Proceedings of
Umeå’s 20th Student Conference in Computing Science
USCCS 2016
S. Bensch, T. Hellström (editors)

UMINF 16.2
ISSN-0348-0542

Department of Computing Science
Umeå University
Preface

The Umeå Student Conference in Computing Science (USCCS) is organized annually as part of a course given by the Computing Science department at Umeå University. The objective of the course is to give the students a practical introduction to independent research, scientific writing, and oral presentation.

A student who participates in the course first selects a topic and a research question that he or she is interested in. If the topic is accepted, the student outlines a paper and composes an annotated bibliography to give a survey of the research topic. The main work consists of conducting the actual research that answers the question asked, and convincingly and clearly reporting the results in a scientific paper. Another major part of the course is multiple internal peer review meetings in which groups of students read each others’ papers and give feedback to the author. This process gives valuable training in both giving and receiving criticism in a constructive manner. Altogether, the students learn to formulate and develop their own ideas in a scientific manner, in a process involving internal peer reviewing of each other’s work, and incremental development and refinement of a scientific paper.

Each scientific paper is submitted to USCCS through an on-line submission system, and receives two or more reviews written by members of the Computing Science department. Based on the reviews, the editors of the conference proceedings (the teachers of the course) issue a decision of preliminary acceptance of the paper to each author. If, after final revision, a paper is accepted, the student is given the opportunity to present the work at the conference. The review process and the conference format aims at mimicking realistic settings for publishing and participation at scientific conferences.

USCCS is the highlight of the course, and this year the conference received thirteen submissions (out of a possible fifteen), which were carefully reviewed by the reviewers listed on the following page.

We are very grateful to the reviewers who did an excellent job despite the very tight time frame and busy schedule. As a result of the reviewing process, eleven submissions were accepted for presentation at the conference. We would like to thank and congratulate all authors for their hard work and excellent final results that are presented during the conference.

We wish all participants of USCCS interesting exchange of ideas and stimulating discussions during the conference.

Umeå, 5 January 2016

Suna Bensch
Thomas Hellström
Organizing Committee
Suna Bensch
Thomas Hellström

With special thanks to the reviewers
Suna Bensch
Johanna Björklund
Frank Drewes
Patrik Eklund
Petter Ericson
Jerry Eriksson
Thomas Hellström
Lars-Erik Jannlert
Pedher Johansson
Stefan Johansson
Thomas Johansson
Jakub Krzywda
Juan Carlos Nieves
Lena Palmquist
Abel Souza
## Table of Contents

Inertial Dead Reckoning for Two-Dimensional Motions .......................... 1  
*Md Reaz Ashraf Al Abedin*

Exploring the Differences in Performance Between Gamers and Non-Gamers When Completing Tasks Viewed from a Third-Person Perspective ......................................................... 13  
*Arvid Bräne*

Using Your Body to Control a 3D Game Viewed from a 2D Third-Person Perspective ................................................................. 31  
*Lars Englund*

Security or Usability when Selecting Password - Priority Differences due to Security Awareness ......................................................... 41  
*Jonas Gustafsson*

Do Coloured Numbers Improve Speed and Accuracy When Entering a Numerical Password? ................................................................. 51  
*Johan Holmgren*

Can a High Color Contrast Touch Interface Increase User Reaction Time when Using a Smart Phone Web Based Application? ................. 61  
*Albin Hübisch*

Context-Free Graph Grammars for Recognition of Abstract Meaning Representations ................................................................. 69  
*Anna Jonsson*

Model-based RSA Using NURBS Models ................................................. 81  
*Heba Shehabeldin*

Using Dependency Parse Trees as a Method for Grounding Verbal Descriptions to Perceived Objects ....................................................... 95  
*Alexander Sutherland*

Evaluating Negotiation Approaches with Opponent Models for Multi-Agent Systems ................................................................. 107  
*Dawit Kaysay Weldemariam*

Evaluation of Color Association when Receiving a Mobile Notification ... 121  
*Victor Winnhed*

**Author Index** ............................................................................... 129
Inertial Dead Reckoning for Two-Dimensional Motions

Md Reaz Ashraful Abedin
Department of Computing Science
Umeå University, Sweden
mrc14aaz@cs.umu.se

Abstract. Dead reckoning or determining current position of a moving object based on prior position using only inertial sensors is a challenging work. Dead reckoning process may fail severely if small error is introduced in calculation of heading or current position; as all future estimates will be affected by current estimate. In this paper, some of the common facts regarding inertial dead reckoning are addressed and analysed. A system is implemented with two error compensation approaches for two-dimensional motions. One is zero velocity update and another is Kalman filtering. The output shows that followed approaches produce better results than general or direct position calculation. However, it is found that, the effectiveness of the system is limited to short duration movements only.

1 Introduction

Inertial sensors are electro-mechanical devices that sense linear or angular motion and measure inertial parameters, like acceleration, angular velocity etc. With the improvement of MEMS (Micro-electromechanical systems), the inertial sensors become light weighted, wearable and better in accuracy which has created opportunity to use those devices in many sophisticated experiments, like human motion tracking, inertial navigation etc.

Dead reckoning is a procedure in navigation that determines the position of an object at a certain state based on it’s position at immediate prior state as well as velocity, travelled time and physical parameters which may affect the movement of that object. This process is the basic form of inertial navigation system which is widely used in air, marine and robot navigation [1]. At the time before deploying GPS satellites, inertial navigation was the primary way to determine the position of an object within the environment. Nowadays, absolute positioning data obtained from GPS are used combined with inertial data to increase the accuracy of the navigation process. Inertial navigation system is broad concept which involves multiple inertial sensors and measurement of physical parameters, like wind speed, air pressure, vibration etc. However, our focus in this research is the dead reckoning using only accelerometers and gyroscopes.

In [2], [3] and [4], the authors used inertial sensors for tracking position of a pedestrian in both indoor and outdoor environments. In [5], GPS position data...
is used in addition with inertial sensor data for pedestrian tracking in outdoor urban environment. For navigation of automobiles or car like robots, measurements from inertial sensors are fused with GPS data to compensate GPS signal latency and short period of signal outage [6] [7]. But inertial sensors are highly susceptible to noise and therefore measurements are often affected by errors [8]. These measurement errors are unavoidable specially for low-cost inertial sensors. Even very high-cost sensors do not provide fully error free output; rather those sensors use more sophisticated technology to reduce error. This factor makes error compensation a vital part in inertial dead reckoning. In this research, our goal is to do study the factors that affect the performance of inertial dead reckoning, discuss some of the error compensation techniques and use those to implement a dead reckoning system for short distances.

2 Theory

Based on the sensor mounting approach, there are two different configurations available. One is the gimbal system or stable platform system. Another is strap-down system. In gimbal type systems, the sensor is mounted to gimbal like platform, which can rotate to keep the sensor’s body coordinate frame aligned to the global coordinate frame. In this research, we are using strap-down system, where the inertial sensor is mounted rigidly with the object we want to track. Therefore, the data measured by sensors are respect to the body coordinate frame [7].

2.1 Attitude and Heading Calculation

To calculate the position we need to know the orientation (roll, pitch and yaw) of the device, so that the acceleration vector in body frame can be projected to global frame. The simplest way to get the orientation is to integrate the angular velocity about all three axes measured by rate gyroscope. But, gyroscopes are affected by drift. It provides non-zero output even if the rotation is stopped. To compensate this error, acceleration data obtained from accelerometer is fused with gyroscope data. There are several approaches to do this sensor fusion. In [9], Kalman filtering is used to fuse multiple inertial sensor data for orientation calculation. This approach is computationally expensive and sometimes not feasible for applications having processor with low computational power. As an alternative approach, an explicit complementary filter is used in [10], where the algorithm considers the attitude and heading estimation as a deterministic observation problem. In [11], the authors used gradient descent algorithm based orientation filter. In this work, we are using the complementary filter suggested by Mahony et el [10] and implemented by Sebastian O.H. Madgwick who also suggested the alternative solution in [11].

The general approach for attitude estimation is based on the following kinematic equation [10]
\[ \dot{R} = R \Omega_x, \]  

(1)

where \( \dot{R} \) is the time derivative of \( R \), and \( R \) is the orientation of the body coordinate frame with respect to global coordinate frame. \( \Omega_x \) is a skew-symmetric matrix given by

\[
\Omega_x = \begin{pmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{pmatrix}.
\]

(2)

Here, \( \omega_x \), \( \omega_y \) and \( \omega_z \) are measured rates of rotation in \( X \), \( Y \) and \( Z \) axes. Mahony et al. suggested to correct the rotation rate vector \( \omega = (\omega_x, \omega_y, \omega_z)^T \) using a proportional-integral (PI) controller. The corrected form of rotation rate vector becomes

\[ \omega' = \omega + (K_p + K_i \frac{1}{s})e, \]

(3)

where, \( K_p \) and \( K_i \) are proportional and integral gain. \( e \) is the error vector which drives the controller and can be found from the cross product of acceleration vector \( a \) with the gravity vector direction \( v \) obtained from current attitude estimate.

This implemented algorithm uses quaternion representation of estimated orientation. The quaternion is a vector in four dimension having three complex and one real element, which can be used to represent the rotation of a point in three dimension. The quaternion has some advantages over Euler angle representation of rotation. It involves less computation than Euler angle and also avoids singularity in the solution [12] [13].

As a first step, the acceleration \( a \) and angular rate \( \omega \) are measured. The acceleration vector is normalized afterwards. The gravity vector \( v \) is calculated from \( z \) component of quaternion representation of rotation \( R \) [12]

\[
v = \begin{pmatrix}
2(q_2q_4 - q_1q_3) \\
2(q_1q_2 + q_3q_4) \\
q_1^2 - q_2^2 - q_3^2 + q_4^2
\end{pmatrix}.
\]

(4)

Then the error vector is calculated from \( e = a \times v \), which gets multiplied with proportional gain as in equation 3. For integral part, total error is calculated by summing all error vectors multiplied by time interval

\[ Int_n = Int_{n-1} + e \Delta t. \]

(5)

Thus, the corrected form of rotation rate vector is found from

\[ \omega' = \omega + K_p e + k_i Int_n. \]

(6)

So, we get the rate of change of quaternion as

\[ \dot{q} = 0.5(q \times \omega'). \]

(7)
Finally, $\dot{q}$ is integrated and normalized to get the quaternion representation of the gravity compensated orientation, which is later converted to Euler angles.

$$q_n = \frac{q_{n-1} + \dot{q}\Delta t}{\|q_n\|}.$$  \hspace{1cm} (8)

Figure 1 shows the Euler angles calculated using this algorithm for a few seconds of motion. The Euler angles basically represent the orientation (roll, pitch and yaw) in three dimensional Euclidean space.

2.2 Velocity and Position Calculation

The calculation of position involves several steps. Firstly, the measured acceleration $a_s(t) = (a_{sx}(t), a_{sy}(t), a_{sz}(t))^T$ in sensor/body coordinate frame is converted to acceleration in global coordinate frame by multiplying a direction cosine matrix (DCM):

$$a_g(t) = R_{GS}a_s(t) = (R_{SG})^T a_s(t).$$  \hspace{1cm} (9)

Here, $R_{GS}$ is the direction cosine matrix. This matrix can be perceived as an specific arrangement of the unit vector components of sensor co-ordinate system (S) expressed in global reference frame (G). Multiplication of this matrix rotates a vector in S to G. $R_{SG}$ denotes the opposite conversion. The quaternion representation that we get from the implementation of Mahony’s algorithm actually provides $R_{SG}$.

This calculated acceleration includes physical acceleration as well as acceleration due to gravity. So, acceleration caused by gravity is subtracted. The remaining acceleration is then integrated to get velocity.
\[ v_g(t) = v_g(t - 1) + \int_0^t (a_g(t) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}) dt, \quad (10) \]

where \( v_g(t - 1) \) is the prior velocity. By integrating velocity, we get traveled distance.

\[ d_g(t) = d_g(t - 1) + \int_0^t v_g(t) dt. \quad (11) \]

The calculation of velocity and position stated above gets affected by tilt error. When the device is tilted, the acceleration due to gravity has a component along horizontal axis and added up with horizontal physical acceleration. This fact causes bias in horizontal acceleration in global coordinate. This tilt error can be compensated by making adjustment of tilt angle from the measurement of acceleration due to gravity when the device is stationary [8]. As in our case, the movement is constrained to 2D plane, therefore the tilt compensation is not performed.

3 Hardware Description

We used MTi-G as inertial sensor which is manufactured by Xsens Technologies B. V. (see Figure 2). It is a motion sensing device consisting of an integrated GPS with MEMS Inertial Measurement Unit (IMU).

![MTi-G](image)

*Fig. 2. MEMS based MTi-G miniature AHRS*

MTi-G has tri-axial accelerometer and gyroscope that provides calibrated 3D acceleration and 3D rate of turn. Besides, it has a 3D magnetometer and barometer which measures earth-magnetic field and static air pressure respectively. Depending on user defined settings, MTi-G provides raw sensor data as well as processed output. The processed output is generated by a sensor fusion algorithm (Xsense Kalman filter) running in an attitude and heading reference system (AHRS) processor. Despite of being provided with several sensors, we are using only accelerometer and gyroscope to narrow down our focus and decrease computational complexity. More information and specifications about MTi-G can be found in [13].
4 Implementation

4.1 Zero Velocity Update

As mentioned in section 1, inertial sensors are affected by drift. Accelerometer and gyroscope both keep producing non-zero outputs even if the device is completely stationary and not rotated. As the acceleration vector is integrated twice to obtain displacement, the error in accelerometer measurement gets accumulated and grows with time quadratically. Similarly, drift error in rate gyroscope leads to wrong orientation calculation which gets projected to global coordinate and introduce more error in position calculation. Through this way the position estimation diverges drastically from true value within few seconds.

Figure 3 shows the displacement of the device, when it was completely stationary for 60 seconds. It also shows measured acceleration and angular rate of change at that period. It is observed that measured values are non-zero all the time and traveled distance is 114 meters in 60 seconds.

![Acceleration, Velocity and Displacement](image)

**Fig. 3.** Acceleration, velocity and displacement when the object was stationary for 60 seconds

In pedestrian dead reckoning, this error is partially handled by implementing Zero velocity update (ZUPT) [2] [3] [4]. When the pedestrian’s sensor mounted foot touches the ground, the sensor is considered stationary and measured velocity is set to zero. This periodic zeroing of velocity prevents the error to get...
Inertial Dead Reckoning for Two-Dimensional Motions

accumulated and results better output. We also adopted similar strategy to compensate the error. Though in our case, the motion is not guaranteed to be periodic as pedestrian movement, but this strategy will at least help to keep the displacement to zero when the device is stationary.

