RWTH Aachen

Department of Computer Science
Technical Report

Utilizing optical sensors from mice for new input devices

Sebastian Ullrich, Jakob T. Valvoda and Torsten Kuhlen

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2006-15

RWTH Aachen · Department of Computer Science · December 2006

Utilizing optical sensors from mice for new input devices

Sebastian Ullrich, Jakob T. Valvoda and Torsten Kuhlen

Virtual Reality Group
RWTH Aachen University, Germany
Email: `s.ullrich@rz.rwth-aachen.de`

Abstract. This paper introduces approaches towards the development of intuitive input devices built with optical sensors from mice. The two main ideas are spatial reconfiguration of the sensor and advanced interpretation of the sensor data. We present new concepts for various input devices that are easy to rebuild. A prototype for kinematics manipulation of virtual characters has been constructed and algorithms for data interpretation have been implemented and evaluated.

1 Introduction

Optical mice are very popular as desktop input devices and also getting quite inexpensive. Little attention has been payed to the fact that a small digital camera is integrated into the sensor chip. This component allows the construction of many new exciting input devices, due to its compact footprint and integrated digital signal processor. In this paper we provide information about the technical capabilities of these sensors, and give suggestions on how to use and further enrich these capabilities with image processing algorithms. The main goal is to use the simple yet technically advanced sensor in areas where it has not been initially designed for (cp. Figure 1).

In the following section we will present some related work on input devices. The main part describes several technical and algorithmical changes and improvements for optical mouse sensors. Then different areas of application with several devices are motivated. Finally we discuss the results and give a conclusion.

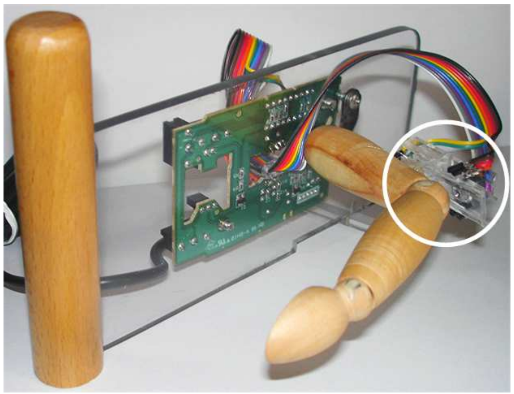


Fig. 1. Prototype constructed from parts of a manikin with an optical mouse sensor, used in order to measure joint rotation

2 Related work

The description of related work is divided into a brief motivation for specialized input devices and concludes with a compilation of resources on how to build new input devices.

2D or 3D mice are general purpose input devices. Obviously, they are not always the best solution for every multidimensional interaction task. One improvement is to design sophisticated input devices for specific tasks in order to increase efficiency and maximize precision of the interaction. This proposal has been motivated in a design guideline by Paley [Pal98]. Further information on design and construction of new input devices has been described in [FL03,BKLP04].

Generally, there are two different approaches: start from scratch or dismantle and rearrange an existing input device:

- In the first case, one has to start by planning the communication between input device sensors and the computer. Due to this fact, there are virtually no technical limitations to the design of the new device, but electrical engineering skills are required. There are some interface kits that are quite helpful to avoid soldering, but are also limited to certain amount and types of sensors [GF01]. Additionally Ballagas et al. present a framework for the construction of wireless devices [BRSB03].
- In the second approach, an existing input device is reconfigured. This allows for an easier start and often faster construction because the interface and sensors are already given. For example, the SpaceMouse is a popular base for new devices like the Cubic Mouse [FP00].

