Hans Weghorn (Ed.)

Information Technology & Computer Science

Proceedings of the 4th Annual Meeting on Information Technology & Computer Science at the BA-University of Cooperative Education

Stuttgart, Germany, February 2008
Preface

The fourth edition of the annual meeting on

Information Technology and Computer Science of the
Berufsakademie Baden-Württemberg – University of Cooperative Education

hosted a full day of interesting presentations. Although the topic range appears rather broad at first glance, there are as well converging denominators in these reports due to the matter of subject. For instance, several of the presentations addressed very recent research fields like robotics and automotive in an amazing way. Doubtlessly, there is also close interrelation between the latter contents and topics like embedded systems, sensors, and graphical and signal processing, especially when concepts about improved HCI or autonomous behaviour of systems is under investigation.

I want to thank all authors for their working efforts, which they invested and which helped bringing our fourth ITCS meeting to success. In addition, I want to thank all attendees and guests for their interest in this event. This booklet shall preserve an impression about all the interesting and elaborated scientific contributions of our fourth ITCS event.

Meeting chair – ITCS 2008

Hans Weghorn
### Table of Contents

1 .......... Analysis of Texture Synthesis Algorithms with Respect to Usage for Hole-Filling in 3D Geometry  
    *Monika Kochanowski, Philipp Jenke, and Wolfgang Straßer*

7 .......... Algorithms and Protocols for Secure Embedded Networking  
    *Axel Sikora*

12 ........ Managing Geo Data for Location-based Services – The Hybris Framework  
    *Jörg Roth*

17 ........ Emotion and Personality in Driver Assistance Systems  
    *Dirk M. Reichardt*

22 ........ OMIcar - Demonstration Platform for VANET Protocols  
    *Andreas Schmeisser, Yvonne Günter, Bernhard Wiegel, Matthias Rabel, and Hans Peter Großmann*

27 ........ The RAW Image File Format Problem - Applications of Digital SLR Cameras in Astronomy and Science  
    *Thilo Bauer*

33 .......... Subtyping in Java 5.0  
    *Martin Plümicke*

38 .......... Brain-Imaging: Measurement of Oxidative Processes with Magnetic Resonance  
    *Hannes Wiesner*

39 .......... Localization of Autonomous Robots with On-Board Sensors  
    *Reinhard Lafrenz*

40 .......... Soccerbot Demonstration  
    *Claus Engelhardt, Tristan Zur, Dominik kleine Kalvelage, Lars Sonnabend, and Dirk M. Reichardt*

41 .......... Author Index
Analysis of Texture Synthesis Algorithms with Respect to Usage for Hole-Filling in 3D Geometry

Monika Kochanowski, Philipp Jenke, Wolfgang Straßer

WSI/GRIS, University of Tübingen, Germany
KoMonika@gmx.net, {pjenke, wstrasser}@gris.uni-tuebingen.de

Abstract. The development of range scanning technology has offered new possibilities to capture large datasets in computer graphics. Due to physical restrictions, captured 3D objects can contain holes that need to be filled automatically. In a different area of interest, the field of texture synthesis, research has been done on a solution for a similar 2D problem. To utilize available methods, the challenge is to find similarities and show the differences between these problems. Hereby a 3D prototype is developed on the basis of an analysis of chosen texture synthesis algorithms, projecting the problem onto an intermediate 2D algorithm. The presented solution works fast and at good quality on artificial and real-world examples.

1 Introduction

Although the technology of new range scanning devices develops quickly, physical limitations will always constrain the quality of the collected data. Holes in objects can also be introduced in the post-processing steps, for example by manual editing or combination of different point sets. Many of the proposed 3D hole-filling algorithms use a surface estimation in the filling process. As this step is computationally expensive, it is reasonable to attempt an experimental approach without it.

Another interesting research field is texture synthesis, which offers many statistical similarity-based approaches working on reconstruction of a larger sample of a smaller input texture with the goal of perceptual similarity [13]. In the last years many different algorithms have been proposed to solve that problem.

Here, we develop a prototype for a simple yet efficient 3D hole-filling algorithm. Therefore existing texture synthesis algorithms are analyzed to identify solutions which are applicable for the 3D problem. The results are presented on real-world examples. Finally, an outlook and a conclusion are given.

2 Related work

In the well-known approach of context-based surface completion [15] a hierarchical algorithm is used by refining a surface estimation. The algorithm in [14] uses a digital signature based on curvature and texture information for the reconstruction process.
Another algorithm developed in [2] uses a voxel-based approach, trying to copy the most similar point for each region.


The field of main interest here is texture synthesis. The algorithms can be classified into three categories: pixel-based, patch-based, and advanced approaches. The basis for many of the later algorithms is the pixel-based approach of Efros and Leung [6]. The theoretical foundation is the Markov random field (MRF) model, which mathematically describes the probability distribution for one pixel \( p \) assuming locality, therefore only considering its neighborhood. In the reconstruction process, for each missing pixel \( p \) the input is searched for the possible best neighborhoods, which are then sampled to synthesize \( p \). An enhancement is given in [19], where a multi-resolution approach is used to improve speed and quality especially for using large pixel neighborhoods. In [1] an algorithm for synthesizing natural textures is proposed. The idea is to add information about structure in a second input picture. In [7], image analogies are used to describe the relationship of two input images to a different set of output images. All pixel-based algorithms have in common that one pixel is synthesized in one algorithm step.

In the patch-based approach image quilting [5], overlapping texture patches are copied before the best cut through the boundary of the overlapping region is found by dynamic programming. A similar method is used in [9], whereby here graph cuts are applied. The hybrid texture synthesis algorithm [13] uses a patch-based approach starting with large patches and subdividing these when no good match can be found. The overlapping parts are repaired pixel-based. The adoption in [12] accelerates this method by using only large patches and a pre-computed search structure in the pixel repair step.

Surveyed advanced algorithms contain texture optimization [8], vector field visualization [16], feature matching and deformation [20], order-independent texture synthesis [18], parallel controllable texture synthesis [11], and appearance-space texture synthesis [10].

The chosen algorithm, which serves as a basis here, has to be general, simple, adaptable, and perform acceptably. Therefore, neither the pixel-based approaches are used, which are inherently slow because of the neighborhood search step, nor the advanced algorithms are considered, which use complicated interacting methods to improve the search result. The patch-based algorithm hybrid texture synthesis [13] offers good results with a relatively simple approach and looks promising for 3D adoption.

3 Hole-filling algorithm

To ensure applicability, a 2D experimental prototype is developed as the basis for the 3D hole-filling algorithm using the main ideas of hybrid texture synthesis [13]. The splitting of the patches with the overlap strategy has shown a performance problem in some hole-filling applications, when for example a 16 x 16 pixel patch is subdivided in 1 x 1 pixels using an overlap of 2, a total of 1280 pixel reconstructions have to be done to reconstruct
256 pixels. The large patches introduce two problems: First, the quality of the best match decreases during the filling process. Secondly, one bad guess distorts the complete picture. To solve the first problem, the patches are chosen based on their generation, as older patches contain more original and less reconstructed information. Among these the patch with the highest variance is used, propagating smooth regions slower. The overlapping patch part is corrected pixel-based. To solve the second problem, a multi-resolution pyramid is created to initialize the hole pixels well. Two results of the algorithm for example images are presented in fig. 1: Linear structure is visible on the right example that is propagated well, and the natural setting in the left picture is filled nicely.

Fig. 1. Results of the 2D prototype for example images from [3] (left) and [4]

To adapt the algorithm to work on 3D objects, the new main step is *patch creation*. Using 3D points as input data, the problem has to be projected down to 2D. In a preprocessing step, the considered region of a possibly large data set can be manually selected. A certain portion is selected randomly as basis for patches. The patch is then considered to be in a sphere around the center in radius \( r \). A principal component analysis (PCA) is employed to find the three main axes describing the chosen points. Therefore, a patch is considered as a plane, and the third dimension is the height over the patch plane. In figure 2 an example of two patches with their respective coordinate axis is shown.

Fig. 2. Example sphere with two marked patches

The gray value of the 2D representations has been estimated by the distribution of height values over the plane: white is highest, black lowest. Squares of the patch where no points fit are considered as broken, marked as red points in the 3D picture, shown yellow in the 2D height field.

Further on, the algorithm processes patches in the same manner as the 2D experimental prototype (see figure 3): First, for the to-be-repaired broken patch, the best-matching complete patch is chosen. The missing points in the 3D data set are created using the height information from the complete patch.
In an optional pixel-based correction step, the pasted points can be smoothed to better fit into the given shape. After the patch has been fixed, all patches which can be affected by the newly pasted points need to be updated. Further on, the set is extended by using some of the pasted points as new starting points for patches in order to fill holes, which might be hit randomly only by few patches, completely.

