recognition and content indexing. Optical Character Recognition (OCR) technologies have existed for decades but generally are still too unreliable to appeal to the general user. Except in rare cases, the effort of scanning, followed by checking and correcting errors in the electronic text, does not warrant the benefits gained. This is because in the past OCR has generally been used to create, from paper, an electronic document, when an electronic version was not otherwise available.

In today’s highly-networked world, however, it is increasingly likely that the electronic version of a document *is* available, or can be made so. In that case, the scanning of a document need serve only to identify the document, after which the electronic original can be retrieved. This deals with the problem of inaccurate OCR. But what about the inconvenience of scanning in the first place?

### Robust Locations

Phelps & Wilensky [Phelps & Wilensky, 2000] have talked for some years about ‘Robust Locations’. These are URLs that include a few words from the content at the end, so that if the URL path becomes invalid, perhaps because the document it referred to has moved from a student’s personal area to the Technical Report area, the document can still be found. An example might be:

```
http://www.something.dom/a/b/c?w=w1+w2+w3+w4+w5
```

where w1...w5 are words taken from the document and chosen to identify the document uniquely on the web site. In the event that the standard portion of the URL fails to find a file, the ‘Error 404 Document not found’ page can be replaced by something returning the results of an appropriate search query. Five words carefully chosen, in fact, will generally select a single document in the entire space of the web, and much of the work by Phelps & Wilensky and more recently by Spinellis has been on the topic of how best to choose these 'discriminant' words. In the extreme case of the *www.something.dom* site being closed down, it can be arranged to forward the last bit of any URL request to Google, so finding the old content if it still exists anywhere on the web. The nature of the language space is such that the discriminants will generally continue to identify a single document for a long time even as many more are added to the web.

The best discriminant words for a document are often chosen from amongst the most unusual words it contains, so as to improve the selectiveness of a search. The fact that they are typically taken from throughout the document, rather than from one area, also improves the robustness of the process if a part of the document is edited. Finding the optimal words can, however, be expensive.

## Scanning and searching – bringing the pieces together

The Exbiblio project[2] is also using a small number of words to locate a document. In this case, however, the words are scanned from a line of text in a printed version of the document. We therefore lose the advantages of being able to choose the 'best' identifying words in a document, meaning that we need more words to increase the chances of a unique match, and we are also adding the possibility of some OCR errors. However, we also have the knowledge that these words are in *consecutive order* in the document, which greatly reduces the space to be searched. For example, at the time of writing, a search on Google for the words "voice recorder, but also incorporates" without the quotes returns approximately 173,000 hits. A search which includes the quotes, so requiring that the exact phrase exists in the original, returns just a single page from the author's weblog[3].

We can therefore bring together a combination of the topics discussed so far:

- content-based indexing
- the ever-increasing likelihood that the electronic version of a particular document is available, or can be made available, on the internet
- the knowledge that this is a 'known-item' search because the words came from a printout of an electronic document
- an unreliable recognition technology, OCR, (whose reliability can be greatly improved by feedback from the search)

This combination allows us to create a remarkably powerful tool: a system incorporating a small scanner which can be used to scan a few words of a paper document, and can locate the electronic original of that document and display it on the screen. It could print a copy of the document. It could offer the user the opportunity to buy the document. When the words scanned are insufficient on their own to identify the source, other contextual information can be brought to bear. If a user has just scanned some words known to be from a particular document within the last few

[2] See www.exbiblio.com

[3] Once this paper is published, the user will need to scan six words instead of five to distinguish between the original page and this reference to it!

minutes, it is highly likely that the next scan will be from the same document. If a user is known to live in France, it is much less likely that they are reading the Seattle Times. Such knowledge of human factors can substantially reduce the space that must be searched.

In addition to finding the document, the system knows the location *within* the document that was scanned, and can take some action based on that knowledge. The simplest example is that a few words in a paper document can become a hyperlink. We can therefore endow paper documents with many of the characteristics of their electronic equivalents. A paper document could be used to purchase items from a catalog, to search for a dictionary definition, to request further information. Some publications may decide to print special symbols to indicate that a document is indexed or that a piece of text has extra functionality if scanned, in the same way that hyperlinks in the early days of the web were always underlined, until user behaviour started to change and people expected to be able to click on particular types of items in a page. But the system works without the need for any special marks or other embedded information in the paper copy, in the same way that HTML links do not depend on the underlining!

In short, without any changes to existing printed documents, or methods of printing, paper can become a more powerful medium than ever before.

We are currently creating prototype hardware and software to allow this vision to be realised.

## References

M. Brown, J. Foote, G. Jones, K. Sparck-Jones & S. Young, "Video Mail Retrieval by Voice: An Overview of the Cambridge/Olivetti Retrieval System" in *Proceedings of the ACM Multimedia '94 Conference Workshop on Multimedia Database Management Systems*, San Francisco, 21 Oct 1994. See also http://mi.eng.cam.ac.uk/research/Projects/vmr/vmr.html

T. Phelps and R. Wilensky, `Robust Hyperlinks Cost Just Five Words Each', UC Berkeley Computer Science Technical Report UCB//CSD-00-1091, Berkeley,USA (2000)

D. Spinellis, “Index-based Persistent Document Identifiers”, *Information Retrieval,* 8(1):5-24, January 2005

## Acknowledgements

I would like to thank Exbiblio for supporting this work, and in particular Martin King, who is the originator of most of the fundamental ideas behind the project and who also made useful comments on a draft of this paper.

# SHOP2 and TLPlan for Proactive Service Composition

Maja Vukovic and Peter Robinson

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
{firstname.lastname}@cl.cam.ac.uk

**Abstract.** Recent advances in computer technology are making the development of context aware applications possible. Such applications are complicated by the variety of contextual types that must be accommodated, together with the range of values for each context type. This makes it difficult to write and extend them. We are addressing this by building context aware applications as dynamically composed sequences of calls to Web services, considered as an AI planning problem. We identify the following three specific technical requirements for planning systems in order to handle Web service composition problem: (1) richness of domain description, (2) control constructs for assembling complex actions, and (3) a mechanism for plan optimization. In this paper, we compare two hand-coded planners, SHOP2 and TLPlan, and discuss their applicability to modelling and composing of Web services, using a specific context aware composition problem.