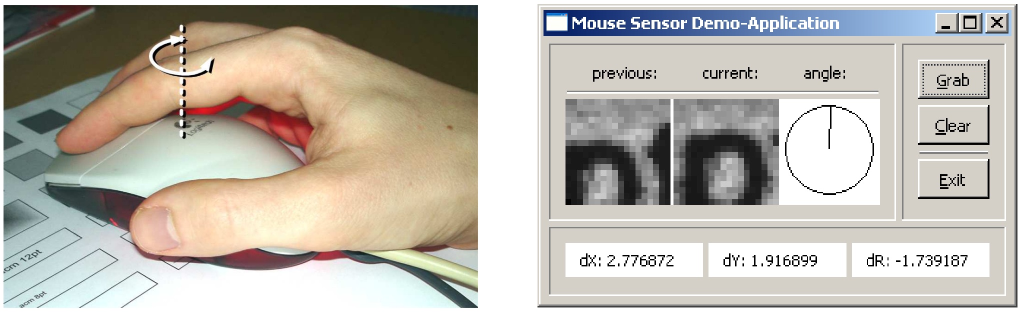
Using a specialized input device instead of an abstract metaphor increases user acceptance in most cases. Thus, we focused our efforts on the research of input devices and tried to further supplement this approach with low-cost electronic components.

3 Working with optical sensors

Our key ideas are to utilize electronic components from optical mice, derive additional information from the sensor and furthermore rearrange the sensors in novel spatial compositions. After a brief description of the technical background, the ideas are explained in detail.

3.1 Technical background

The sensor of an optical mouse consists of a miniaturized digital camera, which is integrated into a small semiconductor chip (typically with an 8 or 16 pin layout). To digitize an image, the chip first triggers a red LED or in newer devices a laser light that illuminates the surface. Then the reflected light is bundled by a convex lens and exposed to the optical sensor. Consecutive images are by default processed by the integrated digital signal processor in order to compute their relative translation between each other. The resolution of the pictures ranges from 18×18 to 30×30 pixels with a color depth of 64 grey values. With a recommended distance of 2.4 mm from lens to object surface 400 to 800 dpi are achieved.



dX: 2.776872

dY: 1.916899

dR: -1.739187

Fig. 2. Mouse with parallel port interface for image capturing (left); demo application displaying the last and the current frame with rotation results retrieved from image registration (right)

3.2 Advanced data interpretation

It is possible to capture the current raw pixel image from the optical sensor. To obtain the image, the pixel values can be read from the chip via its serial data I/O-Port. This allows to use the raw image-stream for further interpretation like pattern recognition, or to calculate the rotation around the normal of the image plane.

In a test-setup we have established communication with a mouse sensor through the parallel port of a computer using two data-bits: a clock and a half-duplex data-line connected to the according pins of the chip (further details can be found, e. g., in [Tec04]). Alternatively, instead of the parallel port a microcontroller can be connected to the chip, which is then able to communicate in the same way.

In our scenario, we capture images and process them in a computer application. For two consecutive images, we obtain the angle of rotation by image registration techniques which are commonly used in image processing (we have used ITK [ISNC05] for this purpose). Thereby we do measure a third degree of freedom (cp. Figure 2), in addition to the two already available translative values that are computed by the internal signal processor of the sensor chip.

3.3 Sensor rearrangement

For rapid prototyping of new input devices the chip can be unsoldered and rewired from the mouse to fit into smaller devices. Thus, a flexible and compact (approx. 10 mm×10 mm×20 mm) optical unit consisting of the sensor combined with the plastic lens and LED can be focused onto any kind of surface. We have used this technique in order to create our prototypes described in subsequent sections.

3.4 Communication Interface

For fast construction of new input devices the interface of the disassembled mouse can be reused to communicate with a computer. Another solution—especially needed for multiple sensors or image capturing—is the utilization of an interface with a microcontroller. We provide a brief description of our own multi-purpose interface design:

- cyclic iteration is triggered by a clock in the controller chip,
- the values of each sensor are singled out by a combination of a binary counter and an array of multiplexer units,
- the resulting raw data is stored in registers of the microcontroller,
- on requests the values are read by the communication controller and sent to the computer.

A corresponding communication protocol has been established and the microcontroller was programmed accordingly. The whole setup has been successfully tested with an ATMEL AT90S8515 microcontroller in combination with Code Mercenaries I/O-warrior as the communication controller. The protocol specifies how to iterate over sensors, encode the data, split it into packets and send these to USB or to a TCP/IP-server (e. g., for a WLAN connection).