To determine whether the device is stationary or not, we followed the simplest way. That is, the magnitude of the acceleration vector in sensor coordinate frame $||a_s(t)|| = \sqrt{(a_{sx})^2 + (a_{sy})^2 + (a_{sz})^2}$ is calculated. This value should be sufficiently low during stationary period. A threshold value is set in this case, where acceleration magnitude below this value indicates that the device is stationary. Figure 4 shows the displacement for the same test setup as before with and without zero velocity update. It is observed that, The displacement value with zero velocity update becomes $2.1e-04$ which is significantly less than before.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{distances.png}
\caption{Displacement of the object during 60 seconds of stationary period with and without zero velocity update}
\end{figure}

### 4.2 Kalman Filtering

The drift error during stationary period is compensated by zero velocity update, but this technique is totally useless when the object is in motion. The more time the object is in motion, the more error is added in position calculation. To reduce the growth of error, a Kalman filtering approach is followed. Kalman filter provides statistically optimal estimation of the system parameters using previously estimated values, new measurements and error covariances of process and measurement noises [14].

The algorithm basically starts with predicting the values of the state variables of the system. State variables define the state of the whole system at a certain time $k$ and also control system dynamics. In our case, the state vector consists of position, velocity and acceleration in all three axes of global coordinate frame.
\[ \hat{x}_k = [p_x \ p_y \ p_z \ v_x \ v_y \ v_z \ a_x \ a_y \ a_z]^T. \quad (12) \]

Considering the object stationary at the beginning, the initial value of all state variables are set to zero. The general equation for prediction step is given by

\[ \hat{x}_k = A\hat{x}_{k-1} + B\hat{u} + w(k), \quad (13) \]

where, \( A \) is state transition matrix, \( B \) is control matrix, \( \hat{u} \) is control input vector and \( w(k) \) is process noise. In current system there is no control input is provided. So, the term \( B \) and \( \hat{u} \) will be zero. \( w(k) \) is considered to be zero mean white Gaussian noise with covariance \( Q \). The state transition matrix is given by

\[ A = \begin{pmatrix} I_{3\times3} & \Delta t I_{3\times3} & \Delta t^2 I_{3\times3} \\ 0_{3\times3} & I_{3\times3} & \Delta t I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{pmatrix}. \quad (14) \]

The covariance of the state vector estimate \( P_k \) is calculated as

\[ P_k = AP_{k-1}A^T + Q. \quad (15) \]

The process noise covariance matrix is obtained from Weiner process acceleration model \([15]\) as

\[ Q = E[w(k)w(k)^T] = \sigma^2_q \begin{pmatrix} \Delta t^5 & \Delta t^4 & \Delta t^3 \\ \Delta t^4 & \Delta t^3 & \Delta t^2 \\ \Delta t^3 & \Delta t^2 & \Delta t \end{pmatrix}. \quad (16) \]

After completing prediction of state variables, Kalman filter uses the measurements to correct the prediction. In our model, we use calibrated acceleration data from the sensor as measurement values. The measurement is related to state vector as

\[ z = H\hat{x}_k + v(k), \quad (17) \]

where \( v(k) \) is measurement noise and considered to be zero mean white Gaussian noise with covariance \( R \). \( H \) is the measurement matrix constructed as

\[ H = \begin{pmatrix} 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{pmatrix}. \quad (18) \]

A correction step calculates Kalman gain \( K' \), new state covariance matrix \( P'_k \) and corrected state vector \( \hat{x}'_k \) as

\[ \hat{k}' = P_kH_k^T(H_kP_kH_k^T + R_k)^{-1}, \quad (19) \]

\[ \hat{x}'_k = \hat{x}_k + K'(z_k - H_k\hat{x}_k), \quad (20) \]

\[ \hat{P}'_k = P_k - K'H_kP_k. \quad (21) \]

The whole process stated above repeats recursively; however it does not need to remember all previous estimations but the last one only. The covariance matrix \( P_k \) is directly related to Kalman gain \( K \) (eq. 19), where the Kalman gain determines the fidelity of new measurements and prior estimates.
5 Test Results and Evaluations

There are several test cases through which the implemented system is evaluated and analysed. As mentioned earlier, our focus was for two-dimensional motion, basically the XY plane of the global coordinated frame. The MTi-G is attached with a rigid body object, and that object is moved along predefined path on a surface. The sensor readings are recorded real-time using a software (MT manager) and later processed using MATLAB.

In section 4.1, it is shown that zero velocity update improves the position output when the object is stationary or in rest between motions. However, the direct method involving double integration of acceleration facilitated with ZUPT does not provide proper position output when the object is in motion. Afterwards, a Kalman filter is implemented to improve the position estimation. Figure 5 and 6 show that Kalman filtering approach follows the actual path of motion more closely than the direct approach, though still the estimated output is not fully accurate.

![Fig. 5. Comparison of determined Positions for a circular path using direct and Kalman filtering approach](image)

The evaluation is done in three ways. One is comparing total travelled distance and actual path length. Another way is calculating error distance between start and stop position for close loop motions. And, third way is to perform visual inspection. That is to check how close the estimated positions follow the actual path. All the distances are calculated in XY plane of global coordinate frame. The lowest test distance was half a meter and the highest was 7 meters.
Fig. 6. Comparison of determined Positions for a rectangular path using direct and Kalman filtering approach

The Kalman filter starts to diverge from actual path after a certain time period depending on the speed and complexity of the motion. Average error in travelled distance and overall position for different close loop motions are calculated. The direct approach exhibits approximately 25% error where Kalman filtering approach reduces it to 12%. Sometimes, inconstancy is observed in the error values. An as example, calculated travelled distance is very close to actual path length, but the distance between start and stop position is considerably high. This fact can be explained by the possibility of imperfect heading angle. No magnetometer data is used in this experiment to avoid magnetic field mapping for soft/hard iron effect compensation. Though it is possible to calculate heading angle without magnetometer data by using measurement from gyroscope, but it may become incorrect with time because of gyro drift. This fact causes wrong heading angle and as a result calculated positions becomes inaccurate.

6 Conclusion

We have studied the challenges in determining position using inertial sensors and implemented a system that uses two approaches to compensate error and improve position calculation for two-dimensional motions. It is observed that, position calculation using only accelerometer and gyroscope is highly error prone and can be used effectively only for short duration movements. A practical use
of this system can be in gaming console where the users typically need short and sometimes periodic movements.

The system performance can be improved by imposing constraints in motion pattern and periodically correcting the velocity of the object. Use of magnetometer with magnetic field mapping can increase heading accuracy to a great extent which will ensure better position estimation. Moreover, fusion between inertial sensor data with other displacement or position sensor data, like GPS, ultrasonic sensors, camera, pedometer etc will extend the scope of this research to use for indoor and outdoor navigation.

References

Exploring the Differences in Performance Between Gamers and Non-Gamers When Completing Tasks Viewed from a Third-Person Perspective

Arvid Bräne
Department of Computing Science
Umeå University, Sweden.
Email: arvidbrane@gmail.com
Website: http://www.arvidbrane.se

Abstract. The concept of third-person perspective in gaming has been around since the start of graphics in video games. This study aims to investigate if there is a measurable difference in performance between gamers and non-gamers when they complete the same tasks from a third-person perspective. Experiments were made using a back-mounted camera rig and pair of video goggles. Results, generated from a small amount of participants, suggest that there is no significant difference in performance between the two groups when adjusting to a third-person view.

1 Introduction

There is a constantly ongoing debate on whether playing video games produce negative side-effects or not [1]. Some studies suggest links between violent video games and increases aggressive behavior, decreases in helping behavior [2] and decreased prosocial behavior [3]. Earlier findings indicate that committing “immoral” virtual behaviors in a video game can lead to increased moral sensitivity of the player [4] and that playing video games do not have any effect on depression, hostility, or visuospatial cognition [5]. There are even results in experiments that suggest violent games reduce depression and hostile feelings in players through mood management [6]. There is also research hinting that video games can result in positive side-effects such as improved cognitive control, emotional regulation, spatial resolution of vision, hand-eye motor coordination, and contrast sensitivity [7]. Other results point towards an improved decision making (probabilistic inference) without loss of measurable accuracy [8].

This study aims to investigate if there is a measurable difference in performance, such as number of errors made and time consumption, between people who have played video games (gamers) and people who have not (non-gamers) when they are prompted to complete specific tasks viewed from third-person perspective\(^1\) (see Figure 1). Similar studies have been conducted, both survey based

\(^1\) A perspective were the view is at a fixed distance behind and slightly above the user, often used in video games.
questioners [9] and game-related experiments using augmented reality [10], but certain aspects about performance differences in tasks that heavily depend on orientation, navigation and balance remain unaddressed. This study was completed using a custom-made rig in order to simulate the experience of a life viewed from a third-person perspective.

1.1 Earlier Work

Studies in literature have previously shown that most readers do not have any recognition about whether a book they have read was written in first- or third-person [11] due to humans capability of “translating” and adapting from one pronoun to another. Kohler’s experiments with inverted vision goggles showed subjects walking and riding bicycles while seeing upside-down [12], pointing towards even greater ability for the brain to adapt. This could suggest that users might be able to adapt to seeing themselves from third-person perspective in a relatively short time, something suggested by prior studies [10].

Experiments measuring navigation and movement performance [13], similar to our experiments, have also been conducted. These were performed in virtual reality (VR) using different interfaces (joystick-only, linear and omni-directional treadmills, and actual walking) to control their navigation in the VR world. Further studies suggest that walking interfaces are to be preferred when navigating three-dimensional virtual environments [14].

2 Material & Method

Studies prior to this one have been done on the differences between gamers and non-gamers, such as [9] and [7], but only a minority using hardware to simulate the out-of-body third-person view experienced in games (see Figure 1) in real life. Our method of choice was to construct a custom designed rig where subjects saw themselves in real-time from a third-person perspective. In order to see the differences between the groups they had to complete the same three tasks in three different perspectives. After the subjects finished their participation, they were prompted to fill in a form regarding the experience and their prior experience with video games. The two groups, consisting of 13 subjects (undergraduate volunteers, 12 male subjects and two female, in ages ranging from 23 to 28), were benchmarked against each other to see which performed better. Originally there were 14 subjects in the study, however one participant could not complete the whole experiment due to the subjects poor eyesight when not wearing his glasses. This subject was therefore excluded from the study after the first task and not included in the results.
Fig. 1. A typical third-person perspective from the game *Grand Theft Auto: V*. The point of view is shifted from the typical position (the character’s eyes) to behind and above the subject resulting in a wider and unreal field of view.

2.1 Rig Design

In order to fully simulate a game-like, out-of-body experience and a third-person perspective (see Figure 1), without leaving the participants nauseated\(^2\) the rig had to be as rigid as possible. The main parts in the rig were:

**Back & Camera Mount** A solid mounting foundation was constructed out of light weight and stiff materials such as carbon fiber, ABS and Polymorph\(^3\) plastic. As a base a snowboard back protector was used in order to connect a carbon fiber rod to the subjects back. Some 3D-printed parts were used to fasten the third-person camera to the rod.

**Third-Person Video Camera** The video camera used for the third-person view, constantly generating a live video stream, was mounted on a rod circa one meter and approximately 45 degrees above/below the participants head and tilted circa 30 degrees downwards in order to frame the video correctly. Since a large field of view\(^4\) and a compact- and lightweight design were the most important requirements for selecting the video camera, a *GoPro Hero 3: Black Edition*\(^5\) was chosen, weighing 163 grams and a diagonal field of view of 149 degrees. The camera was connected to the participants video goggles using a three meter long cable.

---

\(^2\) Early test showed that participants felt sea-sick due to unwanted camera movement created by an unstable test-rig.

\(^3\) More information can be found at [http://www.polymorphplastic.co.uk/](http://www.polymorphplastic.co.uk/)

\(^4\) The restriction in the visible view

\(^5\) More information can be found at [https://goo.gl/dLW5uz](https://goo.gl/dLW5uz)
Video Goggles & First-Person Video Camera To cover the subjects eyes and view the video stream a pair of video goggles were used. These goggles, a pair of SkyZone SKY-01 V2\textsuperscript{6}, have a built in screen and an onboard camera with a diagonal field of view at 120 degrees. This camera was used for the second configuration for each task (described in Section 2.3) to simulate first-person perspective.

The design was inspired by the rig used in Quantifying effects of exposure to the third and first-person perspectives in virtual-reality-based training [15] (but with more up-to-date hardware) and is illustrated in Figure 3. An approximation (the top and bottom of the image is cut of due to the camera not being able to capture non-wide screen video\textsuperscript{7}) of what the subjects saw is demonstrated in Figure 2.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{An approximate (the top and bottom of the image is cut of due to the camera not being able to capture non-wide screen video) view of what the participant saw in third-person configuration during the experiments.}
\end{figure}

\subsection{2.2 Task Design}

Three different tasks were chosen to measure three separate areas; aiming accuracy, balance and movement control:

\begin{itemize}
\item More information can be found at \url{http://www.foxtechfpv.com/skyzone-fpv-goggles-matte-black-preorder-p-1218.html}
\item During the experiment the user was able to see approximately 30 cm behind their feet.
\end{itemize}
Task 1: Accuracy The test subjects rolled, threw or bounced (depending on their preference) a multi-colored volleyball in order to try and hit a target (a regular sized chair) placed approximately 5 meters away to successfully complete the task. If the test subject missed the target they were told to try again until they finally hit it. This test measured the participants precise accuracy and ball control through the number of tries required in order to hit the target.

Task 2: Balance The test subjects walked in their preferred speed on a thin straight line made out of tape, 10 meters long placed on the ground. This test measured the participants balance skill through the number of errors they made. These errors were measured and recorded using one pre-defined rule; if any part of the shoe/foot covered at least the width of the tape (approximately 2 cm wide) it was considered to be a legal foot placement, everything else was illegal. Examples of illegal foot placements can be found in Figure 4 and legal examples can be found in Figure 5.

Task 3: Movement Test subjects walked facing forward in their normal walking speed, through a pre-planned course approximately 25 meters long and circa two meters wide (see Figure 6). Participants were told to do so without touching anything other than the floor. The course was constructed using 40 chairs, the actual width varied from around 1.5 to 2.5 meters throughout the course.
five tables, one large wooden box, a five meter long wall and a tall circular pillar. Participants started between two chairs and finished when they stepped on the cross marked with tape on the floor. This test measured the participants movement and navigational skills through the required time it took in order to complete the task.