4 Results

After some artificial examples including a cube and a bumpy sphere, a complete range scan of a Neptun statue [21] is chosen for reconstruction. One can see in figure 4 that
most of the marked area is reconstructed well. The small white holes in the face reconstruction have not been captured by the identification of broken patches. In the foot reconstruction the distribution of the reconstructed points is similarly dense as the given input data. One can see that most holes have been filled nicely.

The short reconstruction times underneath one minute for all examples show the good performance of the algorithm. The reason therefore is the restriction of the sample set to a well-defined region, the random sampling procedure which uses only a portion of the available points as patch, and the 2D height field, which does not store 3D information. The implemented prototype has been embedded in an existing library [17] and therefore is integrated with various existing functionality.

Areas of improvement are a sharp-edge problem, as these patches are sometimes considered as broken. This could be improved by using an additional density metric for bad patch identification. Further on, an overlap strategy for smoother integration could be used as in the 2D prototype. Finally, the pixel correction step could be performance enhanced.

5 Conclusion

In this paper a 3D hole-filling prototype has been developed on the basis of a survey of existing texture synthesis techniques. The results presented show that the simple algorithm is applicable for a variety of settings and offers good performance considering the size of the input data.

References

21. AIM@SHAPE Shape Repository V4.0. http://shapes.aimatshape.net/ (date: 03/29/2007)
Algorithms and Protocols for
Secure Embedded Networking

Axel Sikora
University of Cooperative Education, 79539 Lörrach, Germany
sikora@ba-loerrach, http://www.ba-loerrach.de/

Abstract. Albeit security is a major challenge for embedded systems in industrial, building and consumer applications, to-day most of the networked embedded devices run at a terribly low security level. This is not mainly due to the restricted resources on embedded systems, but appears to be a question of perception and knowledge of embedded designers. For the integration of security measures into embedded systems, a variety of additional requirements must be taken into account. Those are presented and discussed in this contribution with a special focus on embedded (inter)-networking. Selected solutions of the authors are presented, i.e. for embedded SSL.

1 Introduction

The integration of Internet connectivity and web services into embedded systems brings many potential advantages with it [1]. It combines two enabling technologies, both with low cost and high efficiency. It is crucial to understand that all use cases can be solved with legacy technologies as well. However, the major driver to use embedded web services lies in the huge potential to save cost, as they allow the integration of applications and communication into one device, the use of well known TCP/IP (and HTTP) Internet protocols with the low-cost and seamless integration with Ethernet, the quick, easy and solid development of HTML-based graphical user interfaces that can be deployed locally or remotely, and the use of standard web browsers for remote monitoring, diagnose and control. However, a number of aspects should be regarded when using embedded systems in networks:

- Security is a major issue: Many embedded web servers may give access not only to data but may allow the control of devices, machines, and factories [5]. This results in the challenge to provide the highest possible security level at lowest cost [4].

- The risk of being attacked comes with the mere connectivity to the Internet [6] as there is a large number of systematic port scanners and as the data is transported via the public Internet. Therefore, the risk only partially depends on the application itself.

- Scalability is another issue. On the one hand, Internet protocols allow an optimum degree of scalability, as computing performance may vary from an 8 bit microcontroller to a 64 bit high end enterprise server without changing communication protocols or access mechanisms. Although the communication protocols remain the same for all these levels, the approach for security may differ significantly.
Cryptography of course is a major means to tackle the security issue. The implementa-
tion of cryptography brings some peculiarities with it [3]. The evolution of security threats and new means of deployment and therefore to provide possibilities to update an embedded system.

This contribution demonstrates how embedded SSL can help to overcome these issues (cf. ch. 2). It then discusses the advances in microcontroller technology (cf. ch. 3) presents one implementation (cf. ch.4).

2 SSL – pro’s and con’s

Secure Socket Layer (SSL)-based Virtual Private Networks (VPN) promise many advantages to embedded systems, such as:

- Embedded system can be accessed with standard software such as a web browser.
- SSL allows client authentication without storing passwords on the embedded system. This is an essential benefit since embedded systems might operate in remote, inaccessible locations, where the exchange of secured data storage is not possible.
- SSL provides a variety of encryption standards that might be implemented mandatory. System designers find a balance between security risk, capabilities of the system and the strength of the encryption.
- SSL provides client as well as server authentication.

One major drawback of SSL proves to be the symmetric key generation that is computation intense and can not be easily and efficiently sourced out in hardware. On the other hand asymmetric encryption would increase the load during communication so that the drawback of the intense key generation must be taken into account.

The effort to provide client authentication must be considered carefully. Even though the authentication of clients on the server does not add a significant effort on the design, it requires the use of a PKI to distribute certificates to authorized clients. This effort has to be considered carefully when designing a system. However, these advantages help to overcome practical problems as with [2].

3 Advances in microcontroller technology

In terms of long term security the use of configurable hardware such as Field Programmable Gate Array (FPGA) is the best choice. This enables a scalable hardware – software design. The use of encryption/decryption functions in hardware relieves the CPU significantly. Renowned manufacturers of FPGAs with optimized 32 bit CPU cores are Altera with its NIOS II platform [8] and Xilinx [9].

The drawback of FPGAs in comparison with standard microcontrollers is increased power consumption and higher prices per logic function (which might be alleviated through monolithic system integration), an increased complexity and the need for hardware synthesis tools.
As a result for a major number of embedded applications the choice lies still at standard microcontrollers. The following trends among standard microcontrollers can be observed:

- Increase of performance: 32 bit microcontrollers are more and more used with internal CPU frequency in the range of 50 MHz and higher.
- Increase of monolithic integration of communication peripherals such as Ethernet AC and PHYs.
- Increase of monolithic integration of hardware support for crypto algorithms such as ES, RSA, DES/3DES, or random number generation [10].
- Increasing memory capacities on the chip.

The requirements to the software in regards of the still limited resources of embedded systems and of the vast variety of platforms can be seen in:

- Efficient implementation in terms of execution speed and memory usage.
- Portability can be achieved by a smart software interface design.

Portability with the definition of software interfaces unfortunately contradicts with efficient implementations. However, only portable (and thus reusable) software can be maintained at reasonable costs. The implementation of the crypto wrapper is of high importance. It provides the means of generating target specific code. The network security layer must be able to access the same functions on different platforms. The implementation of the crypto library, however, may vary from one platform to another. If a platform provides hardware support then the crypto wrapper is configured to access those functions in order to offload the CPU. Only the functions that are not available in hardware are then run completely in software. Furthermore this enables the implementation of time critical functions in CPU optimized code such as assembly language.

4 Realization
4.1 Implementation
The emBetter TCP/IP stack [7] has been developed since many years at the institute of the author. It boasts a number of enhancements, e.g. high-speed and real-time extensions. Now, a fully standard compliant SSL-module has been developed, which allows

- Diffie-Hellman-(DH)- and RSA-based key exchange
- symmetric encryption using 3DES, AES or RC4.

The emBetterSSl is already used in various projects that require legal certification for their security mechanisms, e.g. money gambling machines and health card readers. Those are either based on Altera-NIOS [8] or Atmel-ARM9 [10]-microcontrollers.

4.2 Memory Management
The memory management of the emBetter suite is completely kept static. This is an absolute requirement to all microprocessor systems without a memory management unit (MMU). As a consequence, memory fragmentation does not become an issue. In addition, care must be taken concerning the dynamic execution of program code, i.e. stack behavior. Whereas this issue can be handled with a reasonable margin, the coop-
eration with the application software tends to be more error-prone. No single solution can be provided, except extensive testing (cf. ch. 4.6).

4.3 Mutual Authentication

In legacy computing systems, e.g. business IT systems, servers communicate with many clients. One-side authentication seems to be sufficient for the computing devices, as authentication can be ensured by user related credentials. In addition, as servers can be physically secured, user-related information may be stored on them.

In embedded computing, machine-to-machine communication is much more relevant. Thus, devices itself must be authenticated. This should be done on a certificate level. Mutual certificate-based authentication is supported by emBetterSSL.

4.4 Physical Security

Legacy computing systems are separate machines, which can be centralized and secured in dedicated premises, whereas embedded systems are applied in their final system - and thus cannot be mechanically secured with major efforts to keep cost at a reasonable level. Major means to ensure the integrity of the systems are:

- use of microcontrollers with internal flash memory,
- use of internal flash memory, which can be secured against unauthorized read-out,
- use of additional physical measures to secure the microcontroller or an attached chip card.

It should be clear that all of these measures can be tempered at the attacker’s expense. emBetterSSL comes with different solutions to this issue. However, they strongly depend on the target system and application.

4.5 Update

Running machines for years and decades also implies the necessity of updates. They must be secured so that

- the update can be accessed only after a successful mutual authentication,
- the update is stable, before the older version is removed.

emBetterSSL may use the mutually authenticated SSL-secured communication channel via http file upload. Security against loss of data or interrupt of flashing process is achieved via flash mirroring, i.e. running double program memory.