4 Image registration results and discussion

Because of the low resolution from the captured images, we have measured the image registration capabilities. We have conducted tests with pictures grabbed from the mouse sensor (cp. first two rows of Figure 3). In order to evaluate the precision of the image registration results we have created difference images (cp. third row in Figure 3). The inverse transform of the registration results (cp. Table 1) has been applied to the moving image. For better precision of the transformation we have subsampled the images without interpolation (to maintain the original values). Finally, to create the difference image, the transformed image has been subtracted from the reference image. As can be seen in the tested examples the computed rotation matches in the majority of cases.

In general, the registration is highly dependent on the magnitude of the transformation and the similarity of patterns that are present in successive pictures. For this reason we have tested different surfaces on their applicability. Because randomly seeded, recognizable features are best suited, we evaluated the performance of a sheet of paper with tightly printed text. The results can be seen in columns a-c of Figure 3. In the worst case scenario of a blank surface without any features (cp. column d of Figure 3) the image registration algorithm did not yield convincing results.

In order to enhance the initial registration capability, we intend to resolve instability and errors by supplementary parameters for image registration with predicted movement information derived from previous samples. Furthermore, we plan to apply filtering techniques to enhance patterns and remove random noise.

Table 1. Results from image registration (cp. Figure 3)

figure	angle	δx	δy	iterations
3.a	-2.168°	1.314	2.653	73
3.b	-1.139°	0.082	-0.160	48
3.c	0.898°	-2.368	5.379	80
3.d	-0.593°	-0.173	0.135	10

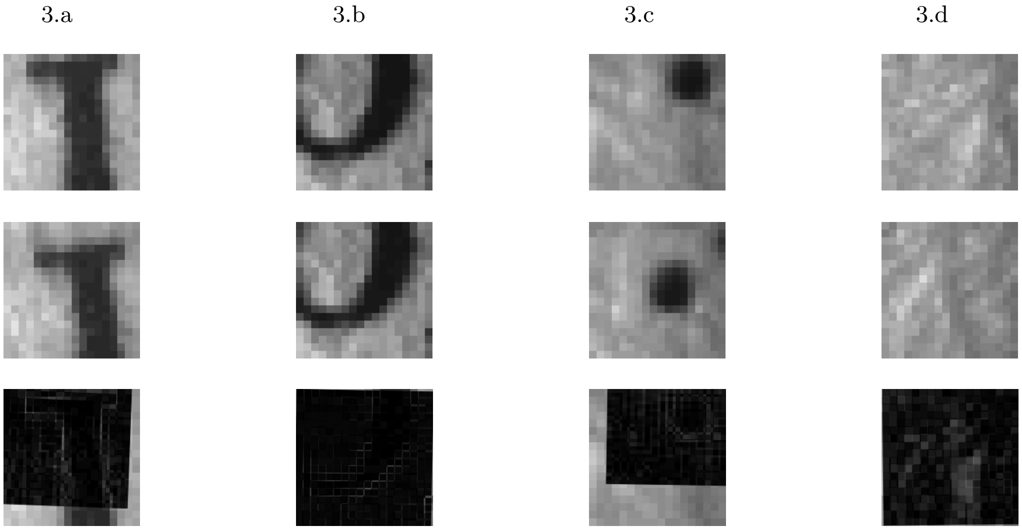


Fig. 3. First and second row: reference images from the mouse sensor; third row: subsampled difference images illustrating the results from image registration (cp. Table 1)

5 Devices and applications

There are different fields of application where the reconfigured optical sensor can be utilized. In this section we will show examples for the presented techniques.