## 1 Introduction

The development of context-aware applications has become a complex task due to the need to accommodate for the potentially vast variety of – possibly even unanticipated – context types and their values that may be encountered. Simply hard-coding the mapping between all possible combinations of context values and the corresponding application behavior, is not only impractical, but also makes systems difficult to later extend to take into account new values of existing context types and new context types.

We are addressing this problem by constructing context aware applications as dynamically composed sequences of calls to fine granularity Web services [1]; where different service compositions of such sequences will result from different contexts such as: resources available, time constraints, user requirements and location.

By explicitly declaring Web services as processes in terms of their inputs, outputs, preconditions and effects, this paper shows how we employ goal-oriented inferencing from planning technology for service composition. We compare two hand-coded planning systems, Simple Hierarchial Ordered Planner 2 (SHOP2) [4] and TLPlan [5], and evaluate their suitability for handling the Web service composition problem.

Hand-coded planners are domain-independent planners, which use domain-specific control knowledge to help them plan effectively. SHOP2 is based on hierarchical task network (HTN) planning. The central motivation for using SHOP2 was to devise a set of (abstract) HTN methods that will encode something akin to “standard operating procedures” capturing multi-step techniques for refining a task, to further facilitate design of patterns for Web service composites. TLPlan does a forward-chaining search in which it applies planning operators to the current state to generate its successors. In contrast to SHOP2, TLPlan uses temporal logics to express search control knowledge.

The remainder of this paper is structured as follows. Section 2 analyzes related work. Section 3 defines the main requirements for the planning systems to handle composition of Web services. Based on the scenario presented in Section 4, Section 5 compares applicability of SHOP2 and TLPlan to the problem of proactive service composition. We conclude and outline areas of future work in Section 6.

## 2 Related Work

Planning technology has been used in a variety of application domains including robotics, process planning, web-based information gathering, and spacecraft mission control. It recently gained much attention to support enterprise application integration as Koehler et al. analyzed [6]. We discuss a number of related projects, which employ planning approach to Web service composition.

Automatic Web service composition using SHOP2 is also investigated by Wu et al. [7]. They observe that exclusion of concurrent processes (split and join constructs) in SHOP2 imposes a serious limitation on the usefulness of this methodology.

McIlraith et al. [8] extend Golog [9], a high level logic programming language built on top of Situation Calculus [10], for use in service composition. They use an off-line planning technique to construct general templates, which are then modified based on user preferences, resulting in a composite plan. They employ sensing actions when the knowledge of the initial state is incomplete, or many actions exist that can change world in unpredictable ways.

McDermott [11] investigates applicability of estimated regression planners, which use a backward analysis of difficulty of a goal to guide a forward search through situation space. By extending the Unpop planner [12] McDermott created Optop (”Opt-based total-order planner”). The main motivation for this work is to relax the assumption of complete knowledge required by classical planners, and to formalize what they do not know and how could they find out more about the world. McDermott also points out the necessity for planners to support synthesis of branching and looping plans.

## 3 Composition Challenges for Planning Systems

Planning systems generate (partially) ordered sequences of actions (or plans) that solve a goal. They start from a domain specification, consisting of valid action descriptions, which includes both the conditions under which an action applies (the preconditions) and the expected outcome of applying that action (the effects). Potentially enormous search space and the difficulty in fully and accurately representing real-world problems are two key challenges for the planning systems.

In this section we pose specific technical requirements that need to be met by planning systems to provide automatic, context aware, Web service composition.

### 3.1 Richness of Domain Descriptions

The Planning Domain Definition Language(PDDL) [13] is the standard, action-centered, language for the encoding of planning domains, based on STRIPS [14] formalism.

PDDL 2.1 is an extension of PDDL for expressing temporal planning domains, and is separated into different levels of expressivity. The following four are required for modelling composite Web services:

**Level 1** ADL [15] Planning: Includes the ability to express a type structure for the objects in a domain, typing the parameters that appear in actions and constraining he types of arguments to predicates, actions with negative preconditions and conditional effects and the use of quantification in expressing both pre- and post-conditions.
**Level 2** Numerical constructs: Allows for numerical variables and the ability to test and update their values instantaneously
**Level 3** Discrete durative actions: Explicit representation of time and duration.
**Level 4** Continuous durative actions: Actions with continuous effects

### 3.2 Control Constructs

Structured composite services prescribe the order in which a collection of activities (services) take place. They describe how a service is created by composing the basic activities it performs into structures that express the control patterns, data flow, handling of faults and external events, and coordination of message exchanges between service instances.

We identify the following four groups of control constructs for assembling primitive actions into a complex actions that collectively comprise an applications:

| | |
|---|---|
| 1. Sequential ordering | 2. Iteration |
| 3. Nondeterministic choice | 4. Concurrency and synchronization |

### 3.3 Plan optimization

In the real world, services consume resources, such as network bandwidth, and have a monetary cost associated with their execution. Therefore a mechanism is required to be able to present metrics and resource constraints on each service as well as the resulting plan.

## 4 Scenario: Mail Replication System

We use Web service composition to synthesize a suitable procedure for mail replication dynamically based on user location, activity, computing device and network bandwidth. Mail replication consists of two subprocesses executed in parallel: *retrieve mail* and *send mail*.

Table. 1 shows the different context types and the expected behavior of the simplified mail retrieval subprocess. Activity and the location of the user determine the presentation mode of the incoming mail. Network bandwidth, type of the computing

| | Input: Context data | | | | Output: Expected behavior |
|---|---|---|---|---|---|
| **Case** | **Activity** | **Network** | **Device** | **Location** | **Retrieve Mail** |
| **1** | Walking | GPRS | Smart Phone | Street | Display headers |
| **2** | Driving | GPRS | Embedded | In-vehicle | Read out headers |
| **3** | Not Driving | WLAN | Embedded | In-vehicle | Display adapted mail |
| **4** | Working | LAN | Laptop | At desk | Display full mail |

**Table 1.** Context and expected application behavior in simplified mail retrieval process.

device (and consequently its screen size and color depth), affect the mail retrieval. For example, rather than retrieving all the mail over the slow connection only the mail headings are initially downloaded.