4.6 Testing

Albeit theoretical analyses are the fundament for a reasonable long-term stability and security, testing remains a major practical means to find weaknesses and loopholes.
emBetterSSL comes with an extended test-suite covering the functional use cases. In addition, further experience is to be collected with port-scanner like Nessus [11]. Due to non-standard implementation and no OS-related loopholes, the results are very relaxing. However, one should not forget about system specific weaknesses.

5 Summary

Long term security for embedded systems is closely related with the use of encryption technologies. More detailed discussion can be found in [12].

The use of standard protocols vs. proprietary protocols promises the use of standard software on remote systems and thus cost reduction on the overall system. On the other hand there is no standard hardware platform on an embedded system and the variety of implementations is big. In some cases the embedded system must provide a continuous uptime so that the update of software is only partially possible.

The use of SSL or TLS on embedded systems is of special interest since it does not require passwords stored within the embedded device. Even though SSL requires significant resources embedded systems are capable to apply it due to increased processor performance, the break through of 32bit platforms and the implementation of hardware accelerators such as crypto co-processors.

References

11. http://www.nessus.org/
Managing Geo Data for Location-based Services – The Hybris Framework

Jörg Roth

University of Applied Sciences Nuremberg
Kesslerplatz 12, D - 90489 Nuremberg
Joerg.Roth@ohm-hochschule.de

Abstract. Even though geo data are getting more and more widely available nowadays, they often do not meet the requirements of location-based services concerning structure, content and format. Often, an application provider thus has to modify or adapt geo data. Moreover, applications still have to deal with the position capturing process and the sensor fusion task. In this paper we present the Hybris framework that provides a geo data authoring environment and position reasoning services that significantly simplify the development of location-based services.

Keywords: Location-based Services, Geo Data, Positioning, Spatial Reasoning

1 Introduction

Geo data form the natural resource for location-based services. Geo objects describe the world in terms of natural, artificial and virtual entities that cover the earth's surface. Important functions such as displaying maps, maintaining points of interests (POIs) and navigation strongly rely on the quality of geo data.

Managing geo data becomes a more and more difficult tasks. A huge variety of sources are available that range from land survey data (e.g. ATKIS), freely available geo data (e.g. OpenStreetMap), privately assembled data such points of interests collected with the car navigation device or GPS tracks stored in GPS loggers. These data have different characteristics concerning quality, structure and contents, thus data not only have to be imported from different formats, but have to be corrected, fused, enriched and finally integrated into a data repository.

In this paper we present the Hybris environment. It is built to collect and manage geo data for location-based services; it contains a component to access positioning systems and provides a location reasoning mechanism.
2 The Hybris Environment

The Hybris environment connects the components External Geo Data, Authoring, Positioning and Applications as illustrated in Fig. 1.

Fig. 1. The Hybris environment

- The geo data authoring tool is called depro. Besides display and modification functions it provides a powerful plug-in interface that allows the easy integration of new components. In addition to the stand-alone editor, a Web-based variation based on Google Maps is realized.
- The Domain Repository stores the corresponding geo data. We identified a number of requirements that the geo data model should meet (see below). In the current implementation, the domain repository simply consists of a directory tree of XML files, but also the storage in spatial databases is conceivable.
- A mechanism called $MAP^3$ allows to reason about the geo data in relation to the user's current position. As position sensors have certain sensor error distributions, reasoning about a position is a probabilistic mechanism. $MAP^3$ provides an efficient approach to represent and process position probability densities and in contrast to Kalman filters also support non-Gaussian densities.

The next sections provide details about these components.

2.1 The Authoring Environment

The traditional field that deals with geo data is land surveying. As nowadays land survey data is stored digitally, it can in principle be used for location-based services. However, the goals of land surveying often do not meet the requirements of location-based services, we thus introduce a new geo data model [3]. We identified a number of require-
ments that the geo model should meet, extended the traditional properties of geo data, which are thematic, geometric and topological properties and introduced structural, organizational and meta data properties.

Commercial systems in the area of location-based services often import geo data from hundreds of geo spatial databases. One of our goals was to represent all required geo information within a single data format. To get the required expressiveness, we first extended the list of traditional properties of geo objects (i.e. geometric, thematic and topological properties). The resulting list of properties (excluding the traditional ones) is:

- **Structural properties** are used to describe storage and access of geo data in technical infrastructures. E.g., inside the Nimbus project [1, 2] we used structural properties to perform a geometric lookup of geo objects that are stored in a distributed federation of servers. In contrast to topological properties, structural properties describe relations in terms, which are not directly perceived by end-users.

- **Organizational properties** are information about the generation process of geo data, e.g., unique identifiers to detect duplicate objects as a result of multiple imports. They answer questions such as: What is the data source, who modified an entry and how long is the entry expected to be valid? If the geometry is based on GPS measurements, how many satellites were available? Such data is important to achieve and preserve the quality of geo data. In contrast to structural properties, organizational properties describe non-technical characteristics.

- **Meta data** are names, images, sounds, or Web links. Even though structural and organizational properties are also meta data in a general meaning, this point only summarizes those properties that describe content perceivable by an end-user as meta data. Some meta data is intended only to be displayed to users, but others have to be included into a resolution process. E.g., the time validity property is used for geo objects that do not exist permanently such as fairs, weekly markets, or construction zones.

![Fig. 2. The depro authoring environment (left: Web-based; right: stand-alone)](image)

Inside the Hybris project, we use an XML language to represent geo objects, whereas each geo object is stored in an individual XML file. That simplifies the distribution of
geo objects to different servers. Fig. 2 shows the main authoring tool to edit geo objects. The different properties are reflected by different views on the data.

The editor provides a powerful plug-in interface to build future extensions. Currently plug-ins for refactoring, statistics, import/export, consistency checks, binary polygonal operations and for the creation of topological connections are implemented.

The plug-in interface significantly simplifies the creation of authoring functions as it already provides powerful functions such as geo coding of mouse positions and displaying maps often required for editing. In addition, the environment preserves consistency if plug-ins modify multiple domains simultaneously.

The major path to integrate huge amounts of geo data goes through the Import/Export interface. For important formats such as ATKIS (EDBS), GPS Logs (NMEA), Google Earth (KML), OpenStreetMaps (OSM) and TomTom POIs (OV2), import plug-ins are already implemented.

2.2 Reasoning Environment

MAP³ (Multi-Area Probability-based Positioning by Predicates) [5] introduces a new approach for position sensor data fusion (fig. 3). Any piece of information about the location at any point in time is mapped to a location predicate that forms a kind of universal interface to any positioning system. Predicates are mapped either to probability density representations or operations on densities.

![Fig. 3. The reasoning environment](image)

Position probability densities $f$ are mathematical constructions that need to be approximated for computation, if they do not follow some hard constrains such as Gaussian densities [4]. We use the following approximation $f_{\text{approx}}$ for a position $p$

$$f(p) \approx f_{\text{approx}}(p) = \sum_{i=1}^{n} w_i \Lambda(p, mph_i)$$

where $\Lambda$ is a characteristic function of a multipolygon with holes ($mph_i$) and $w_i$ the weight of the respective mph. An mph represents the most common approximating two-dimensional structure that is widely available and efficiently implemented in many tool environments, software libraries and spatial databases.

The Spatial Solver component executes density operations (e.g. multiplication, convolution) with the help of these mph operations. Depending on the application, the most...
probable location or a set of local maximum values is computed. For details of the probabilistic reasoning mechanism see [5].

Fig. 3. Hybris applications (left: Buddy Finder, right: Photo Mapper)

3 Conclusions

Managing geo data and locations currently still is a cost intensive task. With the Hybris environment, an application developer mainly can focus on the application-dependent part. We verified the approach with the help of two location-based applications developed as student project: a buddy finder for mobile phones and an application that assigns locations to digital pictures (fig. 3). A developer and can rely on a powerful tool environment to manage geo data, especially on the convenient authoring environment that provides a plug-in interface to integrate new import and processing mechanisms. The innovative spatial reasoning approach MAP$^3$ allows the easy integration of different positioning systems and provides an effective position fusion mechanism.

References

6. The Hybris Homepage: http://hybris.wireless-earth.org
Emotion and Personality in Driver Assistance Systems

Dirk M. Reichardt

BA Stuttgart - University of Cooperative Education
D-70180 Stuttgart, Germany
reichardt@ba-stuttgart.de

Abstract: Driver assistance systems are designed to support a human driver in the driving task. Since drivers are very different, the assistance system must adapt to the driver in order to provide the best support. The driver adaptive assistance systems which are currently designed provide adaptability which is generally done by the driver himself. Some already provide automatical adaptation. One of the key problems is intra-individual difference in behaviour. This is mostly considered as disturbance or covered by short-term models and quick adaptation. What causes intra-individual difference? Emotion is one of the factors in behaviour differences. This makes an emotional component in a driver model a considerable improvement potential for adaptive driver assistance systems. In our research, we investigate a driver model on the basis of the cognitive appraisal model by Orthony, Clore and Collins (OCC) combined with a personality model based on the "Big 5" personality factors. First results are produced in the curvature warning system scenario. The work is based on simulations in which the driver's emotional state is computed by the an emotional driver model. Current experiments provide plausible training data.