2.3 Configurations

Each task was performed three times by each participant, in three different configurations resulting in a total of nine results for each participant and task. The different configurations were completed in the following order:

1. Off: Not wearing the rig, video goggles off.
2. First-Person: Wearing the rig, video goggles on, viewed from first-person camera.
3. Third-Person: Wearing the rig, video goggles on, viewed from third-person camera.

Completing the task three times was done to get a more accurate average of each of the participants performance. The first configuration served as a baseline for how a participant performs when completing the task “normally”. The second configuration was used to compare against the first configuration in order to understand how much the video goggles and the first-person camera affected
Fig. 6. At the top: A side-view of the course used during the Movement Task. At the bottom: A top-view of the course used during the Movement Task. The thin, long red line is the approximate distance where the participants walked, the beige rectangles are tables, the small blue squares are chairs, the short green lines are tape on the floor, the large brown square is a wooden box, and the gray is the long wall and the circular pillar.
the participants performance in completing the tasks. Comparing the third and second configuration was the main focus of this study which was why the third configuration was the most critical one.

2.4 Survey Design

After each test subject finished his/hers participation in the experiment they were prompted to fill out a survey regarding the experience during the experiment and their prior experience with video games. The survey (the full form is found in the Appendix 5) included the following seven questions:

1. Do you consider yourself a *gamer*?
2. What was the hardest parts in the experiment?
3. On average, how many hours per week do you spend playing video games?
4. How many years have you been playing video games?
5. In total, how many hours have you spent playing a game viewed from a third-person perspective?
6. If any, please name some of these third-person games you have played.
7. Did you find your participation in this experiment fun?

Each test subject also filled in details about their name, age and sex so the results from the test data could be paired up with the surveys. The details were later removed in the results in order to keep the test subjects anonymity.

Questions 2 and 7 were asked in order to review the experiment, findings of which can be found in Section 4.1.

2.5 Group Classification

We classified each participant into one of the two groups, either the subject was a *gamer* or a *non-gamer*. While each subject had an opinion about which group they belonged in, the classification needed to be objective. In order to be regarded as a gamer the subject had to fulfill four requirements:

1. An average of five hours or more spent playing games every week.
2. A total of more than 80 hours playtime in a third-person game.
3. Seven years or more of experience playing video games.
4. Listing at least three third-person games they have played.

3 Results

The statistical tests used in the results are paired, two tailed t-tests.
**Accuracy Task**  As seen in Table 1 the average performance, as in number of tries required in order to hit the target, is generally good (as in a low number of tries) for both groups. Whilst the average gamer\(^9\) generally performed slightly better in both first- and third-person configurations, there is no significant difference (p-value at 0.63) between the two groups.

Furthermore, looking at the graph in Figure 7 we see that the percental average individual difference in performance is generally lower for gamers. This indicates that the average gamer had less trouble with readjustment when changing between the different configurations. This conclusion could however not be statistically confirmed (p-value at 0.54).

<table>
<thead>
<tr>
<th></th>
<th>Gamers</th>
<th>Non-gamers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Off</td>
<td>1.3 tries</td>
<td>1 tries</td>
</tr>
<tr>
<td>2. First-Person</td>
<td>1.5 tries</td>
<td>1.6 tries</td>
</tr>
<tr>
<td>3. Third-Person</td>
<td>1.7 tries</td>
<td>1.9 tries</td>
</tr>
</tbody>
</table>

**Table 1.** Average performance (tries required before hitting the target, where less is better) for the two groups for the Accuracy Task for the three test configurations. Standard deviation stretching from 0.2 (non-gamers in configuration one) to 1.2 (gamers in configuration three) at most.

**Balance Task**  Unlike the results from the accuracy task, on an average, a gamer performed slightly worse than a non-gamer in the third-person configuration. As seen in Table 2 the only notable difference between the groups is the last configuration. The numbers of illegal steps dramatically increased when viewed from third-person view, from 0 (for both groups) to 9.7 (non-gamers) respectively 12.3 (gamers) steps.

When turning our attention towards the percental average individual difference in performance presented in Figure 8 we could only conclude that the average gamer had less difficulty with readjustment when changing between the last two configurations\(^10\). This hypothesis was rejected after a t-test (p-value at 0.59).

**Movement Task**  Similarly to the findings in the balance task, results in the movement task (found in Table 3) suggest that the average gamer performs worse, as in more time required to complete the course, than the average non-gamer.

Turning our attention towards Figure 9 we can see that the percental average individual difference in performance is generally higher (0.2, 4, and 6.7 seconds)

\(^9\) The average of the results from all the gamers.

\(^10\) Although the standard deviation was generally high at 528 amongst non-gamers, respectively 160 between gamers.
Fig. 7. Average individual difference (percent, less is better) in performance (tries required, less is better) between two configurations for the two groups for the accuracy task. Standard deviation stretching from 49.2 (non-gamers comparing configuration two and three) to 81.5 (non-gamers comparing configuration one and two). For example, the average individual throws required for a gamer in configuration three is 150% of the required throws in configuration one, therefore 50% worse/more.

Table 2. Average performance (errors made while walking the line, where less is better) for the two groups for the balance task. Standard deviation first-person configuration was 2 for gamers and 3.7 for non-gamers, respectively 3.4 and 4.4 for third-person configuration. No deviation for the first configuration for any of the groups.
Fig. 8. Average individual difference (percent, less is better) in performance (errors made while walking the line, less is better) between the second and third configurations for the two groups for the balance task. For example, the average individual number of errors for a gamer in configuration three is 318.4% of the number of errors in configuration two, therefore 218.4% worse/more. The first two comparisons (first and second configuration, first and third configuration) were inconclusive due to the first configuration being 0 for both groups, therefore not comparable.
for the average gamer than the average non-gamer. Unlike in the two earlier tasks this result was confirmed (p-value at 0.02), after establishing that both datasets are normally distributed with a normality test.

<table>
<thead>
<tr>
<th></th>
<th>Gamers</th>
<th>Non-gamers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Off</td>
<td>20.9 seconds</td>
<td>20.7 seconds</td>
</tr>
<tr>
<td>2. First-Person</td>
<td>33 seconds</td>
<td>29.6 seconds</td>
</tr>
<tr>
<td>3. Third-Person</td>
<td>40.7 seconds</td>
<td>34 seconds</td>
</tr>
</tbody>
</table>

*Table 3.* Average performance (time required in order to finish the course, where less is better) for the two groups for the movement task.

*Fig. 9.* Average individual difference (percent, where less is better) in performance (time required in order to finish the course, where less is better) between two configurations for the two groups for the movement task. For example, the average individual time required for a gamer in configuration two is 157.6\% of the time required in configuration two, therefore 57.6\% worse/more.

### 4 Discussion

Earlier work suggest that third-person perspective causes no significant discomfort while at the same time having a short learning time [10], something we found contradictory to our results. Participants generally performed worse when executing all tasks in the second configuration (first-person perspective) compared
to the first (normal viewing) configuration. Subjects normally performed even worse in the third-person configuration, somewhat rejecting earlier findings.

4.1 Additional Findings

Since the low amount of participants in the study influence the results, no final conclusions can be made about the difference in performance between third-person view and normal viewing when comparing gamers and non-gamers. However, additional findings where made during the experiments, these include:

- All 13 participants in the study said they found the third-person view (configuration three) difficult while only five did for the second configuration (first-person view).
- Walking on a straight line on the floor (as participants did in the balance task) in third-person view (configuration three) was exceptionally hard regardless to the subjects gaming background and sex. This was due to the lack of vision of the line in front of the participants feet and body while walking.
- What defines a gamer is more of a subjective opinion than an objective classification. This became apparent when using the objective group classification (described in Section 2.5), five subjects, where four considered themselves as gamers, meet the criteria. However two participants, that both considered themselves as gamers, did not meet the criteria and were therefore referred to as non-gamers.
- Familiarizing participants with the concept of moving their field of view using their hips rather than their neck turned out to require more time than first anticipated. Even after 15 minutes of wearing the rig in third-person perspective participants were moving their neck instead of their hips to look around.
- Depth perception is generally hard when viewing a camera stream from a wide field of view camera lens, especially without stereoscopic vision.
- Although most participants felt slightly nauseated after the experiment, none lost complete balance and fell. As a plus, all participants said they enjoyed taking part in the experience.
- Findings amongst non-gamers suggest that there is no significant measurable difference in performance between the sexes for any of the tasks in any configuration.

4.2 Limitations and Drawbacks

Due to the time and budget limitations there are several ways to improve upon our experiments. The largest, and possibly most significant, is the low number of participants in the study. Other limitations and drawbacks include:

**Rig Design** Although the rig was rigid enough for this particular experiment, reinforcements should be made in order to continue with further testing.
The biggest drawback of the current rig are the shakes generated on fast movements such as running or fast turns. This could be fixed by connecting one, or preferably two, more booms on an angle to both the back mount and the current booms to counteract horizontal and vertical vibrations. Another solution could also be to purchase an already tested and viable rig such as the 3rdPersonView\textsuperscript{11} from Sail Video System.

**Camera Movement** Normally in a virtual third-person game, such as GTA (see Figure 1), the “camera” follows the characters movements with a slight delay in order to get more fluid camera movements. The current setup does not currently support this due to the cameras fixed attachment to the back mount, however this could be corrected using a three-axis gimbal, something that would also improve the overall stability of the camera. Adding an IMU\textsuperscript{12} to the users video goggles would also allow for the user to look around using his/hers normal head movements.

**Task Design** The tests chosen for this study, especially the task one and two, aimed to test specific abilities, such as accuracy, balance and navigation. While this is a start, more relevant and less specific test could be conducted using more everyday-like tasks, such as riding a bike, walking to work, cooking food etc.

**Video Goggles** Whilst the video goggles used had an average resolution, more sophisticated video goggles, such as the *Oculus Rift*\textsuperscript{13} or the *HTC Vive*\textsuperscript{14}, with a higher pixel count could be used to create a more immerse and believable simulation. Since both of these are made for virtual reality gaming, their field of view is notably greater than in the *SkyZone* goggles used. Using an actual VR headset would also add stereoscopic vision, a feature that might have made a difference on our results.

**More Segregated Groups** As discussed earlier in Section 4, the definition of *what defines a gamer* is not apparent. Since none of the student volunteers were professional, full-time gamers we cannot make any statements about *actual gamers*\textsuperscript{15}. The same goes for non-gamers; most of our subjects have sometime in their life been exposed to video games to some extent, either playing themselves or watching someone else playing. This results in oblique findings about non-gamers as well.

**Sex Ratio** Due to the high male skew in the study (especially amongst gamers), no conclusive findings about difference, or indifference, between males and females were found.

\textsuperscript{11} More information can be found at http://www.sailvideosystem.com/p/3rdpersonview-all-sports-pro-166682
\textsuperscript{12} Inertial Measurement Unit, such as a gyroscope and an accelerometer
\textsuperscript{13} More information can be found at https://www.oculus.com/en-us/rift/
\textsuperscript{14} More information can be found at http://www.htcvr.com/
\textsuperscript{15} A person who spends most of their awake time playing games, mostly professionally but also casually
4.3 Conclusion

We believe this study should serve as a foundation and a guide for further research in the future and not as reference material for any hard proof. In order to fully study the differences between the groups one would need a larger participant group with a greater segregation in time spent playing video games.

5 Acknowledgments

The authors would like to thank all participants in the study for their time, participation and feedback. We would also like to thank all the peer-reviewers (especially Lars Englund for his help during the experiments) who helped form this report into its final shape along with David Källberg for his help with the statistical analysis of the data. Last but not least we would like to thank Umeå University for letting us use their facilities Rotundan (where we conducted our experiments) and Robotlabbet (where we constructed the rig) during the progress of the study.

References


Appendix

All the files for this report, along with all the 3D-design-files and experiment results can be found and downloaded on the GitHub-page for this project.

Survey

\footnote{More information can be found at https://github.com/Kodagrux/Third-Person-Performance-Differences-Between-Gamers-and-Nongamers}
Third-Person Performance Differences Between Gamers and Non-gamers

---

**Third-Person Tests**

Thank you for participating in our study about *Exploring the differences in performance between gamers and non-gamers when completing everyday tasks viewed from a third person perspective.* Your information will be kept secret and anonymous once the scientific results are published, we collect them just so we can tell the different test subjects apart.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>☐ Male ☐ Female</td>
</tr>
</tbody>
</table>

1. Do you consider yourself a “gamer”?
   “Gamer” as in “Video gamer”
   ☐ Yes ☐ No

2. What was the hardest part in the experiment?
   *Circle as many as you like and/or add your own*

   - The tasks themselves
   - The third-person view
   - The first-person view
   - Unclear instructions
   - The resolution in the video goggles
   - Trusting in the rigs design
   - Other hard parts?

---

*Survey continues on the back!*
3. On average, how many hours per week do you spend playing video games?
Circle the number closest to your answer
0 0.5 1 1.5 2 2.5 3 4 5 7 8 9 10+

4. How many years have you been playing video games?
Circle the number closest to your answer
0 0.5 1 1.5 2 2.5 3 4 5 7 8 9 10+

5. In total, how many hours have you spent playing a game viewed from a third-person perspective?
Circle the number closest to your answer
0 1 3 5 10 15 20 25 30 40 50 60 80+

6. If any, please name some of these third-person games you have played:
As many as you can think of

7. Did you find your participation in this experiment fun?
○ Yes ○ No

Thank you, have a nice day! :)}
Using Your Body to Control a 3D Game Viewed from a 2D Third-Person Perspective

Lars Englund
Department of Computing Science
Umeå University, Sweden
id11led@cs.umu.se
www.englundlars.se

Abstract. In virtual reality gaming there is a clear dominance of first-person view games. What if we use third-person view instead, would it then be preferable to view yourself from the top or from the side? There are studies done on third person perspective in virtual worlds, but none comparing two different angles with each other. This empirical study explores this gap by observing people completing an obstacle course wearing video goggles. We have used the real world and video goggles to approach total immersion and simulate a virtual world. Even though the number of participants were not big enough to prove a statistical difference between the two views, there were some indications pointing towards a benefit for the top-view.