5.1 Optical Manikin Input device

Whilst in a typical mouse environment the surface is plane, evenly curved surfaces can also be used for special purposes. For such spherical surfaces it is straightforward to register rotations in two degrees of freedom. In order to compute the angular values (α_x, α_y) in degree, the pixel distance in x- and y-direction are interpreted as two arc angles and normalized by division through the radius r of the spherical target:

$$\alpha_x = \frac{180}{r \cdot \pi} \cdot \delta x, \quad \alpha_y = \frac{180}{r \cdot \pi} \cdot \delta y$$

The pixel values (δx, δy) are calculated by the integrated chip of the sensor and are supplied by the mouse driver software or can optionally also be fetched from the sensor by a microcontroller. This approach allows to reuse and augment an existing interface and interpret the returned values differently.

As a proof of concept the Optical Manikin Input device (O.M.I.) was constructed (cp. Figure 1 and Figure 4). In this setup an optical sensor is tracking limbs from a wooden manikin. The unit is mounted on a fixed adjacent segment and positioned with the lens towards the spherical surface of the movable wooden swivel head (cp. Figure 4). Thereby the sensor measures the joint movement relative to the segment.

5.2 Avatar manipulation

Avatars are commonly used for interaction and collaboration in virtual environments. To enrich communication with gestures the avatar should move accordingly. Examples are: pointing to a virtual object of interest, or to give a figurative

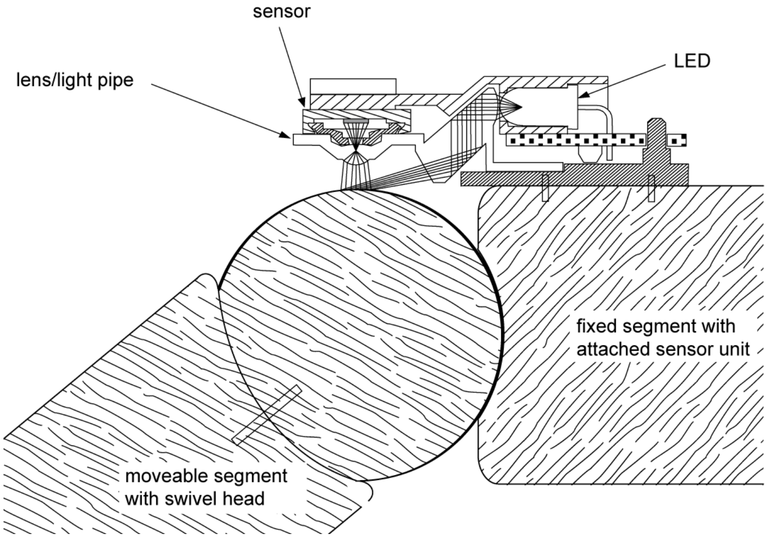


Fig. 4. Sectional view of an optical mouse sensor [Tec04] attached to a segment and focused on a joint

description. We have used O.M.I. to control one arm. In this case, the resulting rotations derived from the sensors are mapped to the according joints of a virtual character in real-time. To process the data we implemented a test-application using the avatar module VRZula [VKB06] and the API of the interface controller (cp. Figure 5). Furthermore, O.M.I. can also be used for character animation in a similar manner. A 6DOF-senor of a tracking-system is attached to the pelvis of the manikin, in order to track the global position and orientation. This allows for rapid creation of spatial keyframes.

5.3 Desktop mouse with a twist

Another scenario involves our proposed image processing to upgrade a "normal" mouse literally with an additional twist. With such a device objects can be translated either in a 2D- or a 3D-application on a plane as usual and can be additionally rotated around the plane normal by twisting the mouse (cp. Figure 2). This can be used for docking- and aligning-tasks.