1 Introduction

Adaptive driver assistance systems monitor the driver's behaviour and determine the right time for information, a warning or even an intervention by the system. Often the driver's behaviour patterns are not clear and vary from day to day or even in shorter time spans. One of the factors which are responsible for this disturbance is the emotional state of the driver.

The aim of our work is to provide a simulation system, which produces training data for adaptive driver assistance systems. In this system, behaviour variations which are based on emotion are explicitly part of the simulation. The emotion can be fed into the system from an external source and it can be computed automatically as direct reaction to the current driving situation. In order to achieve this, we implemented a psychological model of emotion by Orthony, Clore and Collins (OCC) [1] and - as a further parameter for simulation - enhanced it by a personality component. This model is adapted and dedicated to the driving situation. In the following, driver assistance systems are briefly introduced. Furthermore, the basic information on the OCC model and personality is given before we outline the emotional driver model we work on.
1.1 Personalized Assistance Systems

The advantage and necessity of personalisation becomes obvious if the acceptance of assistance systems is regarded. It can be easily made plausible by looking at one of the scenarios we picked for the first tests of our model: the curve warning system.

The assistance systems warn a driver in case he or she may not have noticed an upcoming curve on a country road. As described in [2], the central parameter is the lateral acceleration in the curve. Given the average lateral acceleration in a curve and the average deceleration behaviour approaching curves, one can determine the time for a curve warning. Obviously this may be too late for the stereotype "grandmother" driver and much too early for a race driver like Michael Schumacher. Between those two extremes we find the drivers - and customers of the vehicle manufacturer. If inter-personal differences in driving style and warning time preference are large, an adaptation of the parameters to the current driver is necessary for the acceptance of the product. How does the adaptation take place? Initially an average driver model is assumed and the current driver is watched to determine the key parameters identifying his driving preferences. Using these parameters, a better fitting assistance becomes possible. What is the benefit of an additional emotion component in the simulation model? As mentioned before, emotion is one of the factors that influence intra-personal behaviour variations. This makes it possible to generate training data which are "disturbed" in a controlled and plausible way.

1.2 The OCC model of emotion

There are a few models of emotion and we decided for the OCC model due to earlier experience in another application context [3]. The architecture of the emotion component is based upon the cognitive appraisal model by Ortony, Clore and Collins. The OCC model has become a very popular model for emotion synthesis. It comprises 22 emotion categories which are grouped in three dimensions: goal relevant events, actions of agents and aspects of objects.

The model provides intensity variables which quantify the influence of eliciting events on the subsequent emotion generation. These intensities are derived from the scenario. Each of them is associated with an emotion class. Emotion classes are ‘fortunes of others’, ‘prospect-based emotions’, ‘well-being emotions’, ‘attribution emotions’ and ‘attraction emotions’. The main task of an application that intends to use the OCC model is to adapt the events, actions and objects, as well as the associated intensity values to the application scenario.

The OCC model is specified in a qualitative way. Therefore, some further estimations needed to be made in order to automatically produce quantitative results in terms of emotions. For our first tests we decided to implement a simplified linear model.
1.3 Personality

How does personality come to play a role in the simulation? This is another attempt to control variations in the simulation. The personality of a driver can be seen as a bias for the emotion generation in traffic situations. When generating test runs, different personalities of drivers are generated which leads to different emotional reactions.

The term personality describes the factors that determine the character of an individual. The personality is considered stable over time whereas emotions have a rather short "life-span". In between we have the concept of moods. There are a lot of different theories about personality. The most important and currently dominant theories are the trait theories. Best known are traits like extraversion, introversion or neuroticism as discussed by Eysenck. The "big 5" model of personality, which is the most popular model nowadays, consists of the traits neuroticism, extraversion, openness, agreeableness and conscientiousness. We decided to use this model in our simulation.

2 An Emotional Driver Model

In the first approach we implemented a "co-pilot" which computes the emotional state while the vehicle is driven through the simulation in a defined way. The three main aspects of the emotional driver model are: the adaptation of the OCC model to the driving task, the combination of OCC and personality and the integration of mood.

2.1 Adaptation of OCC

The OCC model uses intensity variables which can be used to adapt the model to situations. There are global and local intensity variables which need to be filled with situation relevant information. As an example for a global intensity variable, we can consider 'proximity'. If a relevant event is far in the future its effect on the emotional state is lower than in case it's very close. In our case, we consider the event of leaving the road accidentally in a curve due to inadequate speed. If the potential event is far in the future (due to the distance of the curve) the emotion of fear is less intense. Local intensity variables refer to a specific emotion class. Considering events, examples are 'likelihood' and 'desirability'. If an event is very likely and not at all desirable, it causes distress. If this event is in the future, the so called prospect based emotions are concerned: in this case it's fear. In our example, the event of leaving the road is of course undesirable. The likelihood of this event depends on the speed and curvature as well as on the subjective judgement of the situation. Here we also have a cognitive component of the driver model which needs to be connected. After the critical situation is passed, the evaluation mentioned above may lead to the emotion of 'relief'. The intensity of relief depends again on likelihood and desirability of the event. In the first approach and in the modeled curve warning situation, we did not consider emotional reactions to the actions of other agents yet.
2.2 Integration of personality

How can personality be integrated with the OCC model? We augmented the basic model with a factor for each of the personality traits. The following examples shall show how we approached this task:

The influence of ‘openness’ can be seen as follows: a person with a high rating in this scale may show more intense emotions even though the global intensity variable ‘sense of reality’ has a low value.

The influence of ‘conscientiousness’ can be seen as follows: a person with a high rating in this scale may have a clearer concept of likelihood and therefore this aspect gets a higher weight as compared to the intensity variable ‘arousal’.

The elicitation function of each emotion class contains the defined intensity variables. In our model, this function is enhanced by five further parameters: the personality traits. From these examples it becomes clear that the integration takes place on the general model level and does not rely on the traffic situation.

2.3 Integration of mood

We consider the mood of the driver as a filter for emotion elicitation. For example, the elicitation of joy is lower if the person is in a ‘bad’ mood. In order to model a mood, we consider the pleasure-arousal-dominance space as described by [4]. The process is as follows: the elicitation values are computed for all emotion classes, these values are transformed in PAD-space (we use the transformation described in [5]) and then the mood is updated by a weighted combination of the current emotions and the current mood. In a first approach, the emotions finally shown are generated by considering the mood as a stabilizing factor. If an emotion is a point in the 3-dimensional PAD-space and so is mood, then the emotion coordinates are determined by pulling the initially computed position towards the mood position. The respective force is determined by the personality.

3 Conclusion and Further Work

In first simulation runs, the emotional reactions seem plausible. Nevertheless, a formal verification of the model cannot be given. Future work will include the completion of the adaptation to the traffic scenario, the integration of further scenarios and the attempt to verify the emotional reactions which are observed. The final step is the closure of the loop: the emotional driver model shall be used to generate driving commands in the simulation.
References

OMIcar – A Demonstration Platform for VANET Protocols

Andreas Schmeiser, Yvonne Günter, Bernhard Wiegel, Matthias Rabel, and Hans Peter Großmann

Universität Ulm, Institut für Organisation und Management von Informationssystemen, 89069 Ulm, Germany
{andreas.schmeiser, yvonne.guenter, bernhard.wiegel, matthias.rabel, hans-peter.grossmann}@uni-ulm.de
http://www.uni-ulm.de/in/omi.html

Abstract. Vehicle-to-vehicle and vehicle-to-infrastructure communication are promising technologies for enhanced road safety and energy efficient transportation. Several groups are working on research and standardization of Vehicular Ad hoc NETwork (VANET) protocols. There are several protocols suggested that tackle specific issues of the VANET challenges. The performance of these protocols is evaluated by simulation. Mostly the VANET scenario is divided into the modelling of the vehicles’ movement and the simulation of data transmission. Especially the model of data transmission cannot consider all physical effects and therefore simplified approaches are used. This implies that solely simulation is not sufficient and a demonstration and test platform with autonomous vehicles is needed. In this paper we introduce OMIcar, a universal communication node which provides the required dynamics. The platform consists of a model car controlled by an ARM9 microcontroller based hardware. OMIcar is equipped with several sensors to move autonomously under laboratory conditions. A Software Defined Radio (SDR) is being developed as flexible communication system for testing different VANET protocols.

1 Motivation

With the ongoing increase of worldwide individual traffic, road safety and reduction of severe accidents are of significance. Therefore numerous efforts are undertaken to avoid crashes or reduce their effects e.g. by the use of driver assistance systems.

Further environment-friendly transportation requires energy-efficient vehicles. Therefore sophisticated systems that influence the traffic flow as well as advanced navigation systems are intended to reduce the waste of energy due to traffic congestion.