## 5 SHOP2 and TLPlan for Web Service Composition

In this section we describe how SHOP2 and TLPlan can be applied to the problem of orchestrating activities (i.e. individual Web services) in order to achieve a complex task (i.e. a composite Web service). We evaluate how SHOP2 and TLPlan meet the requirements set out in Section 3, highlight their benefits and discuss their limitations.

### 5.1 Web Service Composition Using SHOP2

SHOP2 is a domain-independent HTN-based planner. It uses the idea of *hierarchical task network decomposition* to decompose an abstract task into a group of operators that forms a plan implementing the task. Planning progresses as a recursive application of the methods to decompose tasks into smaller and smaller subtasks, until the primitive tasks, which can be performed directly using the planning operators, are reached. In the case where the plan later turns out to be infeasible, SHOP2 will backtrack and try other applicable methods.

**Domain Definition** To model the reference scenario in SHOP2 we initially devised a general problem-solving strategy for mail retrieval, consisting of abstract tasks. We have then implemented these as a description of a planning domain in SHOP2, in terms of a set of axioms, operators, and methods, which prescribe how to decompose tasks., as shown in Figure 1.

An *operator* is used to indicate how a primitive task can be performed. For example, operator for mail decomposition is shown in Fig. 2(a). Operators are STRIPS-based, and are at least as expressive as Level 2 PDDL actions. In addition, each operator also has an optional cost associated with it, which can be used to find the best plan given the objective function. At present we have used the default cost 1.0.

A *method* is used to define the decomposition of a compound task into a partially ordered set of primitive or compound subtasks. Fig. 2(b) shows the method for mail processing. *Conditional expressions* in the method descriptions may be used to enumerate possible flows in the process, and therefore address the lack of branching constructs in SHOP2. This approach is however impractical for context awareness, where

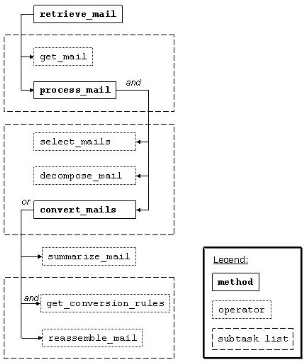


**Fig. 1.** SHOP2 Task hierarchy for the simplified mail retrieval subprocess

```
(:operator
    ;head: name and parameters
    (!decompose_mail ?mail)
    ;precondition
    (know ?mail)
    ;delete list: negative side effects
    (session_created)
    ;add list: positive side effects
    ((know ?mail_id)
     (know ?mail_header)
     (know ?mail_body)
     (know ?attachment)
     (know ?attachement_size)
     (know ?attachement_type)))
```

(a) SHOP2 operator: decompose mail

```
(:method
      ;head
       (process_mails ?list_of_mails)
      ;precondition
       ((session_created)
        (know ?list_of_mails))
      ;subtasks
       (:ordered
        (!select_mail ?list_of_mails)
        (!decompose_mail ?mail)
        (:task convert_mail ?mail)
        ))
```

(b) SHOP2 method: process mail

**Fig. 2.** Sample SHOP2 operator and method definition.

the data range has greater magnitude. Enumerating all the possible conditions that must be planned for is not only infeasible but would result in exponential growth with the number of steps in the plan.

**Problem Definition** The description of a planning problem in SHOP2 consists of an initial state and a task to be accomplished, defined in STRIPS, as shown in Fig. 3(a). The goal is the task "`retrieve_mail`", with input parameters username and password, as well as the type of the device used (e.g. `in_vehicle_inf_sys`). Context data, such as `location` and `connection_type`, also forms the description of the initial state.

**Plan** A plan consists of a list of operators, which can be applied to achieve the goal.

```
(defproblem mail_case2 mail_system(
 (activity driving)
 (location in_vehicle)
 (connection_type GPRS)
 (has bandwidth 9600)
 (device_type embedded_system)
 (embedded_system
    in_vehicle_inf_sys)
 (username john)
 (password testpswd)
 (valid_login john testpswd))
 )
((retrieve_mail
    john testpswd
    in_vehicle_inf_sys)))
```

(a) SHOP2 problem definition: case 2.

```
(!GET_MAIL SERVER1 JOHN) 1.0
(!GET_MAIL SERVER2 JOHN) 1.0
(!GET_MAIL SERVER3 JOHN) 1.0
(!SELECT_MAIL    #:?LIST_OF_MAILS1789) 1.0
(!DECOMPOSE_MAIL #:?MAIL1807) 1.0
(!SUMMARIZE_MAIL
    #:?MAIL_HEADER1819
    #:?ATTACHMENT_TYPE1820
    #:?ATTACHMENT_SIZE1821)  1.0
(!TXT_TO_SPEECH
    #:?LIST_OF_MAILS1789) 1.0
```

(b) SHOP2 plan: case 2.

**Fig. 3.** SHOP2 Problem definition and resulting plan for use case 2.

For example, plan for case 2, shown in Fig. 3(b), is a sequence of the following steps: getting the mail from three different servers, selecting mails, decomposing them, summarizing and then ”presenting” them to user using the text to speech service. In SHOP2 notation `!` is a prefix for operator symbol.

### 5.2 Web Service Composition Using TLPlan

TLPlan uses domain specific search control information to control simple forward chaining search, where the planning operators are applied to the current state to generate its successors. TLPlan therefore knows the current state of the world at every step of the planning process. Control rules, which are written in temporal logic, provide domain-specific knowledge to tell the planner which states should be avoided, therefore allowing the planner to backtrack and try other paths in the search space.

**Domain Definition** The domain definition in TLPlan, partially shown in Fig. 4(a), consists of predicate and function symbols, which can be *described* and *defined*; and operators. Predicates and functions are specified by name and their arity. For example, predicate `device_type` takes one parameter. There must be some described symbols, which are essentially predicated and functions that are updated by actions. For example `know_conversion_rules` gets updated once the knowledge of this predicate is acquired. Furthermore, one can define new predicates and symbols (in the form of first order formulas). For example the predicate `(same ?x ?y)`.