1 Introduction

The virtual reality market and its technology is improving every year. Game makers and hardware developers are constantly producing more advanced gameplay experiences. Examples are the Virtuix Omni\(^1\) and the Cyberith Virtualizer\(^2\), these are omnidirectional treadmills that allows the user to walk or run in any direction (Figure 1). They also register jumps, crouches and can be used with additional equipment such as different game controllers (joy-sticks, rifles, steering wheels etc.). The users movements are transferred to movement in a video game or other virtual worlds, this means that you control the game with your normal body movements. One common application for virtual reality video games is to simulate a first-person view [1], where your eyes see what the character in the game sees. To broaden the variation of virtual reality video games we have to try new angles and approaches. What if a game is supposed to be viewed from a third-person perspective, generating a two-dimensional view (like the Nintendo game Super Mario 1) and still be three-dimensional (you are able to move in any direction). Is it easier to control your body when you see yourself from the side, or from above? To answer this question we have performed an experiment where test subjects made their way through an obstacle course and tested both to see themselves from the side, and from above.

\(^1\) More information at http://www.virtuix.com/
\(^2\) More information at http://cyberith.com/
To get closer to total immersion\textsuperscript{3} the need for low latency visual feedback is critical\textsuperscript{2} and since there is no virtual reality equipment that can achieve total immersion yet\textsuperscript{3}, the participants did not act in a virtual world, but in the real world. This provides every action (body movements) and feedback (touch, sound, hearing etc.) that we are used to. Video goggles and a camera were used to generate and present the side-view and top-view to the participants.

The hypothesis is that viewing yourself from the side is preferable, because it might be easier to see how high (from the ground) the obstacles are. This could be more beneficial than the extra width of view, provided by the top-view. The study was conducted at Umeå University on 10 student volunteers.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{omnidirectional_treadmills.png}
\caption{The omnidirectional treadmills made for virtual reality gaming, Cyberith Virtualizer to the left and the Virtuix Omni to the right\textsuperscript{5}.}
\end{figure}

1.1 Earlier Work

Ruddle, R. A., Volkova, E., and Bulthoff, H. conducted a study\textsuperscript{4} were different alternatives how to move in a 3D VR-world ranging from a joystick to omnidirectional treadmills where tested. The main difference between our and Ruddle’s study is that they used a flat world with virtual walls while we used physical objects and a non flat floor (sand) that the test subjects had to encounter. One

\textsuperscript{3} Immersion into virtual reality is a perception of being physically present in a non-physical world.

interesting finding from their study was that people need some time to adapt to new environments. The time of adaption was something also examined by Kohler [5] where his result showed “the eye’s remarkable ability to correct for distortions”, our experiment resembles this experiment but with another type of distortion. Another study [6] by Ruddle shows that it is preferable to use natural interactions to control your movements in virtual reality. “The third-person point-of-view augments the limited information of the first-person point-of-view, and suggests another aspect of this problem: embodiment is not merely seeing more (i.e., peripherally), but seeing within a context, whose meaning extends well beyond the optical registers privileged by most games.” [7]. A quote from Laurie N. Taylor that inspired this study, she implies that a third-person perspective in video games can replace the loss of other senses than sight. For example can you hear and feel the presence of a person behind you in real life, something that is hard to simulate in a video game.

2 Method & Material

To try the hypothesis an obstacle course containing obstacles with different features was set up to mimic game-like situations, including getting over or under obstacles and navigating in an open world without a flat floor. The test subjects completed the obstacle course multiple times with different views of themselves (top, side, regular and blinded). To compare the different views the test subjects were clocked while completing the course.

2.1 Setting

To strive for total immersion we chose the setting to be reality rather than virtual reality. This to minimize external factors like graphics in the VR world, sensors that can produce delay or misreadings and other technical constraints.

2.2 Setup

The experiment was performed on a beach volleyball court in Umeå, Sweden. A camera was mounted on a pole carried by the test supervisor (Figure 2 and 3). The pole was adjusted to capture a video stream from above or side of the test subject. Extra information was gathered by a questionnaire (Figure 6 in Appendix) asking about age, sex, any involvement with similar experience and which view “they felt to be easier” (the exact formulation from the survey). To facilitate the experiment an assistant was assigned to use a stopwatch and to measure the time for each run of the course.

2.3 Obstacle Course

The obstacle course was 16 meter long. Obstacles in the course consisted of 3 low boxes (height 50 cm), 1 higher box (height 75 cm), 1 small traffic cone
Fig. 2. The top-view experiment setup, the camera was above and slightly behind the test subject, aimed downwards.

Fig. 3. The side-view experiment setup, the camera was aimed straight at the test subject.
and 1 net. (Figure 4). The entire obstacle course was setup on top of sand to prevent fall injuries if participants lost their balance. The instructions given to the participants was to: *go over the boxes, under the net, once at the traffic cone, turn around and complete the same stretch but reversed. You do not have to complete the course as fast as you can, use a pace where you feel secure and in control.*

![Course design](image)

**Fig. 4.** Course design overview, the low obstacle was 0.5 meter high, the high obstacle was 0.75 meter high. The gap between the net and the ground was approximately 1.2 meter.

### 2.4 Test Group

The group consisted of 10 people (10% female) aged between 23 and 28. A majority of the subjects had previous experiences with video-goggles and seeing theirself in real-time. All subjects were undergraduate students at Umeå University.

### 2.5 Experiment

The test subjects made their way through the obstacle course once without any equipment attached, to familiarize themselves with the course. Then followed one of the three options listed below, every participant completed the course two times with every option. The different views were assigned in a random order to counteract biased results (by experience from previous runs).

**Side-view** The supervisor followed the subject in parallel beside the course, constantly aiming the camera at the test subject, approximately 2.5 meter from the subject.
Top-view The supervisor followed the subject from behind, constantly aiming the camera from approximately 1 meter above the subject’s head.

Blinded The test subject got no visual feedback, they had to rely on other senses.

When side-view and top-view were tested the supervisor mounted the video goggles on the subject’s head, and placed a battery to power the goggles in the subject’s pocket. When the subject and supervisor were ready the assistant gave a start signal and started the stopwatch, when the subject completed the course the stopwatch was stopped and the result was recorded in whole seconds.

2.6 Equipment

Equipment used during the experiment

- Camera (GoPro Hero 3 Black Edition\(^6\))
- Mono-pod (110 cm)
- Video goggles (Quanum DIY FPV Goggle Set with Monitor\(^7\))
- Batteries to power video goggles
- Cables

The camera shows 122.6 degrees horizontal, 94.4 degrees vertical and 149.2 degrees of diagonal field of view\(^8\). The video-goggles resolution is 480x360 pixels.

The equipment can not be compared to the human eye when it comes to delay, sharpness, color and optical distortion. But it gives a good approximation of what a video game looks like.

3 Result

In summary, we took the 2 runs per view from each participant and calculated an average. We chose to use an average value due to the small dataset collected and to give a good representation of what an generic value would be. (All individual time measurements can be found in the appendix 6, Table 2).

\[
\frac{x_{first\ run} + x_{second\ run}}{2} = x_{average}
\]

A paired t-test was performed with the average top-view and side-view values from each participant, where \( H_0 \): No difference between the two views, gave \( p = 0.43 \) (Table 3). This value is too high to discard \( H_0 \), indicating that there was no statistical difference between the two views in the collected data.

If we look at the results without any statistical analysis, a majority of the participants completed the obstacle course faster when viewing themselves from the top (Figure 5).

\(^6\) Link to specifications [http://goo.gl/3yjWpB](http://goo.gl/3yjWpB)

\(^7\) Information at HobbyKing’s webpage [http://goo.gl/BxHVOY](http://goo.gl/BxHVOY)

\(^8\) Information at GoPro’s webpage [https://goo.gl/EiFjbT](https://goo.gl/EiFjbT)
Table 1. Mean and variance from the average results and results from a paired t-test of the average runs

**Additional findings**

- 50% of the participants performed best (shortest average time) without any visual feedback (blinded).
- 90% of the participants felt dizzy or nauseous during or after the experiment.
- 70% of the participants answer of their preferred view did not match their actual better view.

The view to result in the shortest average time per participant

![Pie chart showing distribution of views](image)

**Fig. 5.** A majority (60%) completed the obstacle course faster (lower average time) while viewing themselves from the top, 30% of the participants performed better while viewing themselves from the side. 10% got the same average time (measured in second accuracy).

4 Conclusion

The results (Figure 5) from this experiment was shown to contradict the hypothesis. Only 30% of the participants performed better during the side-view, 60% performed better during top-view and for 10% the different views did not differ. Top-views variance (Table 3) is lower than the side-views, indicating that the top-view might generate more regular outcome, independent of person using it. Compared to similar studies [4] [6] [3] [1] this study also measured the task completely blinded. 50% of the subjects completed the task faster this way, indicating that a visual feedback might not help as much as we might suspect, something also found by Laurie N. Taylor [7]
5 Discussion

Simulating virtual reality in the real world with a video camera and goggles is an estimate. But to get closer to total immersion it is a quite good estimate.

5.1 Limitations

- The world showed to the subjects through the goggles during the test was not virtual, it was the real world. This affected the test subjects which tended to move cautiously as they were afraid to fall over, if the experience was truly virtual the subject might have behaved differently.
- The camera equipment produced a small delay between capturing video and showing the video in the goggles, this generate a delayed visual feedback of the subjects body movements. This could influence the test subjects in a negative way which can affect the credibility of this study. On the other hand, the delay were equal on both top-view and side-view.
- As the camera was held by the supervisor, the video shown in the video-goggles was not exactly the same for all runs. Especially when the supervisor moved the camera under the net obstacle.

5.2 Future Work

To conduct this test in greater detail a suggestion is to create a computer generated world and use sensors on the body as input, moving the experiment to true virtual reality. Other test scenarios than an obstacle course could reinforce or reject our conclusion of this study.

6 Acknowledgements

We would like to thank all the participants for their time and effort. Also the peer-reviewers for their feedback. A special thanks to Arvid Bräne for his assistance, feedback and lending of equipment. This study would not been possible without you all.

References


Appendix

The appendix contains the collected data from the experiment and the survey that the participants answered.

Collected data

<table>
<thead>
<tr>
<th>Subject nr</th>
<th>Side-view 1'st run</th>
<th>Side-view 2'nd run</th>
<th>Top-view 1'st run</th>
<th>Top-view 2'nd run</th>
<th>Blinded 1'st run</th>
<th>Blinded 2'nd run</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>02.40</td>
<td>01.26</td>
<td>01.55</td>
<td>01.09</td>
<td>01.17</td>
<td>01.16</td>
</tr>
<tr>
<td>2</td>
<td>02.06</td>
<td>01.24</td>
<td>01.26</td>
<td>01.06</td>
<td>01.22</td>
<td>01.03</td>
</tr>
<tr>
<td>3</td>
<td>01.16</td>
<td>01.02</td>
<td>01.08</td>
<td>01.02</td>
<td>01.43</td>
<td>01.21</td>
</tr>
<tr>
<td>4</td>
<td>01.09</td>
<td>01.08</td>
<td>01.27</td>
<td>01.03</td>
<td>01.16</td>
<td>01.07</td>
</tr>
<tr>
<td>5</td>
<td>00.57</td>
<td>00.49</td>
<td>00.55</td>
<td>00.49</td>
<td>01.07</td>
<td>00.58</td>
</tr>
<tr>
<td>6</td>
<td>01.10</td>
<td>01.03</td>
<td>00.58</td>
<td>00.58</td>
<td>01.06</td>
<td>00.55</td>
</tr>
<tr>
<td>7</td>
<td>00.53</td>
<td>00.41</td>
<td>01.06</td>
<td>00.46</td>
<td>00.32</td>
<td>00.40</td>
</tr>
<tr>
<td>8</td>
<td>01.52</td>
<td>01.36</td>
<td>01.48</td>
<td>01.40</td>
<td>01.39</td>
<td>01.55</td>
</tr>
<tr>
<td>9</td>
<td>00.58</td>
<td>00.58</td>
<td>01.11</td>
<td>01.22</td>
<td>00.51</td>
<td>00.51</td>
</tr>
<tr>
<td>10</td>
<td>01.05</td>
<td>00.53</td>
<td>01.00</td>
<td>00.55</td>
<td>00.56</td>
<td>00.52</td>
</tr>
</tbody>
</table>

Table 2. All measured data from subjects completing the obstacle course. Specified in minutes.seconds

Survey
What's your name?
This is only used to organize data and won't be mentioned anywhere in the report. This test is anonymous.

Your sex
○ Female
○ Male
○ Other

Age

Have you any previous experience with video-goggles and seeing yourself in real-time?
○ Yes
○ No

How often do you play video-games?
○ Never
○ A couple of times a year
○ Every month
○ Every week
○ Every day

Which view felt easiest?
The one seeing yourself from the side or the one seeing yourself from above?
○ Side-view
○ Top-view
○ No difference

Why did your previous selection feel easier?
Please write why any view felt better or easier

Did you feel dizzy or nauseous during or after the experiment?
○ Yes
○ No

Fig. 6. The form used to gather information from the test subjects
Abstract. Compared to user-selected passwords, computer generated passwords are in general harder to crack, but at the same time difficult to remember. On the other hand, self-selected passwords are easy to crack. The purpose of this study is to examine if users at workplaces with high security requirements prioritize security higher than usability when they select their passwords. Their priorities are compared to a group of users with no or low security requirements. The result shows that most respondents prioritize usability and memorability when they are free to select passwords on their own. Even well security trained users act with their own comfort and memorability as the top priority. They have high confidence in their own password handling, but are still worried about its security risk and threats. However, the security-aware users in the study might be more critical to their own behavior due to higher awareness of security threats and risk.

1 Introduction

Data handled and stored in computers need to be protected against unauthorized users, but at the same time continuously available to those who are authorized. The authorized users need to identify themselves to be given access to the computers, applications and stored data. The purpose of an authentication and access control systems is of course to limit the risks for unauthorized access. Among several and different authentication methods [1], one widely accepted authentication method is a username-password combination method, where passwords either are selected by the user or generated by computers [2] [3].