Fig. 5. Example of using O.M.I. (left) to bend the arm of a virtual character (middle and right)

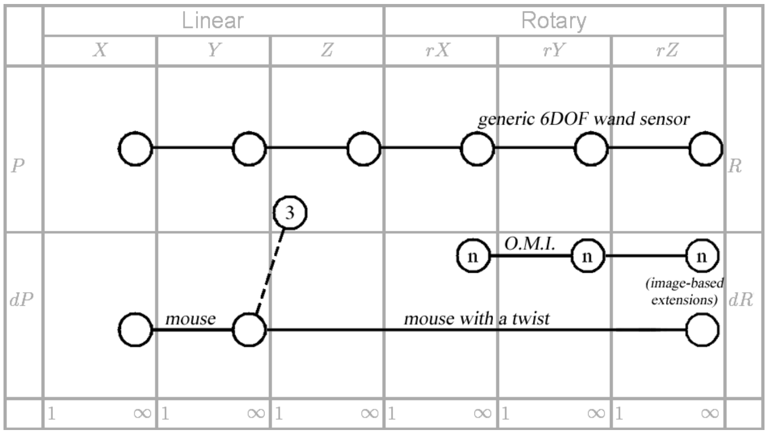


Fig. 6. Classification in the design space [CMR91]

6 Classification

In order to compare our prototype O.M.I. and the proposed mouse with a twist with other more generic input devices like a desktop mouse or a tracked 6DOF wand, we have chosen a taxonomy with physical, mechanical and spatial attributes by Card et al. [CMR91]. The resulting classification can be seen in Figure 6. A mouse differs from our O.M.I. device in the design space significantly, although they are based on the same sensor and interface. Also, the proposed augmented mouse with twist-detection clearly does not replace a 6DOF wand, but still allows for new interaction possibilities.

7 Conclusion and future work

In this paper we have described the utilization of optical sensors from mice. As a proof of concept for rapid prototyping and spatial reconfiguration an input device for avatar manipulation has been constructed. Image registration algorithms have been successfully tested with raw images captured in real-time from a mouse sensor. Further devices—based on the proposed alternate usage of sensors from existing hardware—have been motivated for other application areas.

We intend to continually improve the robustness of the image registration by implementing a runtime error-estimation and also experiment with different starting parameters. It is planned to evaluate the presented devices for user acceptance with the tasks motivated in the applications section. The studies for avatar manipulation will be conducted in large immersive environments and compared with other interaction metaphors like inverse kinematics or real-time motion capture mapping.

8 Acknowledgments

The authors would like to thank Adam Bielecki for helping with his electro engineering skills and Jeroen Domburg for his insightful discussion on interfacing the mouse sensor via parallel port.

References

[BKLP04] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*, chapter 4.7. Home-Brewed Input Devices. Addison-Wesley Professional, 2004.

[BRSB03] Rafael Ballagas, Meredith Ringel, Maureen Stone, and Jan Borchers. istuff: A physical user interface toolkit for ubiquitous computing environments. In *Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems*, 2003.

[CMR91] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems*, 9(2):99–122, 1991.

[FL03] Eric Forman and Cynthia Lawson. Building physical interfaces: Making computer graphics interactive. SIGGRAPH Course 30, 2003.

[FP00] Bernd Froehlich and John Plate. The cubic mouse: A new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 526–531, 2000.

[GF01] Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the Symposium on User Interface Software and Technology*, pages 209–218, 2001.

[ISNC05] Luis Ibanez, William Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, second edition, 2005.

[Pal98] W. Bradford Paley. Designing special-purpose input devices. *Computer Graphics*, 32(4):55–59, 1998.

[Tec04] Agilent Technologies. Agilent ADNS-2610 optical mouse sensor data sheet, 2004.

[VKB06] Jakob T. Valvoda, Torsten Kuhlen, and Christian Bischof. Interactive virtual humanoids for virtual environments. In *EGVE'06: Proceedings of the 12th Eurographics Symposium on Virtual Environments*, pages 9–12, Lisbon, Portugal, 2006.