Operators, in either STRIPS or ADL form, are then specified using these declared and defined predicates and functions. They consist of list of preconditions, adds (predicates that become true) and deletes (negative effects that should be removed from the

```
;Described symbols
 (predicate device_type 1)
 (predicate know_conversion_rules 0)
;Defined symbols
 (predicate same 2)
 (def-defined-predicate (same ?x ?y)
 (= ?x ?y))
;Operator to decompose mail
(def-strips-operator
  (decompose_mail ?mail)
     (pre
        (incoming_mail ?mail)
        (has_bandwidth ?b)
        (<= ?b 9600))
     (add
        (know_mail_body mb ?mail)
        (know_attachment a ?mail)
        (mail_decomposed ?mail)))
```

(a) TLPlan domain definition

```
;; Goal
 (define (retrieve_mail_case2)
 (mail_fetched) (inbox_displayed))
;; Plan
(login john testpswd)
(get_mail john server1)
(get_mail john server2)
(get_mail john server3)
(decompose_mail mail)
(summarize_mail mail)
(txt_to_speech mail)
(mail-retrieved)
```

(b) TLLPlan problem and resulting plan.

**Fig. 4.** TLPlan domain, problem and plan for use case 2.

world).

**Problem Definition** Problem definition in TLPlan is similar to that in SHOP2, and consists of predicates and functions describing the initial state of the world.

In contrast to SHOP2, where the goal is the task (i.e. method) to be achieved, in TLPlan the goal is specified using a list of predicate and function specifications, as shown in Fig. 4(b).

**Plan** The plan generated by TLPlan is quite similar to the one generated by SHOP2, shown in Fig. 3(b). The key difference is the order of operators. This is mainly due to the method abstractions and their ordering constructs used by SHOP2 to define the subtasks.

### 5.3 SHOP2 and TLPlan Comparison

In this section we distill some general observations about SHOP2 and TLPlan and their applicability to Web service composition, given the experience with implementation of

[1] ADL includes the ability to express a type structure for the objects in a domain, typing the parameters that appear in actions and constraining the types of arguments to predicates, actions with negative preconditions and conditional effects and the use of quantification in expressing both pre- and post-conditions.

[1] Bacchus et al. [16] extend TLPlan to handle concurrent actions (with variable duration).

| | **Planning System** | |
|---|---|---|
| **Feature** | **SHOP2** [4] | **TLPlan [5]** |
| **Planning methodology** | HTN | Forward chaining |
| **Richness of domain descriptions** | | |
| PDDL 2.1 Level 1: ADL Planning | ✓ | ✓ |
| PDDL 2.1 Level 2: Numeric Constructs | ✓ | ✓ |
| PDDL 2.1 Level 3: Discrete Durative Actions | ? | ✓ |
| PDDL 2.1 Level 4: Continuous Durative Actions | ? | ✓ |
| **Control Constructs** | | |
| Sequential Ordering | ✓ | ✓ |
| Iteration | ✓ | ✓ |
| Concurrency and Synchronization | ✓ | ?[1] |
| Non-deterministic choice | × | ? |
| **Plan optimization** | ✓ | ✓ |

**Table 2.** Suitability of SHOP2 and TLPlan for Web service composition.
Legend: × = not supported, ? = partially supported or work-around available, ✓ = fully supported

the sample scenario. Table 2 compares relevant features of each planner based on the criteria set out in Section 3.

**Planning methodology and its implications** SHOP2 and TLPlan are both hand-coded planning systems, however they differ in the kind of control knowledge they use. SHOP2 employs HTN methods to guide which parts of the search space should be explored. On the other hand, TLPlan uses the temporal formulas to tell which part of the search space should be avoided. SHOP2's HTN approach gives more structure to the domain and the way a problem should be solved. Furthermore, this concept could be exploited to create patterns of composite Web service.

The main disadvantage of both planners is that whilst hand-coded search does help them plan effectively, it creates a significant overhead. Consequently it requires expertise in both the domain and specifics of the planner, and therefore put limitations on level of automatization of Web service composition process.

**Expressiveness and support for PDDL** These two planning systems have equivalent expressive power and are similar in many respects. They are both Turing-complete, because they allow function symbols. Furthermore both SHOP2 and TLPlan allow attached procedures and numeric computations. They know the current state at each step of the planning process, and use this to prune operators. Both SHOP2 and TLPlan support external subroutines.

TLPlan is capable of reading the problem definition and generating the plan in this format, but does not support PDDL-based domain specification. SHOP2 supports actions of at least Level 2 in PDDL, and even though it does not provide explicit support of the durative actions in Level 3 of PDDL, it has sufficiently expressive power to represent durative and concurrent actions given the following three characteristics.

**Control constructs** While SHOP2 allows for tasks to be sequentially ordered, there is no mechanism to handle the control constructs related to concurrency, namely: parallel split, synchronization and exclusive choice. At the moment this is resolved by enumerating every possible flow in the process using conditional expressions in the method descriptions. This increases the complexity of search space, and planning. Bacchus et al. [16] extend TLPlan to handle concurrent actions (with variable duration).

**Parametric overloading** A further syntactical issue is the problem of parametric overloading, where a number of operators have the same name but different signatures, nevertheless providing the same functionality. It is not supported by SHOP2 at the operator level. Whilst there is a workaround, the lack of support for parametric overloading conflicts with the conceptual model where we associate the planning operators with executable (Web) services. (Because planning involves matching each operator description this concept is not commonly supported by conventional planning systems, as an optimization of the search process.)

**Goal representation** In contrast to TLPlan, in SHOP2 the goal can not be stated declaratively. SHOP2 has to know in advance which method it should call. Consequently the planner fails if asked to solve a completely new, unknown problem for which no method definition exists.

**Domain and problem complexity** The number of the axioms in the problem description impairs planner's performance. In our experiment each problem definition is described with a limited set of facts and our domain is highly simplified, with the intention to keep the search space minimal.