Modern cars already provide several driver assistance systems and therefore have numerous on-board sensors. But these sensors are only capable to provide the actual state of the car and knowledge about its close environment (Fig. 1).
Gathering an exact knowledge about the nearby environment would facilitate future assistance systems to anticipate dangerous situations very early. Thus the road safety can be improved.

The knowledge about the broad environment of a car contains information about the traffic situation. Consequently this influences the power management policy of the drive train and the routing decision of the navigation system for an energy-efficient driving. Hence vehicle-to-vehicle and vehicle-to-infrastructure communication need to be implemented in future to exchange sensor data between vehicles.

2 Vehicular Ad-hoc NETworks (VANETs)

Protocols for VANETs need to satisfy special requirements. A reliable data exchange is crucial for safety applications like the early anticipation of dangers. The transmission of data must be still accomplishable while having high network dynamics and a varying density of nodes. Quality of Service support is fundamental to prioritize safety-related messages over informational content.

A common approach for VANETs is the IEEE 802.11p standard [1] which defines the physical and Medium Access Control (MAC) layer. The channel is accessed using legacy IEEE 802.11 MAC [2].

As the information exchange between vehicles is mainly based on broadcasting messages IEEE 802.11 faces severe data loss. The main cause is that Request To Send/Clear To Send (RTS/CTS) mechanism is not feasible and thus the hidden station problem is not solved.

Therefore other approaches for MAC algorithms are under development, e.g. our approach focuses on a Cluster-based Medium Access Control (CBMAC) [3]. Simulation tools are used to evaluate and compare the proposed protocols.

3 From Simulation to Demonstration

Simulation is always a trade-off between accuracy and calculation time on current computers. That's the reason why VANET simulation tools cannot consider all physical effects and therefore use simplified approaches, especially for the model of data transmission.
Figure 2 shows on the left-hand side the implemented layers of our VANET simulation platform OMIsim [4]. This tool combines a microscopic traffic model and a simulation environment for different communication protocols. The traffic model includes a road network based on map data, the generation and routing of vehicles and the vehicles’ behaviour like speed, turning at intersections and overtaking. The communication environment consists of an estimation of the signal propagation and of protocols for medium access control and location based routing of messages.

A reasonable demonstration platform requires the transfer of the modelled vehicles’ movement into reality. The implementation of an applicable physical layer fit the reality better than the simulation models anyway. A hardware-efficient synthesis of the MAC algorithms might show a different behaviour under these conditions.

The higher layers of the communication protocol are software based modules that are independent of physical effects. So they are kept the same for ease of compatibility between simulation and demonstration. Consequently the right-hand side of Figure 2 shows our further concept.

4 The OMIcar Demonstration Platform

A fleet of common cars in usual road traffic scenarios would result in a most real test platform without any doubt. But due to tremendous costs and legal issues such an approach is not feasible. A further drawback is that the comparability of repeated measurements cannot be guaranteed under the same conditions.

So we introduce OMIcar, an autonomous model car that serves as universal communication node for VANET protocols (Fig. 3).
4.1 ARM9 Microcontroller

A powerful embedded system has been developed at our Institute that acts as main controller of the autonomous vehicle. This hardware provides a large number of standard interfaces for seamless connection of peripherals: Ethernet, Universal Serial Bus (USB), Compact Flash, synchronous and asynchronous serial interfaces, Digital Input/Output, Pulse Width Modulation (PWM).

Embedded Linux is used as operating system because of the huge number of drivers available and the option for development of applications in C or C++. This allows the possibility of porting software components between our simulation and demonstration platforms.

4.2 Autonomous Driving

OMIcar is equipped with the following sensors to enable autonomous movements under laboratory conditions.

Optical motion sensors, based on an optical computer mouse chip, are used for tracking movements and measuring distances over ground. A Radio Frequency IDentification (RFID) reader is used to localize junctions, intersections, and roundabouts that are marked by RFID tags. An ultrasonic sensor detects obstacles and measures the distance to these objects in front of the car. The permanent tracking of the orientation of the car is performed by an electronic compass measuring the earth’s magnetic field. A tri-axial accelerometer and a gyro allow inertial guidance and provide another source for tracking movements.
In a further step sensor fusion techniques ought to enhance accuracy, add redundancy, and help in validation of readings.

Running a scenario requires the following simple steps: applying the RFID tags on the floor, downloading the corresponding map data to the controller board and transferring waypoints to the car. Afterwards the ARM9 application takes over and the routing of the vehicle as well as control of motor and steering are performed autonomously.

4.3 Software Defined Radio (SDR)

The demonstration platform is intended to do flexible tests of different VANET communication systems. Hence a reconfigurable hardware platform is used to develop a SDR.

The lower layers of the communication protocols are implemented using an Altera Stratix II – Field Programmable Gate Array (FPGA) Development Kit [5]. Currently special effort focuses on Orthogonal Frequency Division Multiplex (OFDM) modulation for the physical layer, mechanisms for synchronization and channel coding.

5 Summary

In this paper we presented our demonstration platform OMIcar, a model car with a sensor cluster for autonomous driving and equipped with a SDR. A special feature of OMIcar is that no complex infrastructure is needed thus allowing an easy change of scenarios under laboratory conditions. Thanks to the SDR the same hardware can be used for different implementations of VANET protocols. Our design facilitates an easy transfer of communication algorithms from the simulation tool into the demo platform.

References

1. ASTM E2213-03: Standard specification for telecommunications and information exchange between roadside and vehicle systems – 5GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY), 2003
4. Y. Hörmann, H. P. Großmann, et al.: Simulator for inter-vehicle communication based on traffic modeling. IEEE Intelligent Vehicles Symposium, Univ. of Parma, Parma, Italy, June 14 - 17, 2004
The RAW Image File Format Problem - Applications of Digital SLR Cameras in Astronomy and Science

Thilo Bauer

bluewater multimedia concepts e.K., Kölnstraße 191, D-53757 St. Augustin, Germany
tbauer@astro.bwmc.net

Abstract. This work describes the application of digital single lens reflex (SLR) cameras in astronomy science. A definition of the term raw file format is given. As with formats like FITS or TIFF, the CR2 file format is explained being a disclosed, and well documented raw file format. Results from raw image file decomposition with different software tools, and a comparison of imaging with a digital SLR camera and a professional grade CCD detector are presented. The analysis of noise and image histograms will help to recover problems with raw file format decomposition tools and their adjustment. Without doubt, digital SLR cameras are suited for special scientific applications. Further implications from this work are a detailed analysis of noise characteristics to achieve high quality imaging at very low light levels from raw files taken with digital SLR cameras.

1 Introduction

Scientific imaging at low light levels require imaging sensors with a high linearity and low noise characteristics. The most common applications within the optical astronomy are stellar photometry, the measurement of light intensities, and astrometry, the measurement of positions of stars, planets, quasi stellar objects or complex star formations. Similar requirements of measuring intensities and positions at low light level will be found in optical microscopy, biology, pathology, physics and other sciences. Popular imaging sensors are divided into three sensor technologies: monochrome CCD sensors, CMOS cameras, partially using color filter arrays to distinguish different wavelengths, and a technology often referred as the Foveon technology [1]. The latter two technologies form the basic sensors of modern digital SLR color cameras. The Canon EOS digital camera models are based on proprietary CMOS sensors. A Bayer color filter array in front of the silicon chip separates different colors into red, green and blue pixel values (RGB).

A raw file format herewith shall be defined as containing a two dimensional set of digital numbers obtained from a given imaging sensor, obtained lossless, without any preprocessing, and having a well defined value range and bit representation as yielded by the digitization unit. Scientific analysis furthermore requires linear response or at least a detector response which allows conclusions about the true nature of the original signal.

The FITS file format is the de-facto standard within astronomy science across all physical wavelengths [3]. FITS is provided and supported by the International Astronomical Union (IAU) and its FITS Working Group (IAU-FWG) as the international
control authority [4,5]. One of the early adopted image and graphics standard file formats for general use is the Tagged Image File Format (TIFF) created by Adobe [6]. Similar to the FITS format, TIFF supports different pixel bit counts: 1 to 8, and 16 bit of linear dynamic range with binary, grey scale or false color images (color tables representing different gray values), and true color RGB content. At least three compression methods are provided with this specification. Manufacturers of commercial, digital SLR cameras tend to provide different approaches for compact and reliable file formats to the photographer, which presently leads to a zoo of file formats with every manufacturer. Proprietary raw file formats often are argued as being undisclosed, thus having harmful disadvantages with processing and long-term archiving [7]. Work is done for replacements, like the Adobe DNG specification [8]. Undocumented in the sense of having a format with details undisclosed to the public seems especially true for cameras like the Canon digital EOS cameras.