Aachener Informatik-Berichte

This list contains all technical reports published during the past five years. A complete list of reports dating back to 1987 is available from http://aib.informatik.rwth-aachen.de/. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

2001-01 * Jahresbericht 2000
2001-02 Benedikt Bollig, Martin Leucker: Deciding LTL over Mazurkiewicz Traces
2001-03 Thierry Cachat: The power of one-letter rational languages
2001-04 Benedikt Bollig, Martin Leucker, Michael Weber: Local Parallel Model Checking for the Alternation Free mu-Calculus
2001-05 Benedikt Bollig, Martin Leucker, Thomas Noll: Regular MSC Languages
2001-06 Achim Blumensath: Prefix-Recognisable Graphs and Monadic Second-Order Logic
2001-07 Martin Grohe, Stefan Wöhrle: An Existential Locality Theorem
2001-08 Mareike Schoop, James Taylor (eds.): Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modelling
2001-09 Thomas Arts, Jürgen Giesl: A collection of examples for termination of term rewriting using dependency pairs
2001-10 Achim Blumensath: Axiomatising Tree-interpretable Structures
2001-11 Klaus Indermark, Thomas Noll (eds.): Kolloquium Programmiersprachen und Grundlagen der Programmierung
2002-01 * Jahresbericht 2001
2002-02 Jürgen Giesl, Aart Middeldorp: Transformation Techniques for Context-Sensitive Rewrite Systems
2002-03 Benedikt Bollig, Martin Leucker, Thomas Noll: Generalised Regular MSC Languages
2002-04 Jürgen Giesl, Aart Middeldorp: Innermost Termination of Context-Sensitive Rewriting
2002-05 Horst Lichter, Thomas von der Maßen, Thomas Weiler: Modelling Requirements and Architectures for Software Product Lines
2002-06 Henry N. Adorna: 3-Party Message Complexity is Better than 2-Party Ones for Proving Lower Bounds on the Size of Minimal Nondeterministic Finite Automata
2002-07 Jörg Dahmen: Invariant Image Object Recognition using Gaussian Mixture Densities
2002-08 Markus Mohnen: An Open Framework for Data-Flow Analysis in Java
2002-09 Markus Mohnen: Interfaces with Default Implementations in Java
2002-10 Martin Leucker: Logics for Mazurkiewicz traces
2002-11 Jürgen Giesl, Hans Zantema: Liveness in Rewriting
2003-01 * Jahresbericht 2002
2003-02 Jürgen Giesl, René Thiemann: Size-Change Termination for Term Rewriting
2003-03 Jürgen Giesl, Deepak Kapur: Deciding Inductive Validity of Equations

2003-04	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Improving Dependency Pairs
2003-05	Christof Löding, Philipp Rohde: Solving the Sabotage Game is PSPACE-hard
2003-06	Franz Josef Och: Statistical Machine Translation: From Single-Word Models to Alignment Templates
2003-07	Horst Lichter, Thomas von der Maßen, Alexander Nyßen, Thomas Weiler: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung
2003-08	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Mechanizing Dependency Pairs
2004-01 *	Fachgruppe Informatik: Jahresbericht 2003
2004-02	Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic
2004-03	Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting
2004-04	Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming
2004-05	Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming
2004-06	Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming
2004-07	Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination
2004-08	Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information
2004-09	Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity
2004-10	Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules
2005-01 *	Fachgruppe Informatik: Jahresbericht 2004
2005-02	Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: "Aachen Summer School Applied IT Security"
2005-03	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions
2005-04	Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem
2005-05	Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots
2005-06	Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information
2005-07	Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks
2005-08	Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut

2005-09	Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures
2005-10	Benedikt Bollig: Automata and Logics for Message Sequence Charts
2005-11	Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture
2005-12	Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems
2005-13	Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments
2005-14	Felix C. Freiling, Sukumar Ghosh: Code Stabilization
2005-15	Uwe Naumann: The Complexity of Derivative Computation
2005-16	Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)
2005-17	Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)
2005-18	Thomas von der Maßen, Klaus Müller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlußbericht des GI-Arbeitskreises “Features”
2005-19	Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers
2005-20	Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.
2005-21	Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited
2005-22	Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins
2005-23	Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves
2005-24	Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks
2006-01 *	Fachgruppe Informatik: Jahresbericht 2005
2006-03	Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler
2006-04	Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation
2006-05	Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F
2006-06	Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color
2006-07	Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group "Requirements Management Tools for Product Line Engineering"

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices

[*] These reports are only available as a printed version.

Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.