Passwords, as the first line of defense against unauthorized access, can be more or less weak or strong. Weak passwords can easily be cracked, because they often are short and contain easily guessable data [2]. Strong passwords on the other hand should contain at least eight characters, avoid users or companies actual names, and never be complete word. Strong passwords must be significantly different from previous used passwords, and contain characters from four different categories; uppercase and lowercase letters, numbers and symbols [4] [5]. Authentication methods that allow users to select their own passwords are common but weak from a security point of view [2].
1.1 Background and Previous Studies

A password’s memorability does not automatically match security requirements for authentication. Users given the opportunity to unlimitedly choose their own passwords, tend to choose easily memorable ones such as their own names, street names or other personal information [6]. As a matter a fact, as many as two thirds of all text-based passwords are associated to personal characteristics, such as names or birthdays [7]. Stalling [7] refers to a study by Klein showing that users also choose remarkably short passwords. Another study performed by Spafford [7] confirms the fact that users actually like memorable passwords, when nearly one-fourth of 14 000 observed passwords were possible to guess. Richard E Smith, researcher and information security architect, states in his book Authentication [6], that passwords easy to remember are also easy to guess, which sets focus on the challenge to balance between memorability and security.

Moreover, users nowadays have to handle passwords in many contexts, both at work and in their private life. Memorability becomes more challenging with an increasing amount of contexts asking for passwords or when a password is needed in an application which is only sporadically used [2]. When using several applications or devices, individual users tend to choose and reuse combinations of previous passwords. Each password may in the best of worlds be secure, but if crackers identify the password patterns, these passwords can hardly be considered as strong or safe [8].

Security managers prefer computer generated, arbitrated distributed and strong passwords, while users prefer simplicity and memorable passwords. Proctor et al [2] have by experiments proved that password restrictions aimed to support users to choose strong passwords, actually make it hard to choose acceptable password. As a matter a fact, passwords restrictions may not support its attended goal.

Simplicity to remember seems to go hand in hand with simplicity to guess. Many efforts to improve information security have so far not taken users into account. Proctor et al [2] state that users in general are not aware of the importance of their interaction with systems, they do not act as intended and also find ways to circumvent security procedures. The use of huge amounts of passwords or strong, computer generated ones cause a memory load, but users invent simple ways to solve that kind of problem. Password policies aimed to lead to less predictable passwords may also lead users to write down passwords or make them more averse to changing passwords, because the effort of memorizing new ones [3]. Studies [6] [3] have shown that as many as one out of three users write down their passwords, and some of them keep them under the keyboard or mouse pads.

Humans are considered as a weak link in the security chain, of critical concern is to inform, train and educate in order to ensure security [7]. As a solution, many security managers argue for increased security awareness supported by information, training and education. Companies and organizations invest in so called security awareness programs, as they consider their employees as both part of the security problem and part of its solution [9] [4]. The security awareness
programs are built up by modules of information, training and education, and must continually promote the security message in a variety of ways [7].

1.2 Purpose of the Study

For developers, system requirements on usability and information security are hard to balance, and even contradictory [6]. The question asked is to what extent security awareness influences how users prioritize between usability and security.

The purpose of this study is therefore to examine the relationship between security awareness and password handling, assuming that users with high security requirements at work reach higher levels of security awareness and therefore prioritize security higher than usability.

1.3 Definitions

Memorability is one aspect of usability, and refers to something worth remembering or is easy to remember. Usability is defined in ISO 9241 [10] as “the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments” In other words, a user’s experience of a technology as easy to learn, easy to use and adequate for the task to perform.

Computer security is defined by the National Institute of Standards and Technology (NIST 95) as “the protection afforded to an automated information system in order to attain the applicable objectives if preserving the integrity, availability and confidentially of information systems resources” [7].

2 Method

A questionnaire was used to examine how different users prioritize between security and usability, when they select passwords on their own. The questionnaire was sent to students and professionals working at different workplaces. Based on answers on two specific questions, users were divided into two groups, 1) users with high security requirements at work, 2) and the remaining users that served as a control group. Our criteria for high security requirements are mandatory security training together with smart card authentication solutions. Group 1 was assumed to have a higher level of security awareness, and therefore more prone to prioritize security before usability.

2.1 Questionnaire

The questionnaire contained questions about security training, knowledge of password strength and weakness, password behavior, attitudes and habits, grade of confidence versus risk in own passwords, and self-estimated level of security awareness. Most questions referred to their usage of passwords in general, not restricted to their professional context. A few questions separated between usage in private and professional context. Background facts such as age and gender were
gathered. The information was handled confidentially, and destroyed after being analyzed. The questionnaire was constructed in Google Forms and distributed as a link sent to the respondents by e-mail. The purpose of the study was described as well as instructions of how to fill in and submit the form. The results were processed in Excel, the answers between groups compared and discussed in relation to literature and corresponding studies.

### 2.2 Respondents

The questionnaire was answered by 29 users, 15 women and 14 men. Seven were under 30 years, nine over 50 and the remaining 13 between the age of 30 and 50. The users were assigned to either of two groups. Group 1 contained ten users obligated to participate in security training on a regular basis at work, working in a context where they had to identify themselves using smart cards together with PIN/password. The remaining 19 users served as a control group (group 2).

### 3 Result

#### 3.1 Security Training and Knowledge

<table>
<thead>
<tr>
<th>Group</th>
<th>No. of Users</th>
<th>Security training</th>
<th>Mandatory</th>
<th>Performed last year</th>
<th>Knowledge of weak/strong passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>1+2</td>
<td>29</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 1. Security training and knowledge.

The respondents were asked if they were offered security training at work. If the answer was positive, two more questions followed; a) if the training was mandatory, and b) if they have performed or repeated any security training during the last year.

About half of all respondents (14) were offered security training at work, and for 13 of them (93 %) the training were mandatory. Eight had performed/repeated security training during the last year. All users in group 1 were offered mandatory security training at work, and six had performed it during the last year.

All respondents were asked to describe and give examples of both strong and weak passwords. Regardless of security training, 90 % of all respondents gave correct examples. In group 1, 100 % of the users gave correct explanations.

#### 3.2 Security Awareness

Group 1 rated themselves as more security aware compared to the control group. Within a scale from 1-5, group 1 valued their level of security awareness to 4.5
Table 2. Self-estimated security awareness

compared to 3.26 in group 2. In both groups, men were more confident with their own security awareness than women, see Table 2.

3.3 Password Behaviors, Attitudes and Habits

Group 1 valued themselves as highly security aware. 70 % of them had confident in their information security and password handling. Still, 40 % of users in group 1 admitted that their passwords sometimes contain personal information, such as names, numbers, addresses, name of dogs etc. 70 % sometimes made notes of their passwords, i.e. stored them outside their own memory storage. The same 70 % were aware of risks in their own password handling. See Table 3.

Table 3. Password behavior: awareness in relation to confidence and risks.

As seen in Table 4, 100 % of the users in the security aware group 1 had patterns for how to construct their own passwords. 9 of 10 reuse parts of old passwords when creating a new one, compared to 79 % in group 2. See Table 4.

Table 4. Password habits: password patterns, password reuse, computer-constructed passwords.

Users prefer self-selected passwords. 100 % of the respondents did always or often change a computer-generated passwords into new, self-selected ones. No differences were found between group 1 and 2.
3.4 Usability and Memory

All users in group 1 (100 %) had created their own patterns for how to construct passwords, compared to 63 % in group 2. Still, more users in group 1 than in the group 2 need additional support to remember their password. Note that all users under 30 years were found in group 2.

<table>
<thead>
<tr>
<th>Group</th>
<th>Self-constructed password patterns</th>
<th>Password storage other than memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 %</td>
<td>70 %</td>
</tr>
<tr>
<td>2</td>
<td>63 %</td>
<td>64 %</td>
</tr>
<tr>
<td>1+2</td>
<td>75 %</td>
<td>66 %</td>
</tr>
</tbody>
</table>

Table 5. Use of Self-constructed password patterns and need for password memos.

The respondents were asked about their most important criteria when free to select passwords without restrictions. The question was without any given alternatives, and free for respondents to answer in their own words. All users referred to usability, and nine of them highlighted memorability, expressed in phrases as ‘easy to remember’, or ‘memorable’ or ‘user-friendly’. 18 users described their own password constructions patterns and habits, as a usability matter, and described their use of different characters, password length, reuse of passwords, and passwords related to private information still unknown to others. One respondents had no criteria at all, and another one wrote that the most important criteria was to change passwords now and then. There was no differences between group 1 and 2 when answering this question.

4 Discussion

Text-based passwords are still the most common authentication methods [2], and the passwords can be generated either by computers, or selected by users. In turn, user-selected passwords can either be regulated by security rules within a user is obliged to construct a password, or totally free for a user to selected. All users in this study preferred self-selected passwords, and 100 % of all respondents did always or often change computer-generated passwords into new, self-selected ones. This indicates how important it is for users to create their own passwords, possible to remember and easy to use.

Users in group 1 in this study work at workplaces with high security requirements and with security rules and passwords policies to follow. 100 % of all users in group 1 had developed their own password patterns or personal strategies for how to handle passwords. Detailed and demanding password policies make it difficult for users to create acceptable passwords or to remember them when passwords need to be replaced very often. To cope, users create their own password strategies or patterns [3].

An interesting finding in the study, is that group 1 rated their security awareness higher than group 2, 4.5 to 3.26 on average using five-point scale. Still, they
act similar to group 2, when it came to passwords containing personal information and the need for memos or written lists with passwords. They did also reuse passwords more frequently than users in group 2. Despite this, group 1 have high confidence in their own password handling. A possible explanation is that group 1 had more advanced password patterns, that allowed reuse, and that notes were stored in password protected devices. Users younger than 30 and users in group 2 did not need to write down or store their passwords outside their own memory. Age and memory abilities may correlated, but also the use of several passwords in several contexts, a factor not controlled in this study.

Today’s definition of strong passwords is often implemented in information security rules and tough to users at security training or parts of security awareness programs [2]. However, there are no evidence that password policies support users to select secure and strong passwords. The result can be the opposite, with users who avoid to change passwords and need to have notes under their mouse pads [3]. A study [3] presented at a CHI conference in Vancouver 2011 indicates that longer passwords with few or no restrictions about choice of characters, create the best balance between security and usability.

One can also raise a discussion about value of security awareness programs as well as the value of password policies. Both groups were equally aware of strong and weak passwords. Group 1 seems to feel more confident and secure about how they handle and protect their information, while group 2 have more doubt about their information security and password handling. On the contrary, group 1 valued risks and threats to their passwords and information security higher than group 2, perhaps related to a higher awareness of security risks.

Memorability is important for all users in this study, regardless of gender, age, security requirements or awareness. Even users working in high security environments and with high security awareness prioritize their own usability and a passwords’ memorability before security. Regardless of high security requirements in contexts where valuable or sensitive information is handled, users are humans. Password-composing policies and authentication methods have to take usability into account.

### 4.1 Method Discussion

This study is conducted with a limited group of users, and it is therefore hard to generalize to larger groups of users. Some questions did not separate professional and private use clearly enough. The open question about most important criteria when selecting a password was difficult to interpret and needs to be complemented with personal interviews. Furthermore, asking about people’s personal passwords contains an ethical dilemma, as passwords are considered to be kept secret. Users may have answered questions about behavior as they thought expected, rather than with the actual truth.
5 Conclusions

Users prioritize their own usability, despite high security requirements at work and high self-estimated security awareness. Actually, there seem to be a gap between what a user knows and how a user acts, i.e. between theoretical knowledge and practical behavior. Well trained and security aware users still act with their own comfort and memorability as the top priority. However, self-aware users might be more critical to their own behavior due to a higher risk awareness.

6 Future Work

If users in general prioritize their own usability before security in password handling, it can be of interest to study the contents of self-constructed password patterns, and examine if it helps users to balance between security and usability. Moreover, it can be interesting to examine in what way or to what extent password restriction programs support or prevent users to select passwords that are both memorable and secure. Both self- and security awareness must be considered as background factors that might cause differences between groups of users.

7 Bibliography

References

Do Coloured Numbers Improve Speed and Accuracy When Entering a Numerical Password?

Johan Holmgren
Department of Computing Science
Umeå University, Sweden
id10jhn@cs.umu.se

Abstract. In this paper, two numerical password systems were compared. One password system where the numbers had differently coloured backgrounds, and one where all the numbers had backgrounds of the same colour. A user study was performed to investigate if the password system with coloured backgrounds is quicker and less error prone. Input duration and number of errors were measured in the study, where 20 students from Umeå University participated. Each participant performed five tests with each password system, totalling ten tests. The results showed no statistically significant difference between the two password systems in either input duration or number of errors. From this, no conclusion could be made that the password system with coloured backgrounds is either faster or less error prone.

1 Introduction

Smartphone use is widespread in Sweden, with 65% of the population having used a smartphone to access the Internet outside their home [1]. Smartphones are pocket-sized computers being used for many different things, sometimes also for work. With this large reach, passwords on smartphones are more important than ever.

User authentication is of great importance for confidentiality and keeping private information private. In order to keep both privacy and usability, mobile passwords must be both secure and easy to remember. In an article comparing two alternatives to alphanumeric passwords, Kaplan states that creating and remembering secure alphanumeric passwords is hard [2]. This paper focuses on the usability aspect of the mobile password by examining an alternative method to the common numerical password, i.e. PIN (Personal Identification Number). This is done to test if adding a coloured background to the numbers can affect speed and accuracy of entering the password.

Studies have shown that it appears that perception and working memory are cooperative systems [3]. In another study, subjects were shown a so called target image. This image contained a varying amount of objects, divided into different stimulus classes. Two of the stimulus classes were letters and boxes of colour. All objects in an image shown to a subject were of the same stimulus class. After having been shown the target image, the subjects observed a blank
image in an interval. Following the blank interval, the subjects were shown an array with objects of the same stimulus class as in the target image. They were then asked to identify whether the objects from the target image were present in the array. The class with the highest success rate was the class with boxes of colour. This implies that colours are easier to remember than letters, with short-term memory [4], which could mean that colours can be used to be able to more quickly identify and better remember numbers when they are assigned a colour. Assigning a colour to a number also gives an alternative method of remembering the password. The hypothesis of this paper is that adding a coloured background to the numbers in a numerical password can improve speed and accuracy when entering the PIN. However, if colour can be used to better remember and more quickly identify the numbers, this could be an added advantage.

A user study was conducted in order to test if numbers with coloured backgrounds are quicker and less error prone to enter than numbers with same coloured backgrounds. The number 0 was omitted to make the grid symmetrical. Nine different colours were assigned to the nine numbers, one colour to every number. The numbers were placed randomly in a 3x3 grid to remove any muscle memory component when entering the password. The tests were performed by students at Umeå University in the ages 20-30.