This raises one of the central challenges in optimizing the composition process—where does the information about the state of the world come from and at which point of time? For example, how and when does one retrieve axioms describing the attachment conversion rules for the mail system? One approach is to create a set of so-called "sensing" actions, which when necessary retrieve additional axioms about the world, as McIlliarth et al. [8] demonstrate.

## 6 Conclusion and Future Work

We are tackling the increasing complexity required for context awareness by building context aware applications through the dynamic, planning-based, composition of Web services.

In this paper, we compared two hand-coded planning systems, SHOP2 and TLPlan for their suitability to automate Web service composition, based on the following three technical requirements: (1) richness of domain description, (2) control constructs for assembling complex actions, and (3) a mechanism for plan optimization.

By composing a specific context aware application automatically, using SHOP2 and TLPlan planners, we identified the lack of complex control structures involving concurrency, iteration and nondeterministic choice to generate expressive compositions as the key shortcoming. Another open problem arises from the deterministic nature of SHOP2 and TLPlan, as they assume that that the state of the world is always accessible, static and deterministic. In contrast, Web services tend to create new objects at runtime, and this needs to be accommodated for.

Our future work will involve investigating nondeterministic planners, motivated by the unpredictability of pervasive computing environments.

**Acknowledgements.** IBM Zurich Research Laboratory is supporting this work.

## References

1. Vukovic, M., Robinson, P.: Adaptive, Planning Based, Web Service Composition for Context Awareness. In: Advances in Pervasive Computing. Volume 176. (2004) 247–252
2. Shaw, M., Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice-Hall, Inc. (1996)
3. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (1995)
4. Nau, D., Munoz-Avila, H., Cao, Y., Lotem, A., Mitchell, S.: Total-Order Planning with Partially Ordered Subtasks. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco (2001) 425–430
5. Bacchus, F., Kabanza, F.: Using Temporal Logic to Control Search in a Forward Chaining Planner. In: Proceedings of Second International Workshop on Temporal Repre sentation and Reasoning (TIME), Melbourne Beach, Florida (1995)
6. Koehler, J., Srivastava, B.: Web service composition: Current solutions and open problems. In: ICAPS 2003 Workshop on Planning for Web Services. (2003)
7. Wu, D., Sirin, E., Hendler, J., Nau, D., Parsia, B.: Automatic Web Services Composition Using SHOP2. In: 13th International Conference on Automated Planning & Scheduling. Workshop on Planning for Web Services., Trento, Italy (2003)
8. McIlraith, S., Son, T.: Adapting Golog for Composition of Semantic Web Services. In: Eighth International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France (2002)
9. Levesque, H.J., Reiter, R., Lesperance, Y., Lin, F., Scherl, R.B.: GOLOG: A Logic Programming Language for Dynamic Domains. Journal of Logic Programming **31** (1997) 59–83
10. McCarthy, J., Hayes, P.J.: Some Philosophical Problems From the Standpoint of Artificial Intelligence. In Meltzer, B., Michie, D., eds.: Machine Intelligence 4. Edinburgh University Press, Edinburgh (1969) 463–502
11. McDermott, D.V.: Estimated-Regression Planning for Interactions with Web Services. (In: AI Planning Systems Conference (AIPS)) 204–211
12. McDermott, D.V.: A Heuristic Estimator for Means-Ends Analysis in Planning. (In: AI Planning Systems Conference (AIPS)) 142–149
13. Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL—The Planning Domain Definition Language (1998)
14. Fikes, R.E., Nilsson., N.J.: STRIPS: A new approach to the application of theorem proving to problem solving. Artifical Intelligence **2** (1971) 189–208
15. Pednault, E.: ADL and the state-transition model of action. Journal of Logic and Computation (1994)
16. Bacchus, F., Ady, M.: Planning with Resources and Concurrency: A Forward Chaining Approach. In: IJCAI. (2001) 417–424

# Exact Learning Geometric Objects

Nikolai Yu. Zolotykh

University of Nizhni Novgorod,
Gagarin ave. 23, Nizni Novgorod, 603950, Russia,
zny@uic.nnov.ru,
WWW home page: http://www.uic.nnov.ru/~zny

**Abstract.** We give an overview of some results concerning exact identification of geometric objects over discrete instance space (domain) using membership queries.

## 1 Introduction

The problem of learning geometric concepts is one of the most extensively studied topics in computational learning theory. The most popular models of concept learning are inductive inference, Valiant's Probably Approximately Correct (PAC) learning model [16] and Angluin's exact learning model [1, 2].

In PAC model the learner is given a reasonable number of labeled instances. The learner's task is to generate, with high probability of success, a hypothesis that is a good approximation of the target concept using a reasonable amount of time.

In exact learning model, the learner asks queries in order to exactly identify the target. The most common types of queries are membership queries and equivalence queries. The learner asks an equivalence query by presenting his hypothesis to an oracle (teacher). In return, the oracle answers either 'yes', if the hypothesis coincides with the target concept, or otherwise a counterexample that witnesses the difference. The learner asks a membership query supplying an instance. An oracle only replies 'yes' or 'no' according to whether the instance is or is not in the target concept.

The model of exact learning is closely connected with condinional tests [6] and with the problem of deciphering an unknown function in some known class of discrete functions (see, for example, [10, 14]).

In geometric concept learning, an instance space (domain) is arbitrary finite non-emty subset of $\mathbf{R}^n$. Usually, a hyper-cube $\{0, 1, \ldots, k-1\}^n$ is considered as the domain. Any subset of the domain is called a concept. Arbitrary non-empty family of such concepts is called a concept class (see, for example, [11] where PAC learning geometric concepts is considered).

Here we consider exact learning geometric concepts with membership queries.

## 2 Preliminaries

Let $n \geq 2$, $k \geq 2$ be natural numbers. We write $E_k^n = \{0, 1, \ldots, k-1\}^n$ to denote the set of all points (vectors) $x = (x_1, \ldots, x_n)$ such that $x_j$ are integers and $0 \leq x_j \leq k-1$ $(j = 1, 2, \ldots, n)$. The set $E_k^n$ is considered as an *instance space*. Any $c \subseteq E_k^n$ is called a *concept* over $E_k^n$. Arbitrary non-empty family $C \in 2^{E_k^n}$ of such concepts is called a *concept class*.