2 Imaging with Astronomical Telescopes

NGC 7635 is a catalogue name of a H-II region of interstellar gas, often referred as the Bubble Nebula [9], and invisible for the naked eye. The nebula was chosen as a test candidate of a very faint light source. The author took images with a Canon EOS 400D camera mounted on a Vixen VC200L Cassegrain 20cm telescope on Aug 4, 2007. Conditions were a few days after full bright moon taken from the authors garden and with typical light pollution from the moon and city light illumination. To achieve a higher dynamic range 30 single exposures were recorded at a temperature of +14°C. The camera provides an angular resolution of about 1.3 arcsec/pix at 3888x2588 pixel in the primary focus of the telescope. Having 12 bit dynamic range, and not to clip the intensities of bright stars, the camera was set to 30 seconds of exposure time at a gain representing 800 ASA. Internal pre-processing, noise reduction, color and gamma correction were switch off. The images were processed with modifications of a stacking method developed by the author [10,11]: automatic detection of field stars above 10 noise sigma, detection of the stellar positions, selection of images without position movement of the stars (due to turbulent atmosphere and guiding errors during exposure), and shift & add the selected individual images recorded. The resulting image yielded a recentered composite of 27 selected frames. As a comparison with the equipment explained above, a single image of NGC 7635 taken by the author with the 1m Cassegrain telescope at the Hoher List Observatory is presented. The image was recorded on Nov 3, 1993. Exposure time was 600 seconds with an interference filter blocking background light except the red hydrogen emission from the nebula. A scientific CCD imaging system from Astrophed Inc., UK at 416x576 pixel resolution was used within this period providing a similar angular resolution per pixel with the focal reducer of the telescope. With cooling down to about -70°C, using slow scan mode, the camera achieved 16 bit dynamic range. Unfortunately this CCD imaging device was destroyed in the meanwhile due to an irreparable hardware defect.
3 First Results

A novel image processing software written in Java with native C libraries for fast image processing is presented with this work. Within the tool-chain the author used these tools for image decomposition and further image processing: the original software provided from the camera manufacturer[12], and DCRAW, Revision: 1.397, provided by Dave Coffin[13].

A quick reengineering of the Canon raw files showed, that the metadata extracted from the raw files represent a well formed TIFF structure with EXIF extension [14]. The TIFF compression tag identifies a JPEG compression method provided with the TIFF standard [6]. As JPEG is generally believed as an image format with lossy compression, the original JPEG specification also defines lossless compression methods [15,16]. Unfortunately, not every image processing software supports all possible TIFF variants and derivatives, like EXIF. Moreover, and to make confusion perfect, Canon created a different file extension for the EXIF raw image files: CR2, officially named „Canon Raw, Version 2“ , which is a TIFF derivate.

A code review of DCRAW with its many options confirms the use of several pre-processing techniques used for image decomposition. Compared to conventional photography, standard astronomical interpretation of colors, like stellar color photometry, relies on different rules and generally will avoid non-linearities, color and noise correction. The developer of DCRAW advises to use certain options only for obtaining linear and unprocessed image data [17]. Option „-D“ produces linear grayscale output representing the isolated R, G (2x) and B pixel values to be rearranged by the application software. The options „-h -k 0 -r 1.0 1.0 1.0 1.0 -o 0 -4 -T“ provides linear 16 bit TIFF output no additional processing at half of the pixel resolution of the sensor and yielding virtual RGB pixels. This is confirmed by the image analysis. Histograms show uniformly distributed values as it would be expected from distributing 12 bit values into 16 bit dynamic resolution from a multiplication by 16. Decomposition with the latter options will result in a color shift of about +/- 0.5 pixel in R and B colors at an angle of -45° due to the geometry of the Bayer matrix which is arranged as a square RGGB pixel pattern.

With the original Canon software the raw files have also been temporarily decomposed and converted into TIFF (16 bit, RGB). No options exist to control image decomposition and conversion into more appropriate formats like standard TIFF. From reading these TIFF files, unequal distributed digital numbers, noticeable asymmetric, non-gaussian, and clipped histogram profiles have been found. The noise of decomposed dark frames show broadened histograms with about 5-10 times the noise at FWHM compared to the results of DCRAW from the same raw files. This indicates a complex, non-linear but not very successful pre-processing stage like non-linear gamma scaling and intensity clipping at the lower and bright end of the image intensities. This is a very surprising result, as even a conventional photographers would expect noise properties being at least as small as the camera will it provide raw and digital. Especially, if the user would expect a further image optimization stage from such a software tool.
4 Noise Properties, Sensitivity and the Effects of Correlated Noise Terms

Noise was estimated from histograms of single dark exposures. From a sample of the frames about 13 DU (digital units) at FWHM for the EOS 400D (30 seconds exposure, 12 bit, 4096 total units), and about 30 DU for the Astromed camera (600 seconds exposure, 16 bit, 65536 total units) were found respectively. As a result of a relatively larger read-out noise at less dynamic range, sky illumination (photon noise), and reduced camera sensitivity in the red light (~20-25% relative quantum efficiency at hydrogen emission), the faint red nebula degraded in contrast with the EOS 400D compared to the result with the scientific camera. The pixel dimension of the EOS camera is smaller compared to the scientific camera with in gaps between different color pixels, and reduced full-well capacity. However, both images show stars of nearly the same limiting stellar magnitude (faintest stars detected) with both camera and telescope setups. The limiting stellar magnitude is not a result of the aperture diameter of the telescopes, but related to the pixel dimension, illumination intensity, optical spread of point light sources, signal-to-noise ratio, and finally the focal length of the optics [18]. The ratio between the focal length and pixel dimension was nearly identical within this seemingly unequal comparison of two astronomical telescopes having very different aperture diameter.

Different noise models exist within astronomy depending on detectors and algorithms. However, a standard ISO procedure is defined to determine noise characteristics of digital SLR cameras and comparable imaging detectors [19,20]. First results from the astronomical observations indicate, that there remain additional and correlated noise terms close to and below read-out and fixed pattern noise level. Asymmetric (multiple gaussian) noise profiles from histograms of the EOS 400D are obviously caused by line bias variations within individual frames (temporal noise). As a preliminary result of this work, these terms can be corrected resulting in a slightly better S/N. Conversion of analog image intensities into digital numbers may introduce effects like deferred charge, hot sensor areas, bad pixels, periodic terms and so on [21]. Regularly distributed peaks at powers of two (e.g. distances of 2, 4, 8,...) indicate non-linear bit conversion errors from the histograms. This effect was observed from the files of the scientific camera, but invisible with raw files of the EOS 400D. Vertical, non-temporal bias variations remain after correction of line bias variations. Assuming constant pixel dimension and response is not applicable to real imaging detectors. The local light intensity response may vary over the whole pixel area within subpixel range. This is true for both devices, the digital SLR camera and the scientific camera. A Canon EOS 40D (received 26 Jan., 2008) digital SLR camera offers 14 bit of dynamic range from a new signal processing unit. With the same settings (t=30 sec, 800 ASA), dark exposures were taken at room temperature (T=22°C). Histograms again yield asymmetric profiles from dark frames having temporal line bias variation. However, a smaller total noise distribution of 27 DU of noise at FWHM was found. Thus, gain in S/N with this camera is about a factor of two compared
to the EOS 400D model. Set to 400 ASA, the camera presents 17 DU noise at FWHM (T=1°C). Correction of line bias again yields a slightly better S/N. Gaussian distributed noise profiles from the compensated raw dark frames now tend to a S/N similar to that of the scientific camera. A detailed analysis of noise properties and a research for compensation of correlated noise terms is ongoing.

4 Critical Evaluation and Remarks

Concerns about disadvantages of camera raw file formats appear unsubstantiated under certain conditions, like those described within this paper. Some digital SLR cameras provide long-exposure imaging representing unprocessed sensor characteristics with disclosed raw file formats. This may imply some advantages like simplified use of cameras without the need for dedicated image recording hardware with some astronomical applications. Scientists would prefer high-end camera models operating at 14 bit or more dynamic range. A noticable remark is a surprisingly, more noisy result and additional non-linearities obtained from image decomposition with original software from camera manufacturers. This, however, is intolerable even for conventional photographers expecting high quality imaging at low light levels. Thus, the term raw image file format in no way will automatically be a guarantee, or a synonym for high quality imaging under any circumstance, as it possibly would be expected by the user. Instead, results of image decomposition may depend on the software tools and adjustments. Scientists are strongly advised to consider their hardware requirements, tools and techniques for the decomposition of the raw image files and before applying further image processing and analysis. Implications from this work are detailed measurements of noise properties, and algorithmic development to achieve high accuracy with astronomical observations. The algorithm used within this work to compensate temporal line bias noise seems robust, but cumbersome.