The users in the test were not checked for colour blindness. Only 8% of men and 0.4% of women suffer of colour blindness, or colour vision deficiency [5]. Users were expected to mention any colour vision deficiency when given the instructions and discovering that the test consisted partly of also using colours to identify numbers.

In the study, the length of the PIN was set to four digits with four corresponding colours. This is due to the standard length of a PIN being four digits. This is used for example in bank cards, SIM card and screen lock PINs. Four digits were also chosen in an earlier study, comparing numerical and graphical passwords [6]. The study concluded that graphical passwords are faster than numerical passwords. A different study, also using four digits, comparing numerical passwords to graphical passwords over 21 days [7] concluded that PINs are quicker and less error prone than graphical passwords. Numerical passwords are also more widely used and as numerical passwords are established in today’s society, they warrant further research.

In Section 2, the method used to conduct the study is described with information about the test application and the user study. Section 3 describes the results and calculations based on gathered data. In Section 4, conclusions are made based on analysis of the data. Section 5 discusses the results, limitations of the study and suggested future work.

2 Method

To perform the user study, the numbers one through nine were associated with a colour. The colour assigned to each number is shown in Table 1. In the password
Comparing Same Coloured and Differently Coloured Numerical Passwords

The comparison of the two password systems was performed through an iPhone application developed for this study. Before performing the tests, the users were given instructions on how to perform the tests. Included in the instructions were a PIN (1-2-3-4), shown by images of the buttons used in the tests. The PIN was shown as two different sets of buttons, one set where the numbered buttons had coloured backgrounds and one where the numbered buttons had backgrounds of the same colour. This is illustrated in Figure 2. The same PIN was used in all of the tests.

When the users had memorised the PIN and the colours associated with the numbers, they each performed five tests with both password systems. This means that each user performed a total of ten tests where input speed and number of errors were recorded.

2.1 Application

An iPhone application was created for this study to be able to compare two password systems. One password system with numbers of same-coloured (i.e. grey) backgrounds, and one with numbers of differently coloured backgrounds. In the system where the numbers have coloured backgrounds, nine colours were assigned to the nine numbers according to Table 1. One colour was assigned to a number throughout the tests, i.e. the number 1 did always have a red background in the coloured test.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>2</td>
<td>blue</td>
</tr>
<tr>
<td>3</td>
<td>green</td>
</tr>
<tr>
<td>4</td>
<td>yellow</td>
</tr>
<tr>
<td>5</td>
<td>brown</td>
</tr>
<tr>
<td>6</td>
<td>purple</td>
</tr>
<tr>
<td>7</td>
<td>orange</td>
</tr>
<tr>
<td>8</td>
<td>grey</td>
</tr>
<tr>
<td>9</td>
<td>magenta</td>
</tr>
</tbody>
</table>

Table 1. Table describing which colour is assigned to which number

The password systems were compared in a test where the numbers were placed randomly in a position in a 3x3 grid. The placement of the numbers was stochastic and followed no pattern or model. The users were given a password expressed in both numbers and the assigned colours. The two password systems with randomly placed numbers can be seen in Figure 1.

The application displayed each of the two password systems five times in a random order, meaning every user performed a total of ten tests. Each password system test started when the number grid appeared on the screen and ended with the user entering the last number in the PIN correctly. To test speed and
accuracy, the time to enter the password and the number of errors were recorded. Both the time to enter the password and the number of errors were recorded separately for each test.

**Input duration** To include the whole process of solving the task, the time to enter the password was measured from when the password system was shown on the screen. By doing this, the input duration of a test shows the total time of the user correctly finding and entering all the numbers in the PIN. The time was stopped when the user entered the last number in the PIN correctly.

**Number of errors** While entering the PIN, indicators at the top of the screen showed how many numbers the user correctly had entered, as can be seen at the top of the pictures in Figure 1. This was included to give the user feedback on when numbers were entered correctly or not. If the user made an error while entering a PIN, an error text was shown at the bottom of the screen. The entered numbers were then reset as well as the indicators, meaning the user has to start entering the password from the beginning. In this test, an error was defined as a user not entering the correct number.

(a) Example of numerical password system with randomised number placement and differently coloured backgrounds
(b) Example of numerical password system with randomised number placement and backgrounds of the same colour

**Fig. 1.** The two password systems used in the application. Both images are examples with randomised number placement, the placement of the numbers is randomised in every test.
2.2 User Study

The application was used in a user study with 20 participants. The participants of the study were students at Umeå University in the ages 20-30 with some experience of using smartphones. The test was conducted using the same iPhone, which was provided. The test was also supervised to make sure the test was performed correctly. Each participant performed five tests with both systems, ten tests in total. The entire study consisted of 200 tests.

In the user study, users were given a four digit numerical password and the colours assigned to the numbers. The same PIN was used in all of the tests. The users were given a trivial PIN to remember, i.e. 1-2-3-4. The numbers 1 through 4 had common, different colours in the test where the numbers have coloured backgrounds, compared to for example the similarity between purple and magenta of the numbers 6 and 9. Another reason for choosing the trivial PIN was to get the participants of the study to focus on the different colours instead of remembering numbers.

Since the placement of the numbers is random in the 3x3 grid, every four digit PIN is equally hard to enter. This is due to the placement of the numbers being stochastic. Every test has the same probability of a certain placement of the numbers. The effects of an ordered pattern in one test should be negligible due to the large number of tests performed. By placing the numbers at random, all participants in the study have the same conditions when entering the password.

The password was given with images of the buttons used in the test to enter the numbers. The password was given with both numbers with all grey background colours and numbers with different background colours, as can be seen in Figure 2. The users were given instructions that the PIN was shown two times in the image, one for each test, and that the PIN should be read from left to right. The users were given the time they needed to memorise the password and the colours. The test consisted of the two different password systems appearing five times each in a random order. During the test, the input duration and number of errors were measured.
2.3 Data Assumptions

The similarity between the two password systems means that a two sample t-test [8] can be performed to determine whether the differences of the two different password systems are statistically significant in relation to input duration and number of errors. However, to perform the two sample t-test the samples must be independent and normally distributed.

The central limit theorem [8] states that a large number of independent random variables have a sum that converges to a normal distribution. This is true even if the variables have different probability distributions, as long as they have finite variances. The samples of the study can be assumed to be independent since the tests were performed individually. Regardless of the shape of the population, the normal approximation of the central limit theorem will be good if \( n > 30 \). This study contains 200 samples and can therefore be assumed to be normally distributed according to the central limit theorem [8].

Having shown that the samples are independent and normally distributed, a two sample t-test can be used to examine the null hypothesis (\( h_0 \)) that the means (\( \mu \)) of the two different password systems are equal. Our null hypothesis will thus be \( h_0 : \mu_c = \mu_w \), where c stands for colour and w stands for without colour. A confidence interval of 95% was chosen, which means that if the two sample t-test generates a statistically significant result we can reject the null hypothesis and with 95% percent certainty say that one mean differs from the other in a non-random way.

**Input duration** In the first two sample t-test, the t-test is used to investigate the null hypothesis that the means of the input duration between the two password systems differ in a non-random way. That is, \( h_0 : \mu_c = \mu_w \) with respect to the duration of input in both password systems. To get a statistically significant result, the null hypothesis must be rejected.

**Number of errors** The second two sample t-test is used to examine the same null hypothesis in regard to the number of errors while performing the tests. \( h_0 : \mu_c = \mu_w \) in regard to the number of errors when entering the numerical password. Also in this test, the null hypothesis must be rejected to get a statistically significant result.

The two sample t-tests were performed by using Minitab, which is a tool for statistical analysis. Minitab was also used to generate a plot of the distribution of the input duration in the two password systems.

3 Result

Having met the conditions of independent and normally distributed samples, two different two sample t-tests were performed. One in relation to input duration and one in relation to number of errors by the users.
Comparing Same Coloured and Differently Coloured Numerical Passwords

**Input duration** In the first two sample t-test, the null hypothesis is $h_0 : \mu_c = \mu_w$ with respect to the duration of input in both password systems. After choosing a confidence interval of 95%, the p-value that the test results in must be less than or equal to 0.05 ($p \leq 0.05$) for the test to be statistically significant.

![Scatter plot of the input durations of the two password systems. Mean values are marked with black lines.](image)

**Fig. 3.** Scatter plot of the input durations of the two password systems. Mean values are marked with black lines.

Figure 3 shows the input durations of the two password systems. As can be seen in Table 2, the mean value of the password system with colour was higher than the password system without colour. This means that the password system with colour was slower than the password system without colour. The test results in a p-value of 0.3754, meaning that there is not a significant difference between the two password systems.

<table>
<thead>
<tr>
<th>$\mu_c$</th>
<th>$\mu_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8390</td>
<td>3.5185</td>
</tr>
</tbody>
</table>

**Table 2.** Table describing the mean input durations of the two password systems

**Number of errors** The second two sample t-test has the null hypothesis $h_0 : \mu_c = \mu_w$ with respect to the number of errors by the user in both password systems. By using the same confidence interval as the first test of 95%, also here the obtained p-value must be less than or equal to 0.05 ($p \leq 0.05$) for the test to be statistically significant.

The second test results in $p = 0.5306$, meaning that also in this test there is no statistically significant difference between the two password systems.
Table 3. Table describing the mean number of errors of the two password systems

By observing the mean number of errors in Table 3, we can see that very few errors were made in both password systems.

4 Conclusion

Given that the conditions of independent and normally distributed samples were met, two separate two sample t-tests were performed. From the results of these tests we can draw the following conclusions.

**Input duration** The results of the first two sample t-test showed no statistically significant difference between the two password systems in relation to input duration. The conclusion that the password system with colours is faster cannot be made. From this, we cannot confirm the hypothesis that the password system with colour is faster. In this test, the password system with colours was slower than the password system without colours.

**Number of errors** The second two sample t-test resulted in no statistically significant difference between the two password systems in relation to the number of errors performed by the user. We cannot draw the conclusion that the password system with colours is less error prone than the password system without colour. The hypothesis that the password system with colours is less error prone cannot be confirmed.

In summary, neither the hypothesis that the password system with colours is quicker nor the hypothesis that the password system with colours is less error prone can be confirmed.

5 Discussion

Even though users were instructed to memorise the colours, there is no guarantee that the users identified the numbers using colour when possible. User feedback after the test indicated that some users only looked for the numbers in all tests, even though instructed to also use colour when possible. This may have affected the study and more clear instructions may have given a different result.

Similarly, the low amount of user errors indicated that many users focused on entering the right number instead of entering the PIN as quickly as possible. This may also have affected the result, with different instructions perhaps changing the result of the study.

Since the numbers were placed randomly in a 3x3 grid, there is a possibility that some tests were easier to perform than others. If the numbers are placed
Comparing Same Coloured and Differently Coloured Numerical Passwords

where the user first looks for them, entering the PIN is quicker. This should not affect the results of the study. The numbers are placed randomly in all tests and with a sample size this large, both password systems should be affected equally.

Similarly, the random selection of which password system was shown made it possible for users to be shown the same password system multiple times in a row. When studying the input durations of the test, there may be a possibility that the individual tests were affected by this. Also in this case, with the large sample size, this should affect both password systems equally.

Finally, there is a possibility that the users having previously used numerical password systems without colour may have affected the study. The password systems with colours may be initially confusing. By first giving the users an introduction to the coloured password system, the results may have changed.

5.1 Limitations

Colours can be perceived differently by different people, which can have an effect on the study. For example, colour blind people may have difficulty distinguishing certain colours which can affect the study. Differently perceived colours may affect the input duration and number of error when a user enters the PIN.

Since numerical passwords are widely used, the participants in the study probably have experience of entering a numerical password. Numerical passwords with same-coloured backgrounds are common and this may result in the two password systems not having equal conditions.

By choosing predefined colours when creating the iPhone app, the options were limited. When avoiding white and black background colours, options were few and as a result grey and magenta were chosen. Grey may not be as easily distinguishable as the other colours, where magenta is similar to purple. This could have affected the study, where more easily distinguishable colours could have been more distracting to the users.

5.2 Future Work

In this study, no conclusions could be drawn about a coloured numerical password system being faster than a numerical password system without colours. However, the result could be affected by users having previously seen only same coloured number grids before. An interesting future study would be to let users get accustomed to the coloured password system before performing a similar test.

Security is an important area in mobile passwords. Using a password system with colours, like the one used in this test, could be effective against smudge attacks\(^1\) [9]. By placing the numbers at random, the smudges on the screen can not be used to gather the password. However, other attacks like shoulder surfing\(^2\)

---

\(^1\) An attacker being able to recover a password by finger oil left on the screen
\(^2\) An attacker gathering the password by observing the user enter it, possibly over the shoulder
could be made easier. A future study could be made about the effects of coloured numerical passwords on security.

6 Acknowledgments

We would like to thank Mikaela Berg for providing the initial idea for this paper with her submission in last year’s conference. We would also like to thank the subjects in the user study for participating. Finally, we would like to thank the peer-review group for helping refine the idea and improving the paper by providing valuable feedback.

References

Can a High Color Contrast Touch Interface Increase User Reaction Time when Using a Smart Phone Web Based Application?

Albin Hübsch
Department of Computing Science
Umeå University, Sweden
albin.hubsch@gmail.com
http://albinhubsch.se

Abstract. This study aims to investigate if it is possible to receive better user performance when using a high-color-contrast compared to a low-color-contrast touch screen interface in a mobile web application. The performance was measured by combining the time each user took to respond to instructions (reaction time) and the number of input errors the user made. Received test data appeared to be corrupt, most probably due to limitations in web application technology, despite this our result shows that the low color contrast interface produced 475% more user input errors compared to the high color contrast interface. No difference in reaction time could be found between the two versions.

1 Introduction

The main goal with this paper is to evaluate if color contrast has a significant impact on usability in web based smart-phone applications. Consumer touch screen devices such as smart-phones have rapidly increased in number and availability recent years. The touch screen technology has made great advances [1] and it is frequently used as a way of receiving user input. When this technology moves to a broader audience, higher demands on usability needs to be set [2]. To further explore how usability can be improved using touch screen technology we have in this paper investigated if reaction time (the time it takes for a user to react on instructions, further referred to as RT in this paper) and user input errors (the number of user input errors when interacting with a system, further referred to as UIE in this paper) can reach better results by increasing color contrast within the interface. With “better results” we intend to mean fewer UIEs and faster RTs. Increased number of user input errors means that users repeatedly presses the wrong buttons which tells us that it is hard to accomplish desired tasks. Faster reaction times does not necessarily mean “better”, but in this experiment we had tasks that should be performed as fast as possible which led us to the conclusion, faster reaction times are “better”.