In the paper we consider Angluin's *exact learning model* [1] with *membership queries*. In this model the learning process is considered as a game of two persons: a learner and an environment. The goal of the learner is to identify an unknown target concept $c$ chosen from a known concept class $C$, making membership queries ("Is $x \in c$?" for some $x \in M$) and receiving yes/no answers. Suppose learning algorithm $\mathcal{A}$ to identify a concept $c$ uses $\mathrm{MEMB}(\mathcal{A}, c)$ queries and $\mathrm{COMP}(\mathcal{A}, c)$ computational time. The *learning complexity* of a learning algorithm $\mathcal{A}$ is the maximum number of queries it makes, over all possible target concepts $c \in C$, i.e.

$$\mathrm{MEMB}(\mathcal{A}) = \max_{c \in C} \mathrm{MEMB}(\mathcal{A}, c).$$

The *learning complexity* $\mathrm{MEMB}(C)$ of a concept class $C$ is the minimum learning complexity, over all learning algorithms for this class, i.e.

$$\mathrm{MEMB}(C) = \min_{\mathcal{A}} \mathrm{MEMB}(\mathcal{A}).$$

Analogously, we can define a *computational complexity* of a learning algorithm $\mathcal{A}$ and a *computational complexity* of a concept class $C$:

$$\mathrm{COMP}(\mathcal{A}) = \max_{c \in C} \mathrm{COMP}(\mathcal{A}, c), \qquad \mathrm{COMP}(C) = \min_{\mathcal{A}} \mathrm{COMP}(\mathcal{A}).$$

We remark that the problem of exact learning of a concept can be obviously reduced to the problem of conditional test construction [6] and deciphering a function of $k$-valued logic (see, for example, [10, 14]).

There are other models of learning, for example, Valiant's Probably Approximately Correct (PAC) learning model [16].

A set $T \subseteq M$ is said to be a *teaching set* for a concept $c \in C$ with respect to the class $C$ if no other concept from $C$ agrees with $c$ on the whole $T$, i.e. for any $f \in C \setminus \{c\}$ it holds that $(c \setminus f) \cap T \neq \emptyset$ or $(f \setminus c) \cap T \neq \emptyset$. If a teaching set is of minimum cardinality, over all teaching sets for a concept $c$, then we call it *minimum teaching set* for $c$. If no subset $D$ of $T$ $(D \neq T)$ is a teaching set for $c$ with respect to the class $C$ then we call $T$ *minimal teaching set* for $c$. It is obvious that for any concept class $C$ and for any concept $c \in C$ any minimum teaching set is minimal. Denote by $\mathrm{TD}(c, C)$ the cardinality of a minimum teaching set for a concept $c$. Define

$$\mathrm{TD}(C) = \max_{c \in C} \mathrm{TD}(c, C).$$

The quantity $\mathrm{TD}(C)$ is called *teaching dimension* for the class $C$. We also consider *the average cardinality of minimum teaching set*, $\overline{\mathrm{TD}}(C)$ which is defined as

$$\overline{\mathrm{TD}}(C) = \frac{1}{|C|} \sum_{c \in C} \mathrm{TD}(c, C).$$

It is clear that

$$\mathrm{MEMB}(C) \geq \mathrm{TD}(C).$$

The concept $c \subseteq E_k^n$ is called a *half-space* over $E_k^n$ if there exist real numbers $a_0, a_1, \ldots, a_n$ such that

$$c = \left\{ x \in E_k^n : \quad \sum_{j=1}^{n} a_j x_j \leq a_0 \right\}. \tag{1}$$

The inequality in (1) is called a *threshold inequality* for $c$. The hyper-plane $\sum_{j=1}^{n} a_j x_j = a_0$ is called *threshold hyper-plane.* Denote by $\mathrm{HS}_k^n$ the set of all half-spaces over $E_k^n$. Each half-space over $E_k^n$ is a concept. The class $\mathrm{HS}_k^n$ is a concept class. We will say that a learning algorithm *exact identifies* the target concept $c \in \mathrm{HS}_k^n$ if it determines some numbers $a_0, a_1, \ldots, a_n$ for which (1) holds.

The concept $c \subseteq E_k^n$ is called an *$m$-hedron* over $E_k^n$ if $c$ can be represented as a intersection of $m$ halfspaces, i.e. if there are real numbers $a_{ij}$ $(i = 1, 2, \ldots, m;$ $j = 0, 1, \ldots, n)$ such that

$$c = \left\{ x \in E_k^n : \quad \sum_{j=1}^{n} a_j x_{ij} \leq a_{i0} \quad (i = 1, 2, \ldots, m) \right\}. \tag{2}$$

Denote by $m$-$\mathrm{HS}_k^n$ the set of all $m$-hedrons over $E_k^n$. We will say that a learning algorithm *exact identifies* the target concept $c \in m$-$\mathrm{HS}_k^n$ if it determines some numbers $a_{ij}$ for which (2) holds.

The concept $c \subseteq E_k^n$ is called a *ball* over $E_k^n$ if there are a point $y \in E_k^n$ and an integer $r$ such that

$$c = \left\{ x \in E_k^n : \quad \sum_{j=1}^{n} (x_j - y_j)^2 \leq r \right\}. \tag{3}$$

Denote by $\mathrm{BALL}_k^n$ the set of all balls over $E_k^n$. We will say that a learning algorithm *exact identifies* the target concept $c \in \mathrm{BALL}_k^n$ if it determines some numbers $r, y_1, \ldots, y_n$ for which (3) holds.

# 3 Identification of Half-Spaces

## 3.1 Sequentional Algorithms

Shevchenko [13] proposed a learning algorithm that exact identifies any concept $c$ in $\mathrm{HS}_k^n$ using *poly*$(\log k)$ membership queries and *poly*$(\log k)$ computation steps ($n$ is fixed). Hegedüs [8] proved that Shevchenko's algorithm requieres $O(\log^{(n-1)\lceil \frac{n}{2} \rceil + n} k)$ queries.