References

1. H. T. Hytti, Characterization of digital image noise properties based on RAW data, SPIE 2005
16. E. Hamilton, JPEG File Interchange Format Version 1.02, September 1, C-Cube Microsystems, 1992
17. Dave Coffin, Jan 4, 2007, private email communication
18. E. H. Geyer, Bemerkungen zur Nachweigsgrenze astronomischer Objekte bei erdbundenen Beobachtungen, DFG Rundgespräch und Workshop, Bad Honnef Nov. 1984, (manuskript received by private communication)
Subtyping in Java 5.0

Martin Plümicke
University of Cooperative Education Stuttgart/Horb
m.pluemicke@ba-horb.de

1 Introduction

With the introduction of Java 5.0 [1] the type system has been extended by parameterized types, type variables, type terms, and wildcards. As a result very complex types can arise. For example the term

Vectors<? extends Vector<AbstractList<Integer>>>

is a correct type in Java 5.0.

Considering that, it is often rather difficult for a programmer to recognize whether such a complex type is the correct one for a given method or not.

This has caused us to develop a Java 5.0 type inference system which assists the programmer by calculating types automatically [2]. This type inference system allows us, to declare method parameters and local variables without type annotations. The type inference algorithm calculates the appropriate and principle types.

In [1] the Java 5.0 type system is specified. This specification is done in a semi-formal way. The presentation is sometimes less clearly arranged. In this paper we present a summary of an integrated framework for the Java 5.0 type system.

2 Java 5.0 Simple Types

The base of the types are elements of the set of terms $T_{\theta}(TV)$, which are given as a set of terms over a finite rank alphabet $\Theta = \Theta_{n\in N}$ of class names and a set of type variables $TV$, where the index $n$ represents the number of type parameters. Therefore we denote them as type terms instead of types.

Example 1. Let the following Java 5.0 program be given:

```java
class A<a> implements I<a> { ... }
class B<a> extends A<a> { ... }
class C<k extends I<b>,b> { ... }
interface I<a> { ... }
interface J<k> { ... }
```

The rank alphabet $\Theta = \Theta_{n\in N}$ is determined by $\Theta^{(1)} = \{ A, B, I, J \}$ and $\Theta^{(2)} = \{ C \}$. For example $A<$Integer$>, A<$Boolean$>$, and $C<$Object$>, Object$ are type terms.
As the type terms are constructed over the class names, we call the class names in this framework type constructors.

If we consider the Java 5.0 program of Example 1 more accurately, we recognize that the bound of the type parameters b in the class C is not considered. This leads to the problem that type terms like C<C, b>, a> are in the term set $T_{\Theta}(IV)$, although they are not correct in Java 5.0.

The solution of the problem is the extension of the rank alphabet $\Theta$ to a type signature, where the arity of the type constructors is indexed by bounded type variables. This leads to a restriction in the type term construction, such that the correct set of type terms is a subset of $T_{\Theta}(IV)$. Additionally the set of correct type terms is added by some wildcard constructions. We call the set of correct types set of simple types $\text{SType}_{TS}(BTV)$ (Def. 3).

**Definition 1 (Bounded type variables).** Let $\text{SType}_{TS}(BTV)$ be a set of simple types. Then, the set of bounded type variables is an indexed set $BTV = (BTV^{(\Theta)})_{\Theta \in \text{SType}_{TS}(BTV)}$, where each type variable is assigned to an intersection of simple types.

$\text{SType}_{TS}(BTV)$ denotes the set of intersections over simple types. In the following we will write a type variable $a$ bounded by the type $t$ as $a_t$.

Type variables which are not bounded can be considered as bounded type variables by Object.

**Example 2.** Let the following Java 5.0 class be given.

```java
class BoundedTypeVars<a extends Number, b extends Number> {
    <t extends Vector<Integer> & J<a, b> & I, 
    r extends Number> void m (...) { ... }
}
```

The set of bounded type variable $BTV$ of the method $m$ is given as $BTV^{(\text{Number})} = \{ a, r \}$ and $BTV^{(\text{Vector<Integer>} & \text{J} <a, b, I>)} = \{ t \}$.

**Definition 2 (Type signature, type constructor).** Let $\text{SType}_{TS}(BTV)$ be a set of simple types. A type signature $TS$ is a pair $(\text{SType}_{TS}(BTV), TC)$ where $BTV$ is an indexed set of bounded type variables and $TC$ is a $(BTV^*)$-indexed set of type constructors (class names).

**Example 3.** Let the Java 5.0 program from Example 1 be given again. Then, the corresponding indexed set of type constructors is given as $TC^{(a_{\text{Number}})} = \{ A, B, I, J \}$, $TC^{(b_{\text{Object}})} = \{ C \}$, and $TC^{(\text{J} <a, b, \text{Object}>)} = \{ D \}$.

For the following definitions, we need the concept of capture conversion ([1], §5.1.10). The capture conversion of $C<\theta_1, \ldots, \theta_n>$ is denoted by $\text{CC}(C<\theta_1, \ldots, \theta_n>)$.

The following definition of the set of simple types is connected to the corresponding definition of parameterized types in [1], §4.5.

**Definition 3 (Simple types).** The set of simple types $\text{SType}_{TS}(BTV)$ for a given type signature $\text{SType}_{TS}(BTV)$, $TC$ is defined as the smallest set satisfying the following conditions:
35

- For each intersection type $\text{}$ $\textbf{ty}$: $BTV(ty) \subseteq \text{SType}_{TS}(BTV)$
- $TC(\alpha) \subseteq \text{SType}_{TS}(BTV)$
- For $ty_i \in \text{SType}_{TS}(BTV)$ union
  \{ $\Box$ \} $\cup \{$ $?$ extends $\tau$ | $\tau \in \text{SType}_{TS}(BTV)$ \}$
  $\cup \{$ $?$ super $\tau$ | $\tau \in \text{SType}_{TS}(BTV)$ \}$
  and $C \in TC(\alpha_1, \alpha_2, \ldots, \alpha_n)$ holds
  $C\langle ty_1, \ldots, ty_n \rangle \in \text{SType}_{TS}(BTV)$ if after $C\langle ty_1, \ldots, ty_n \rangle$ subjected to
  the capture conversion resulting in the type $C\langle \overline{ty_1}, \ldots, \overline{ty_n} \rangle$ for each actual
  type argument $\overline{ty_i}$ holds: $\overline{ty_i} \subseteq^* b_j[a_j \mapsto \overline{ty_j} | 1 \leq j \leq n]$, where $\subseteq^*$ is a
  subtyping ordering (Def. 4).

Example 4. Let the Java 5.0 program from Example 1 and the corresponding indexed set of type constructors $TC$ from Example 3 be given again. Let additional $\text{Integer, Boolean} \in TC(\alpha)$. The terms $\text{A<Integer>}$ are $\text{A<B<Boolean>>}$ are simple types. $C\langle \text{A<Object>, Object} \rangle$ is also is a simple type as $\text{A<Object> \subseteq^* I<Object>}$.
In contrast $C\langle \text{C<a,b>, a} \rangle$ is no simple type, as $C\langle \text{C<a,b>, a} \rangle \not\subseteq^* I\langle\text{a}\rangle$.

3 Subtyping in Java 5.0

In the following we derive the “subtyping relation” (cp. [1], §4.10) as the reflexive, transitive, and instantiating closure of the extends/implements relation given by the Java 5.0 program.
We will use $\theta$ as an abbreviation for the type term “$?$ extends $\theta$” and $\overline{\theta}$ as an abbreviation for the type term “$?$ super $\theta$”.

Definition 4 (Subtyping relation $\subseteq^*$ on $\text{SType}_{TS}(BTV)$). Let $TS = (\text{SType}_{TS}(BTV), TC)$ be a type signature of a given Java 5.0 program. The subtyping relation $\subseteq^*$ is given as the reflexive and transitive closure of the smallest relation satisfying the following conditions:

- if $\theta$ extends/implements $\theta'$ then $\theta \subseteq^* \theta'$.
- if $\theta_1 \subseteq \theta_2$ then $\sigma_1(\theta_1) \subseteq \sigma_2(\theta_1)$ for all substitutions $\sigma_1, \sigma_2 : BTV \rightarrow \text{SType}_{TS}(BTV)$, where for each type variable $a$ of $\theta_2$ holds $\sigma_1(a) = \sigma_2(a)$ (soundness condition).
- $\alpha \subseteq^* \theta_i$ for $\alpha \in BTV(\theta_{i,0}, \ldots, \theta_{n,0})$ and $1 \leq i \leq n$.
- It holds $C\langle \theta_1, \ldots, \theta_n \rangle \subseteq^* C\langle \theta'_1, \ldots, \theta'_n \rangle$ if for each $\theta_i$ and $\theta'_i$, respectively, one of the following conditions is valid:
  \begin{itemize}
  \item $\theta_i = \theta'_i$, $\theta'_i = \tau \theta'_i$ and $\theta_i \subseteq^* \theta'_i$.
  \item $\theta_i = \tau \theta_i$, $\theta'_i = \tau \theta'_i$ and $\theta_i \subseteq^* \theta'_i$.
  \item $\theta_i, \theta'_i \in \text{SType}_{TS}(BTV)$ and $\theta_i = \theta'_i$.
  \item $\theta_i = \tau \theta_i$
  \item $\theta'_i = \tau \theta'_i$
  \end{itemize}

1 For non wildcard type arguments the capture conversion $\overline{ty_i}$ equals $ty_i$. 
Corollary 1. The subtyping relation is an ordering.