Contrast is determined by the difference in color and brightness of an object and objects within the same field of view. High contrast is the degree of difference
between dark and light where it approaches the maximum possible difference. Low contrast is the opposite, low degree of difference between dark light.

Earlier studies completed in this field have explored how we perceive different color contrast and how it can affect our reading performance [3]. It has also been proved that chromaticity, contrast, and cone opponency in color space can affect RTs [4]. Many best practices for mobile development have also shown that “requirements for sufficient color contrast” [5] must be met to best amplify the content. Still the question remains, can a high color contrast interface increase the reaction time and decrease the amount of user input errors?

High RTs and low UIEs are especially desirable when designing user interfaces for situations with high demands on quick user input and low error tolerance such as in emergency situations. The results of studies like this are especially useful for designers and front end developers.

2 Method

In order to be able to test if color contrast has a substantial impact on the RT and UIE of a touch screen user interface we have designed a simple web application for an Android smart-phone. The application exists in two versions, one with low color contrast (referred to as LCC in this paper) and one with high color contrast (referred to as HCC in this paper) interface as can be seen in Fig 1. We conducted an A/B test to measure the performance difference between the versions.

We designed the application by following design guidelines introduced and presented here [6, 7, 2, 8]. These guidelines can be seen as a set of rules to follow when designing any kind of user interface. We used guidelines to get proper sized buttons, good spacing between content and best positioning of navigation.

2.1 Designing the A/B test

We conducted an A/B test to measure the differences in RT and UIE between the two versions. An A/B test is a commonly used method by developers and designers to test differences in performance between applications when the differentiated factors are known [7]. Where in this case our known factor is the contrast difference.

The application consists of 4 buttons, each button representing a function. To differentiate each button from the others we put a unique shape in each button. The shapes we used were a, square, circle, triangle and a rhomb (Fig 1).

To be able to give the test persons consistent instructions through the complete test we designed a program that presented the instructions on a secondary screen. The instructions were a series of 20 instructions with shapes identical to the ones in the application buttons. The test persons were told to press the representative shape in the application as fast as they could in order to measure

\[1 \text{ On-screen button}\]
Can High Color Contrast Increase User Reaction Time

Fig. 1. Our two versions of the application designed with different contrast. To the left the low contrast interface and to the right the high contrast interface. The application and its interface was designed specifically for this test. The bottom menu was added to give the interface more of a complete application feel.

Fig. 2. Our program designed to present user instructions. Instructions were given in a series of 20. Between each instruction there was a delay set to be between 2.5 and 4 seconds. Each time a new symbol was showed a timestamp were logged in a unique user instruction file.
the reaction time. If they realized that they pressed the wrong button they were just told to continue the test by pressing the right button as fast as they realized. The application registered all wrong, non-matching input as an UIE. Each test was performed indoors with varying surroundings such as different lightning conditions and noise levels.

We tested each version of the application on five persons, all within the same age group (20-30) and with a variety of backgrounds. In total we got 10 test results with a gender split of 50% women and 50% men that grouped into LCC and HCC. We analyzed them against each other to measure if any differences in RT or UIE between the two versions could be statistically confirmed.

2.2 Evaluation of the A/B test

Our data received from the tests consisted of UNIX timestamps measured in milliseconds. Each time the user interacted with the application it registered which button that was pressed and saved it together with a timestamp. Equally, every time the instruction program showed a new symbol it logged the symbol together with a timestamp. As a result we received two data sets of symbols with corresponding timestamps for each test person, resulting in a total of 200 records. Although the data was hard to interpret due to its compact look, a python script was written to structure the resulting timestamps into a more readable format. We calculated each RT by subtracting the timestamp when the instruction was given from the registered user input timestamp. Out of this we could calculate the mean RT for both groups LCC and HCC. The calculation of RT did not take in consideration if user input was a match with the instruction or not. It was still considered as a reaction.

UIEs were counted every time the user pressed a non matching object compared to the one given in the instructions. All input errors in each group, LCC and HCC, were counted to see if one of the groups produced more UIEs.

3 Result

From our results we can not draw any statistically significant conclusions about differences in reaction time between the two groups high color contrast and low color contrast. This is most probably due to corrupt test data, a result from limitations in web application technology, see Section 4. With user input errors we found that the low color contrast group produced significantly more user input errors than the hight color contrast group.

3.1 Reaction Times (RT)

Table 1 shows the average reaction time and standard deviation for each group. According to the results people in the high color contrast group had insignificant slower reaction times compared to the ones in the low color contrast group.

---

2 More information at: http://www.unixtimestamp.com/
Can High Color Contrast Increase User Reaction Time

Table 1. Average reaction time and standard deviation for each group, low color contrast and high color contrast. A two tailed t-test was performed on the two groups and showed that no significant difference could be determined between them.

<table>
<thead>
<tr>
<th></th>
<th>Average RT</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCC</td>
<td>783 ms</td>
<td>969 ms</td>
</tr>
<tr>
<td>LCC</td>
<td>550 ms</td>
<td>1393 ms</td>
</tr>
</tbody>
</table>

To check if the RTs in the two groups are significantly different we assumed the two data sets where normal distributed based on the $n>30$ preference, and performed a two tailed t-test. The test produced a p-value of 0.17 which is bigger than a desired error margin ($\alpha$) of 5%. This shows that the difference in RT between the two groups is not statistically significant. Regardless what the t-test says our standard deviations are indicating that something might be wrong with our data (see section 4).

3.2 Input Errors (IE)

None of the groups where completely free from user input errors. As can be seen in Table 2, users that were given the LCC interface produced 475% more UIE than the HCC group.

<table>
<thead>
<tr>
<th></th>
<th>Errors</th>
<th>Standard Deviation</th>
<th>Errors/User</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCC</td>
<td>4</td>
<td>1.10</td>
<td>0.8</td>
</tr>
<tr>
<td>LCC</td>
<td>19</td>
<td>3.49</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Table 2. Number of user input errors registered in each group and a calculation of input errors per user in each group. The low color contrast interface produced 475% more input errors compared to the high color contrast version. A two tailed t-test also showed that there is a significant difference between the two versions. A high color contrast interface can therefore be a preferred design method if a low amount of user input error is desired.

Table 2 shows us that the users in the HCC group generally had a more stable and consistent performance compared to the LCC group. We also performed a two tailed t-test on the UIE data set. The outcome of this test was a p-value of 0.001 which is significantly smaller than a $\alpha$ of 5% telling us that the HCC group performed better.

4 Discussion

From our test results we cannot draw any conclusions about differences in reaction time between the high color contrast interface and the low color contrast interface. However we could see that the high color contrast interface was more reliable in performance when it comes to the amount of user input errors, according to our results.

What we have seen in our collected test data can strongly negotiate the legibility of the results. The calculated standard deviations gave us hints on
that something was wrong. We dug deeper into each test persons results and found that the data must be corrupt. We found that we had randomly received negative RTs which in theory should be completely impossible. We were prepared that the timestamps could be off sync due to individual system clocks and no boot up pairing. However this would only have resulted in a consistent off sync. Instead we got inconsistent hiccups in our timestamps. We suspect this is a result from either hardware limitations or limitations in the JavaScript engine for web applications on Android.

Although we can not draw any conclusions out from the results we have learned that smart phone web based applications should only be used if sometimes slow technology, user input errors and long reaction times from the user can be accepted. The HCC interface could also be seen as a preferred design method, even if the data was corrupt, because the users performed more consistent with UIEs compared to the users in the LCC group, see number of UIEs and their standard deviations in 3.2.

4.1 Drawbacks and Limitations

The main drawback of this study is the corrupt data set. This has to be in mind when we draw any conclusions out of the results. The problem could most probably be solved by simply rerunning the complete test using a native smart phone application instead of a web based solution.

Each test person performed only one test with a supervisory selected interface, the LCC or HCC version. To eliminate possible affecting surrounding factors each person should have done the test several times and with both the LCC and HCC interfaces. This would also have made it possible to recognize if the test persons memorized the patterns in any way.

Small test group. To be able to get a more statistical valid result the test group has to be bigger. A good rule of thumb is at least 30.

Many affecting parameters. The prototype is simple in its appearance but there are still many parameters that can affect the users interaction and their RTs. Icons, icon size, optimal number of on screen buttons, size of hand-held device etc. All this, and more, should have been researched and taken into account if done again.

Limited test environment. Our tests were made using an Android based smart-phone and the application was made using HTML5, CSS3 and JavaScript. We can not surely imply, without testing, that we would get the same results on an iPhone or any other smart-phone, or even with a native application.

---

3 A prototype based dynamic scripting language
4 OnePlus Two https://oneplus.net/2
5 HyperText Markup Language, markup standard for WWW
6 Cascading Style Sheet, styling standard for WWW
4.2 Future Work

Due to the failed data set that we received in this paper there are things that could be improved or continued to be worked on. We do have some concrete suggestions that could be case for future work.

- Perform tests using a native application.
- Extend the test group with more test subjects. It is possible to think that the patterns that slightly appeared in our results will appear even more significant with a larger group of test subjects.
- Our collected test data can be downloaded\(^7\) and used freely to investigate other aspects not mentioned in this paper. One proposal is to look at the different shapes and try to detect possible error patterns between them. Is the rhomb more frequently a case for IE compared to the other shapes?
- Do our results apply on other systems and techniques? Our tests are limited to the Android system and a web based application solution. It remains to answer if our results applies on all other systems. This is also a way of isolating some of the affecting parameters mentioned in Section 4.1.

5 Acknowledgments

The author would like to thank the peer reviewers who provided valuable comments on both the content and structure of this paper.

References


\(^7\) Download here https://goo.gl/x8URSB
Context-Free Graph Grammars for Recognition of Abstract Meaning Representations

Anna Jonsson
Department of Computing Science
Umeå University, Sweden
aj@cs.umu.se

Abstract. We attempt to recognise a corpus of semantic representations of natural language sentences on a limited domain using hyperedge replacement grammars. The semantic representations used are abstract meaning representations, which are effectively directed graphs. The results show that 88.8% of the graphs in the corpus are recognised, which indicates that hyperedge replacement grammars can be a suitable formalism for recognising abstract meaning representations.

1 Introduction

Methods for structuring semantic data are constantly being developed for usage in various computer science applications with the common goal to make computers able to communicate in natural language [1] [2] [3]. The abstract meaning representation (AMR), first defined in [4], is a semantic representation of a natural language sentence and is therefore useful within the field of natural language processing. Formally, an AMR is a directed, rooted, acyclic, labelled graph. It is thoroughly described in [5] and in the more updated GitHub repository.

The aim of previous and ongoing work has been and still is to determine a suitable formalism that can be used in order to describe, recognise, parse and generate AMRs. Graph structures similar to, but slightly more general than AMRs have successfully been used as an intermediate step in machine translation in [6], where it is also argued that a purely semantic representation is needed. We shall note that AMRs cannot be considered an interlingua [7] and that they are biased towards English [5].

The first parsing effort for turning natural language sentences into AMRs, so called string-to-AMR parsing, was done by Flanigan et al. [8]. Their method consists of, simply put, searching for the maximum spanning connected subgraphs in the graph formed by adding all possible nodes (representing the words) and edges (representing the relations between the words) based on the input string. The approach of Peng et al. [9] to the same problem, i.e, string-to-AMR parsing, was to train a hyperedge replacement grammar (HRG) whereas Pust et al. [10] modify a syntax-based machine translation system and Artzi et al. [11] use combinatory categorial grammars.

1 https://github.com/amrisi/amr-guidelines/blob/master/amr.md

Thus, there are several formalisms that can be used for capturing, at minimum, the string-to-AMR parsing. However, in our investigation, the focus is solely on the recognition of AMRs, i.e., the act of verifying correct AMRs and rejecting incorrect ones where the conditions for correctness are formulated in terms of the formalism. The formalism we consider is graph grammars, and, in particular, HRGs. An HRG [12] is a context-free graph grammar, i.e., it describes a language (a set) of graphs as opposed to the usual string languages described by context-free grammars. The difference is that the rewrite rules of an HRG work by exchanging hyperedges for hypergraphs instead of exchanging non-terminals for strings. A hyperedge is an edge than can be connected to more than two nodes and a hypergraph is a graph that can contain hyperedges. This means that an HRG in fact describes a language of hypergraphs rather than ordinary graphs.

The motivation for restricting ourselves to HRGs and investigating only the recognition of AMRs comes from Drewes et al. [13] in which the authors give a method for predictive top-down (PTD) parsing for a restricted type of HRGs. The parsing algorithm takes an HRG and a hypergraph as input and returns the answer to whether or not the hypergraph can be generated by the HRG. Moreover, they have, analogously with the so called boy-girl corpus generated by Knight and colleagues and presented in Braune et al. [14], created an HRG for recognising AMRs over the same domain as the corpus.

The aim of this study is to move towards the understanding of the necessary properties of the sought-for formalism for handling AMRs through an evaluation of the use of HRGs given in [13] for recognising AMRs given in [14]. We want to be able to tell whether or not the HRG formalism is powerful enough to recognise the AMRs from the boy-girl corpus. This yields the broader question: Can HRGs describe AMR corpora that are restricted to a specific domain?

The evaluation is done by running the implementation of the PTD parser using the HRG developed for the same domain as the boy-girl corpus, both provided by the authors of [13], on the AMR in the boy-girl corpus generated by Knight and colleagues in order to find out whether or not the HRG defines the same language as the corpus. The results show that the grammar does in fact recognise 88.8% of the boy-girl corpus. This calls for further investigation of the usage of HRGs for the recognition of AMRs done on other corpora and with other HRGs in order to find the cases that can not be covered by this formalism, if any.

The remainder of the paper is structured as follows: In Sect. 2 we give some basic terminology, and the main concepts, i.e., AMRs, HRGs and PTD parsing are explained in Sects. 3, 4 and 5, respectively. In Sect. 6 the method used for recognising the AMRs given an HRG is described and in Sect. 7 the results are presented. Finally, conclusion and future work are discussed in Sect. 8.
2 Preliminaries

This section contains definitions that lay the theoretical ground for the rest of the paper.

2.1 Formal Languages

Natural languages, such as English and Spanish, are not easily described formally since there are usually more exceptions than rules. Formal languages are on the other hand defined formally, as we will see, which makes them less complex than natural languages and easier to handle by a computer. In this paper, the term languages will refer to formal languages. In order to provide a proper definition of languages, we first introduce some basic concepts and terminology and then use them in the definition itself, following [15].