**Theorem 1.** [9, 22] *There exists a learning algorithm that exact identifies any concept $c \in \mathrm{HS}_k^n$ using $O(\log^n k)$ membership queris and polynomial in $\log k$ running time ($n$ is fixed).*

We remark that from results of Moshkov in the test theory [12] it follows (see [9]) that

$$\mathrm{MEMB}(\mathrm{HS}_k^n) = O\left(\frac{\log^n k}{\log\log k}\right)$$

but Moshkov's algorithm requires exponential running time (even for fixed $n$).

For the case of $n = 2$ there are efficient polynomial algorithms [5, 17, 21] that exact identifies a half-plane $c \in \mathrm{HS}_k^2$ using $O(\Theta(\log k))$ membership queries and polynomial in $\log k$ running time. The best known result follows.

**Theorem 2.** [18] *There exists a learning algorithm that exact identifies any concept $c \in \mathrm{HS}_k^2$ using at most $6\log(k-1)+4$ membership queris and $O(\log k)$ computational operations.*

### 3.2 Teaching Dimension and Lower Bounds

To obtain lower bounds for the learning complexity $\mathrm{MEMB}(C)$, the following simple inequalities are widely used:

$$\mathrm{MEMB}(C) \geq \log |C|, \qquad \mathrm{MEMB}(C) \geq \mathrm{TD}(C).$$

Applying the well-known bound

$$\log |\,\mathrm{HS}_k^n\,| = \Theta(n^2 \log k) \tag{4}$$

($n$ is fixed) to the first of mentioned inequalities we get $\mathrm{MEMB}(\mathrm{HS}_k^n) = \Theta(n^2 \log k)$. For $n = 2$ the bound (4) can be made more precise [14]:

$$|\,\mathrm{HS}_k^2\,| \sim \frac{6}{\pi^2} k^4 \qquad (k \to \infty).$$

This implies $\mathrm{MEMB}(\mathrm{HS}_k^2) \geq 4 \log k$. Now we obtain the following.

**Corollary 1.**

$$\mathrm{MEMB}(\mathrm{HS}_k^2) = \Theta(\log k).$$

A characterization of teaching sets for the class $\mathrm{HS}_2^n$ is proposed in [3]. This result was extended to the class $\mathrm{HS}_k^n$ for any $k \geq 3$ in [15]. Denote by $N(c,a)$ the set of vertices of $\mathrm{Conv}\{x \in c : \sum_{j=1}^n a_j x_j \leq a_0\}$ where $a = (a_0, a_1, \ldots, a_n)$. Denote by $M(c,a)$ the set of vertices of $\mathrm{Conv}\{x \in E_k^m \setminus c : \sum_{j=1}^n a_j x_j \geq a_0 + 1\}$.

The point $x \in E_k^n$ is called *essential* for the concept $c \in C$ with respect to the concept class $C$ if there exists a concept $f \in C \setminus \{c\}$ such that $c \setminus \{x\} = f \setminus \{x\}$. In [15] it is proved that a minimal teaching set $T(c)$ for a concept $c$ with respect to $C$ is the set of all essential points. Moreover,

$$T(c) = \bigcup_a \left(N(c,a) \cup M(c,a)\right),$$

where the union is over all $a = (a_0, a_1, \ldots, a_n) \in \mathbf{Z}^{n+1}$ such that the inequality $\sum_{j=1}^n a_j x_j \leq a_0$ is a threshold inequality for $c$.

This characterization yelds the following.

**Theorem 3.** [15] *If $n$ is fixed then*

$$\mathrm{MEMB}(\mathrm{HS}_k^n) \geq \mathrm{TD}(\mathrm{HS}_k^n) = \Omega(\log^{n-2} k).$$

We remark that $\mathrm{TD}(\mathrm{HS}_k^2) = 4$.

The set $X \subseteq \mathbf{R}^n$ is called *simple* if $X$ contains no integer point except vertices. The characterization of simple sets in $\mathbf{R}^3$ proposed in [7] implies the following.

**Proposition 1.** [20] *Minimal teaching set of a concept $c \in \mathrm{HS}_k^3$ consists of the points that lie on the parallel planes. One of these planes contains all the points of $T \cap c$ set and the other contains all the points of $c \setminus T$.*

This proposition allows us to obtain the asymptotic estimation of $\mathrm{TD}(\mathrm{HS}_k^3)$.

**Proposition 2.** [20] *If $k \to \infty$ then*

$$\mathrm{TD}(\mathrm{HS}_k^3) \sim 12 \log_{1+\sqrt{2}} k.$$

The mentioned results on $\mathrm{TD}(\mathrm{HS}_k^n)$ can be compared with the following theorem concerning the average cardinality of minimum teaching set, $\overline{\mathrm{TD}}(\mathrm{HS}_k^n)$.

**Theorem 4.** [19]

$$\overline{\mathrm{TD}}(\mathrm{HS}_k^n) \leq n^2 \log k.$$

This extends the bound $\overline{\mathrm{TD}}(\mathrm{HS}_2^n) \leq n^2$ obtained in [3].

### 3.3 Parallel Algoritnms

Bshouty and Cleve [4] introduced a model of exact lerning in parallel. This model is called *UPRAM* that is a variant of PRAM (parallel random access machine) extended to allow for queries. An UPRAM with $p$ processors can ask $p$ queries in one learning step (see [4] for details).

It is not hard to see that the algorithm in [9, 22] can be made suitable for UPRAM with $O(\log^{n-1} k)$ processors ($n$ is fixed). Hence, there exists a learning algorithm for UPRAM with $O(\log^{n-1})$ processors that exact identifies any concept $c \in \mathrm{HS}_k^n$ using $O(\log k)$ lerning steps and polynomial in $\log k$ running time ($n$ is fixed).

The following theorem concerns learning concepts in $\mathrm{HS}_k^2$.