Example 5. Let the Java 5.0 program from Example 1 be given again. Then the following relationships hold:

- A<a> ≤<⇒> I<a>, as A<a> implements I<a>
- A<Integer> ≤<⇒> I<Integer>, where σ₁ = [a → Integer] = σ₂
- A<Integer> ≤<⇒> I<Object extends Object>, as Integer ≤<⇒> Object
- A<Object> ≤<⇒> I<Object>, as Integer ≤<⇒> Object

4 Soundness of the Java 5.0 type system

Let us consider again the definition of the subtyping relation (Def. 4). It is surprising that the soundness condition for σ₁ and σ₂ in the second item is not σ₁(a) ≤<⇒> σ₂(a), but σ₁(a) = σ₂(a). This is necessary to get a sound type system. This property is the reason for the introduction of wildcards in Java 5.0 (cp. [1], §5.1.10). Let the following Java 5.0 classes be given.

```java
class Super { ... }
class Sub extends Super { ... }
class Application {
    public static void main(String[] args) {
        Vector<Super> v = new Vector<Sub>(); // not really correct
        v.addElement(new Super());
    }
}
```

An element of the type Vector<Sub> is assigned to the variable v of the type Vector<Super>. This is no problem, as all elements which have the type Sub have also the type Super. Then a new element of the type Super is added to the vector which is assigned to the variable v. Now we have the problem, that elements of this vector have the type Sub and Super is no subtype of Sub. If this would be type correct, the type system would be unsound.

Therefore expression assignments like this are prohibited. The restriction demands that the declaration must be Vector<Super> v = new Vector<Super>(); But, sometimes assignments like Vector<Super> v = new Vector<Sub>(); would although be preferable. Therefore wildcards are introduced. For example the declaration Vector<? extends Super> v = new Vector<Sub> (); is allowed. Now Vector<Sub> is a subtype of Vector<? extends Super>, which means the assignment is type correct. In this case v.addElement(new Super()); is prohibited as Super is no subtype of "<? extends Super". This means that the unsoundness problem is also solved.

On the other hand, if an element of a subclass should be added to a vector of its superclass, the parameter of the vector must have a lower bound:
Vector<? super Super> v2 = new Vector<Super> ();  
v2.addElement(new Sub());

In this case only vectors with a parameter of a supertype of Super can be assigned to v2. This means that no unsoundness arises.

We have used wildcard types like “? extends Super”, although there are no simple types. Therefore we have to extend the set of simple type.

**Definition 5 (Extended simple types).** Let SType\( _{TS}(\text{BTV}) \) be a set of simple types. The corresponding set of extended simple types is given as

\[
\text{ExtSType}_{TS}(\text{BTV}) = \text{STypes}(\text{BTV}) \\
\cup \{ ? \} \cup \{ ? \text{extends } \theta \mid \theta \in \text{SType}_{TS}(\text{BTV}) \} \\
\cup \{ ? \text{super } \theta \mid \theta \in \text{SType}_{TS}(\text{BTV}) \}.
\]

Wildcard types cannot be used explicitly in Java 5.0 programs. But they are allowed as instances of type variables, which means that types like this occur implicitly during the type check of Java 5.0 programs.

Finally, we have to extend the subtyping relation to wildcard types.

**Definition 6 (Subtyping relation \( \leq^* \) on ExtSType\( _{TS}(\text{BTV}) \)).** Let \( \leq^* \) be a subtyping relation on a given set of simple types SType\( _{TS}(\text{BTV}) \). Then \( \leq^* \) is continued on the corresponding set of extended simple types ExtSType\( _{TS}(\text{BTV}) \) by: For \( \theta \leq^* \theta' \) holds: \( \theta \leq^* \gamma \theta' \), \( \gamma \theta \leq^* \theta' \), and \( \gamma \theta \leq^* \gamma \theta' \).

5 Conclusion and Outlook

In this paper we presented a formalization of the Java 5.0 type system. We defined the set of Java 5.0 simple types as type terms, which are explicitly allowed in Java 5.0 programs. We extended this set by wildcard types, which appear implicitly during the type checking. We defined a subtyping ordering at first on the set of Java 5.0 simple types and extended it to wildcard types. Additionally, we considered the soundness of the Java 5.0 type system. We showed, how the Java 5.0 type system becomes quite flexible by introducing wildcards without losing the soundness.

The Java 5.0 type system is the base of the type inference algorithm [2]. We have implemented a prototype of this type inference algorithm.

References

Brain Imaging: Measurement of Oxidative Processes with Magnetic Resonance

Hannes Wiesner
Max-Planck Institut für biologische Kybernetik, Spemannstraße 38, 72076 Tübingen, Germany
hannes.wiesner@tuebingen.mpg.de

Abstract. With information technology our view of the brain is simplified, with neurons, like transistors on a chip, integrated in a massively parallel computer. Now with functional magnetic resonance imaging (fMRI) the analysis of human brain function at high spatial and temporal resolution is established. Thus, digital information processing is involved at many steps from the transformation of electromagnetic signals up to statistical analysis and simulation. To study the dynamics of neural information processing, one can measure biochemical energy consumption, apart from the electrical activity of neurons. Simply put, neural information processing in the brain is fueled by two major energy substrates: sugar (i.e. glucose) and oxygen. However, the consumption of these substrates is only indirectly measured in fMRI originating from linked changes of both cerebral blood flow (CBF) and oxygen consumption (CMRO2). Although the blood oxygenation level dependent (BOLD) contrast in fMRI is an indirect measure, it can be calibrated. Therefore, it would be of great advantage if indirect changes in underlying energy consumption could be transformed into direct medical indicators before pathological structures, like the growth of a tumor, become visible. In contrast to glucose the current results for oxygen consumption lack clear consistency. However, by the direct detection of 17-O in magnetic resonance (MR), a novel method for 3-dimensional imaging of oxidative metabolism has been established recently. The technology still requires very strong magnetic fields, but specificity and reproducibility can be extended substantially in comparison to previous methodological limitations. As a result of the quantification of oxygen consumption, it is be expected, that a link between the indirect fMRI signal and energy metabolism from neural activity can be found.
Localization of Autonomous Robots with On-Board Sensors

Reinhard Lafrenz

IPVS, University of Stuttgart, Universitätstraße 38, 70569 Stuttgart, Germany
lafrenz@ipvs.uni-stuttgart.de

Abstract. One of the most fundamental skills in mobile robotics is to self-localization, i.e. to give an answer to the question "Where am I?" Especially in indoor environments, there is usually no positioning information such as GPS available. In this case, a robot needs to detect and interpret environmental features with its own sensors and relate them to a (known) map. For some algorithms need an initial position estimation, two problems have to be solved. First, how to obtain the initial position, and secondly, what if the robot is "kidnapped" and brought to a new position without knowing about the transport and thus having a wrong position hypothesis. In this talk, some algorithms for individual robots and for groups of robots, which have proven to be successful in the RoboCup (soccer playing robots) context, are presented.
Soccerbot Demonstration

Claus Engelhardt, Tristan Zur, Dominik kleine Kalvelage, Lars Sonnabend, and Dirk M. Reichardt

Information Technology, BA Stuttgart, Rotebühlplatz 41, 70178 Stuttgart, Germany
reichardt@ba-stuttgart.de

Abstract. A workshop for projects on robotics was founded in 2007 in the studying course of information technology and applied computer science at BA Stuttgart. In this, the students are working aside their regular teaching plan on long-term projects like soccer-playing robots at different elaboration stages. Currently, experiments and investigations are conducted with the Sony Aibo Robodog (Fig. 2) and the Qfix Soccerbot (Fig. 1). For these developments, sensoring, control, and artificial intelligence have to be combined effectively. The current working state was demonstrated at ITCS2008 event.
Author Index

Bauer, T. ....................................................................................................... 27
Engelhardt, C. ........................................................................................... 40
Großmann, H. P. ....................................................................................... 22
Günter, Y. .................................................................................................. 22
Jenke, P. ...................................................................................................... 1
Kleine Kalvelage, D.................................................................................. 40
Kochanowski, M.......................................................................................... 1
Lafrenz, R. ............................................................................................ 39
Plümicke, M............................................................................................... 33
Rabel, M.................................................................................................... 22
Reichardt, D. .......................................................................................... 17, 40
Roth, J. ...................................................................................................... 12
Schmeiser, A............................................................................................... 22
Sikora, A. .................................................................................................. 7
Sonntabend, L. ......................................................................................... 40
Straßer, W................................................................................................... 1
Wiegel, B. .................................................................................................. 22
Wiesner, H. ............................................................................................... 38
Zur, T. ...................................................................................................... 40