**Theorem 5.** [18]

1. *There exists a learning algorithm for UPRAM with $2$ processors that exact identifies any concept $c \in \mathrm{HS}_k^2$ using at most $3\log(k-1)+2$ learning steps and $O(\log k)$ computational operations.*

2. *There exists a learning algorithm for UPRAM with $4$ processors that exact identifies any concept $c \in \mathrm{HS}_k^2$ using at most $2\log(k-1)+1$ learning steps and $O(\log k)$ computational operations.*

## 4 Identification of Other Geometric Objects

It is clear that in order to exact identify a concept $\emptyset \in m$-$\mathrm{HS}_k^n$ it is necessery to ask queries in each point of a domain $E_k^n$. Therefore, $\mathrm{MEMB}(m\text{-HS}) = k^n$. Howerever, suppose that before a learning process a learner is provided by some points belonging to target concept. This assumption allows to obtain some positive results.

Denote by $m$-$\mathrm{HS}_{k,\rho}^n$ the set of polygons $P$ in $m$-$\mathrm{HS}_k^n$ such that the angle $\alpha$ between any two adjacent edges of the polygon $P$ satisfies $\rho \leq \alpha \leq \pi - \rho$ and each edge of $P$ has length at least $16 \cdot \lceil 1/\rho \rceil$. Bultman and Maass [5] propose a learning algorithm to exactly identify any $c \in m$-$\mathrm{HS}_{k,\rho}^n$ from any given point in $c$ with $O(m(1/\rho + \log k))$ membership queries. The number of computation step of the algorithm is bounded by a polynomial in $m$, $\log k$ and $1/\rho$. This result can be compared with the following.

**Theorem 6.** [18] *There exists a learning algorithm that exact identifies any target concept $c \in m$-$\mathrm{HS}_k^2$ provided by 3 non-collinear points in $c$ using at most $(10m + 1)\log(k - 1) + 34$ membership queris and $O(m \log k)$ computational operations.*

An other result relates to the class of balls.

**Theorem 7.** [18] *There exists a learning algorithm that exact identifies any target concept $c \in \mathrm{BALL}_k^2$ provided by any point in $c$ using $O(n \log k)$ membership queries.*

## References

1. Angluin, D.: Queries and concept learning. Machine Learning (2) (1988) 319–342
2. Angluin, D.: Queries Revisited. Lectures Notes in Artificial Intelligence 2225 (2001) 12–31
3. Anthony, M., Brightwell, G., Shawe-Taylor, J.: On specifying Boolean functions by labelled examples. Discrete Applied Mathematics 61 (1) (1995) 1–25
4. Bshouty N. H., Cleve R.: Interpolating arithmetic read-once formulas in parallel. SIAM J. Comput. 27 (2) (1998)
5. Bultman, W. J., Maass, W.: Fast identification of geometric objects with membership queries. Information and Computation 118 (1) (1995) 48–64
6. Chegis, I. A., Yablonskii, S. V.: Logical methods of electric circuit control. Trudy MIAN SSSR 51 (1958) 270–360 (Russian)
7. Chirkov, A. Yu., Veselov, S. I.: The structure of simple sets of a three-dimensional integer lattice // Automation and Remote Control 3 (2004) (to appear)
8. Hegedüs, T.: Geometrical concept learning and convex polytopes. Proceedings of the 7th Annual ACM Conference on Computational Learning Theory (COLT'94). ACM Press New York (1994) 228–236
9. Hegedüs, T.: Generalized teaching dimensions and the query complexity of learning. Proceedings of the 8th Annual ACM Conference on Computational Learning Theory (COLT'95). ACM Press New York (1995) 108–117

10. Korobkov, V. K.: On monotone functions of logic algebra. Cybernetics Problems. "Nauka" Moscow 13 (1965) 5–28 (Russian)
11. Kwek, S.: Geometric Concept Learning and Related Topics (Ph. D. Thesis). University of Illinois at Urbana-Champaign, Department of Computer Science, Technical report UIUCDCS-R-97-1980 (1997)
12. Moshkov, M. Yu.: Conditional tests. Cybernetics Problems. "Nauka" Moscow 40 (1983) 131–170 (Russian)
13. Shevchenko, V. N.: Deciphering of a threshold function of many–valued logic. Combinatorial–Algebraic Methods in Applied Mathematics. Gorky (1987) 155–163 (Russian)
14. Shevchenko, V. N.: Qualitative Topics in Integer Linear Programming. "Fizmatlit" Moscow (1995). English transl.: AMS Providence Rhode Island (1997)
15. Shevchenko, V. N., Zolotykh, N. Yu.: Lower Bounds for the Complexity of Learning Half-Spaces with Membership Queries. Algorithmic Learning Theory: 9th International Conference; proceedings. Michael M. Richter etc. (ed.). Springer (1998) (Lecture Notes in Computer Science; Vol. 1501: Lecture Notes in Artificial Intelligence) 61–71
16. Valiant, L. G.: A theory of the learnable. Commun. ACM, 27 (11) (1984) November 1134–1142
17. Veselov, S. I.: A lower bound for the mean number of irreducible and extreme points in two discrete programming problems. Manuscript No. 619–84, deposited at VINITI Moscow (1984) (Russian).
18. Veselov, S. I., Zolotykh, N. Yu.: Identification of geometric objects in the plane (in preparation)
19. Virovlyanskaya, M. A., Zolotykh, N. Yu.: An upper bound for the average cardinality of minimum teaching set of a threshold function of many-valued logic // Bulletin of University of Nizhni Novgorod. Mathematical modeling and optimal control 1 (26), Nizhny Novgorod, University of Nizhni Novgorod (2003) 238–246
20. Virovlyanskaya, M. A., Zolotykh, N. Yu.: On the cardinality of the teaching set of a threshold function of multivalued logic. VI Internatopnal Comgress on Mathematical Modeling / Book of Abstracts / September 20-26, 2004, Nizhny Novgorod, University of Nizhni Novgorod (2004) 381
21. Zolotykh, N. Yu.: An algorithm of deciphering a threshold function of $k$-valued logic in the plane with the number of calls to the oracle $O(\log k)$. Proceedings of the First International Conference "Mathematical Algorithms". NNSU Publishers Nizhny Novgorod (1995) 21–26 (Russian)
22. Zolotykh, N. Yu., Shevchenko, V. N.: On complexity of deciphering threshold functions. Discrete Analysis and Operations Research. Novosibirsk 2 (1) (1995) 72–73 (Russian)