An alphabet $\Sigma$ is a non-empty, finite set of symbols. $\Sigma^*$ denotes the set created by combining the symbols in $\Sigma$ in all possible ways. For example, if $\Sigma = \{0, 1\}$, we get the set of strings $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$ (note that picking no symbol at all yields the empty string $\varepsilon$). A language $L$ over an alphabet $\Sigma$ is a subset of $\Sigma^*$. The elements in $L$ are known as words. For example, the language of strings over $\Sigma$ from our previous example that start with a zero is $L_0 = \{0, 00, 01, 000, 001, 010, 011, \ldots\}$. Since we in this case build words that are strings, $L_0$ is an example of a string language.

2.2 Grammars

In this section we recall the concept of grammars in the context of string languages since string grammars are less technical but follow the same pattern as HRGs, which will be explained in detail in Sect. 4.

A language can be defined by a grammar $GR$ and is then denoted by $L(GR)$. The definition of a grammar follows.

Definition 1 (Grammar [15]). A grammar is a 4-tuple $(N, T, R, S)$ where

- $N$ is a non-empty, finite set of non-terminals, which can be seen as the variables of the grammar, and they will not appear in the words of the language,
- $T$, for which $T \cap N = \emptyset$, is a non-empty, finite set of terminals (the alphabet),
- $R$ are the rewrite rules consisting of a left-hand side from $N$ that is separated with ::= from a right-hand side that consists of a string from $(N \cup T)^*$ (the empty string $\varepsilon$ is also allowed), and
- $S \in N$ is the start symbol.

In particular, the grammar defined above is a context-free grammar. These grammars are called context-free since there is no restriction on the non-terminals such that they have to appear in a certain context (e.g. always with a terminal on
\[ N = \{S, A, B\} \]
\[ T = \{a, b\} \]
\[ R = \{ S ::= A | B | \varepsilon, \quad A ::= AA | a, \quad B ::= BB | b \}. \]

**Fig. 1.** An example of a context-free grammar \( GR \).

\[ S \Rightarrow A \Rightarrow AA \Rightarrow AAA \Rightarrow AAa \Rightarrow aAa \Rightarrow aaa \]

**Fig. 2.** An example derivation for the grammar \( GR \) in Fig. 1. First, the rule \( S ::= A \) is used. Next, \( A ::= AA \) is used twice. Finally, \( A ::= a \) is used until there are no more non-terminals in the string (three times).

its right or left as is the case with regular grammars). An example of a context-free grammar can be seen in Fig. 1 in which \( | \) separates different right-hand sides when several rules share the same left-hand side.

A derivation from \( S \) to any word in the language defined by \( GR \) is done by iteratively matching any non-terminal \( A \) to the left-hand side of a rule in \( R \) and replacing \( A \) with the right-hand side of the rule until there are no more non-terminals to replace. Thus, the result of any derivation consists of only terminals (or the empty string) and is defined to be a word in \( L(GR) \). The usage of a rewrite rule in a derivation step is denoted by \( \Rightarrow \). An example of an entire derivation for the grammar \( GR \) in Fig. 1 can be found in Fig. 2. We see that the terminal string derived is \( aaa \). Thus, \( aaa \) is a word in \( L(GR) \).

A string language is said to be context-free if its words correspond to all possible derivations of a context-free grammar.

### 2.3 Parsing

The term parsing (as defined in [15]) in the context of a grammar \( GR \) refers to the act of finding a derivation for a word \( w \) using \( GR \). If such a derivation exists, we conclude that \( w \in L(GR) \), and we say that \( GR \) generates \( w \). Otherwise, \( w \) is not in \( L(GR) \) and is therefore discarded.

The parsing is done by testing all possible derivations that can be done from the start symbol \( S \) in \( GR \). Whenever a word is produced (that is, when no non-terminals remain), it is compared to \( w \). If they match, we know that there is such a derivation. Otherwise, backtracking is used, i.e., we go back in the derivation chain until we encounter a non-terminal for which we could have used a different rule. Then the alternative rule is used instead and the derivation is continued until another word is produced. This goes on until every possible combination of rules has been tried. If no derivation was found, we know that \( w \notin L(GR) \).
Since we start with only $S$ and build the word as we go, this type of parsing is referred to as *top-down parsing*. Its counter-part, *bottom-down parsing*, would have started out with $w$ and tried to match rules covering all parts of $w$ and combined them in a backwards manner until only $S$ would remain.

### 2.4 Directed Graphs and Hypergraphs

A *directed graph* (found in, e.g., [16]) is a pair $G = (V, E)$ where $V$ is the set of *nodes* of the graph and $E$ contains the directed *edges* in the form of pairs of nodes $(s, t)$ where $s$ is the source node and $t$ is the target node of the edge, i.e., the edge goes from $s$ to $t$. Both the nodes and the edges can be labelled by adding labelling functions that assign labels to the components of the graph. For example, the function $\text{label} : V \rightarrow \mathbb{N}$ labels the nodes of the graph with natural numbers, and a graph with node labels would then be defined as $G_{\text{labelled}} = (V, E, \text{label})$. $G$ is *rooted* if there is a node that does not have any incoming edges. The *leaf nodes* of $G$ are the nodes with no outgoing edges, and the *non-leaf nodes* are the nodes that have at least one outgoing edge.

The generalisation of a graph $G = (V, E)$ by replacing the edges $E$ with a set $H$ of *hyperedges* yields a *hypergraph* [17] [12] [13]. A hyperedge is an atomic item that has a fixed number of *attachment points* ordered in a certain sequence controlled by the hyperedge. This number is referred to as the *type* of the hyperedge. An example of a labelled hyperedge can be found in Fig. 3 in which the black nodes along with the dotted lines are attachment points and the square holds the label of the hyperedge. A hyperedge can be attached to a hypergraph if the number of *external nodes* (attachment points) of the hypergraph is equal to the type of the hyperedge. Thus, when part of a hypergraph, a hyperedge can be defined as the sequence of nodes of the graph to which the hyperedge is attached. As we will see in Sect. 4, the operations of attaching hyperedges to, and detaching them from hypergraphs is central for HRGs.

![Fig. 3. A labelled hyperedge of type 4.](image)

Note that a (directed) graph is simply a (directed) hypergraph where all hyperedges are of type two.

### 3 Abstract Meaning Representations

An AMR [5] is a directed, rooted, acyclic, labelled graph. The nodes in the graph represent so called *concepts*, and the edges represent *relations* between the
concepts. The non-leaf nodes are also considered to be events, and the relations define the nature of the connection between the event and the other concepts and events. In particular, the special relations arg0 and arg1 are used, and their definitions are as in PropBank [18] [19]. For all events presented in this paper, this makes arg0 point at the agent (the performer of the action expressed by a verb) and arg1 at the patient (the target of the action). For example, the relation arg0 for the concept want defines the wanter and the relation arg1 gives the wantee. An example of an AMR can be found in Fig. 4.

Since the idea behind the AMR formalism is to represent the semantics of an event, an AMR can often be realised into a natural language in more than one way. For example, the AMR in Fig. 4 can be parsed into several English sentences of which two are “The boy wants the girl to believe him” and “It is wanted of the boy that the girl has belief in him”.

![Fig. 4](image-url) An AMR representing the event want, where the wanter is the concept boy and the wantee is the event believe for which the believer is the concept girl and the believee is the formerly mentioned concept boy.

### 4 Hyperedge Replacement Grammars

An HRG [12] is a context-free graph grammar that by replacement of hyperedges can define languages of any structure containing hyperedges. Here, the only such structures we look at are hypergraphs, which is why the definition is formulated in terms of them rather than a more general structure. HRGs are considered context-free since the replacement of a hyperedge does not change its surroundings.

An HRG is a grammar \( HRG = (N, T, R, S) \) in which both \( N \) and \( T \) are hyperedge labels and \( S \) is the start symbol, which is also a label embedded in a hyperedge. Hyperedges with type 1 are considered node labels and are in this paper always terminal. The rules in \( R \) are defined with a left-hand side consisting of a hyperedge with a non-terminal label and with a resulting hypergraph in which the hyperedges can be labelled from either \( N \) or \( T \) or with \( \varepsilon \) as the right-hand side. The number of attachment points in the right-hand side has to be
the same as for the left-hand side. More intuitively, the hyperedges labelled with non-terminals are used as placeholders that can be replaced by a hypergraph. 

HRG defines the hypergraph language \( L(HRG) \).

**Fig. 5.** An example of an HRG consisting of six rules, each numbered by a letter.

An example of an HRG can be seen in Fig. 5. In order to illustrate the rules, we use black dots connected with a dotted line as attachment points, circles to represent the nodes and squares for non-terminal hyperedge labels. Terminal hyperedge labels are portrayed as plain text, and we let the hypergraphs on the right-hand sides of the rules contain directed edges (represented with arrows) with terminal labels. For simplicity, hyperedges of type 1, i.e., node labels, are written inside the node circles rather than in a separate hyperedge with a terminal label connected to the node. The order of the attachment points is not explicitly portrayed using numbers as in Fig. 3 but is implied by the positions of the attachment points. Note that the start symbol \( S \) is of type 0, i.e., it has no attachment points and can therefore only be replaced by a hypergraph without any attachment points. To clarify how the rules are used, an example derivation is depicted in Fig. 6. Above each \( \Rightarrow \) indicating a single derivation step, a letter corresponding to the rule from the HRG in Fig. 5 used in that particular step is written.

### 5 Predictive Top-Down Parsing

Parsing of hypergraphs is not as straight-forward as the parsing of strings explained in Sect. 2.3 since some of the hypergraph languages are NP-complete [20].

Drewes et al. present in [13] a parsing algorithm for HRGs which the authors call *PTD parsing*. It is an extension of the usual top-down parsing for grammars that first checks whether or not the input grammar \( HRG \) is *PTD parsable*. PTD
Fig. 6. The derivation of the AMR with meaning “the girl believes the boy” from the HRG in Fig. 5.

parsability implies that the HRG can be parsed in quadratic time [13] whereas the general HRG parsability problem is, as previously mentioned, NP-complete. Then the algorithm builds a parser $P_{HRG}$ that can be used to decide whether or not an input hypergraph $G$ is in $L(HRG)$. The two steps are in fact done in parallel using a complex analysis of the grammar which effectively builds the parser in a number of steps producing partial results as it goes. The partial results allow for the checking of a number of conditions that are also defined in [13] that need to be fulfilled in order for $HRG$ to be PTD parsable.

6 Method

In order to find out whether or not HRGs could be a suitable formalism to use when it comes to describing AMRs, we want to conduct an initial case study that investigates if it is possible to describe a set of AMRs on a given domain using an HRG. In particular, we attempt to get all AMRs in an already existing corpus successfully parsed by an HRG using a PTD parser. Below, the AMR corpus used is presented followed by an explanation of the manner in which the recognition of the AMRs in the corpus using an HRG was done.

6.1 The AMR Corpus

The boy-girl AMR corpus from [14] was chosen since its domain is restricted to few concepts and relations which helps limiting the possible size of an HRG describing it. It is therefore suitable for a smaller experiment such as this.

The corpus consists of a graph file containing 10000 AMRs over the domain of the concepts (node labels) girl, boy, want and believe and the relations (terminal edge labels) arg0 and arg1 (which are as used in Sect. 3). Moreover,
at most one occurrence of each of the concepts girl and boy is allowed in each AMR.

6.2 Recognition of AMRs Using HRGs

In order to recognise the AMRs in the boy-girl corpus, we exploited the fact that they are directed graphs with both node and edge labels by simply observing that they are thereby also node and edge labelled hypergraphs. Being able to treat AMRs as hypergraphs is wanted since the PTD parsing algorithm works on hypergraphs. Thus, we could simply feed the PTD parser the already existing HRG that is defined on the same domain as the boy-girl corpus, let us call it $HRG_{b,g}$, and get as a result a parser $P_{HRG_{b,g}}$ that recognises the, in this case, AMR language defined by $HRG_{b,g}$. Both $HRG_{b,g}$ and the PTD parser used were retrieved by courtesy of the authors of [13].

Since the AMRs in the boy-girl corpus are not in the same format as the input AMRs to the PTD parser, a small program for converting from the corpus format to the PTD parser format was implemented and the AMRs were converted. Finally, $P_{HRG_{b,g}}$ was run on all AMRs in the boy-girl corpus, and the AMRs not recognised by $P_{HRG_{b,g}}$ were recorded.

7 Results

Out of the 10000 AMRs provided in the boy-girl corpus, 8880 were recognised. All of the 1120 graphs that were not recognised contained the subgraphs in Fig. 7, i.e., the concepts want and believe without any arguments. They failed because there are no rules in the HRG that allow these subgraphs to be used as leaves of the graphs. The subgraphs are in themselves graphs in the corpus, and were therefore not recognised. A further example of an AMR that was not recognised can be found in Fig. 8, and we see that the subgraph in Fig. 7(a) is embedded in this AMR — its occurrence is marked with a square.

![Fig. 7. The two subgraphs that cause the recognition to fail.](image)

It should be noted that the AMRs in Fig. 9 and Fig. 10 were recognised, i.e., the lack of one of the outgoing relations arg0 and arg1 for an event does not cause the recognition to fail.

8 Conclusion and Future Work

The results show that the description of the AMRs in this particular corpus is almost entirely covered by the HRG used in our experiments checking the rate
Fig. 8. An AMR that was *not* recognised. It can be realised into the English sentence “The girl wants to want”. The subgraph that causes it to fail is marked with a square.

Fig. 9. An AMR that was correctly recognised in which the concept *believe* has no arg1 relation. It can be realised into the English sentence “The girl believes”.

Fig. 10. An AMR that was correctly recognised in which the concept *believe* has no arg0 relation. It can be realised into the English sentence “The boy is believed”.
of recognition. Furthermore, we strongly believe that the HRG can recognise the entire corpus if we allow for some modifications to it. By adding rewrite rules that enable the subgraphs in Fig. 7 to be used as leaves, we expect the HRG to include the AMRs that failed in our experiment in the language defined by it. Should the modified HRG not be PTD parsable, we would like to figure out why and if it would still be possible to achieve polynomial-time parsing for the HRG.

In this study, we do not check nor prove that the HRG does not over-generate, i.e., that the HRG does not generate AMRs that are not in the corpus. This is an element that would have to be incorporated into an extended study.

Future work could, apart from the topics that have already been mentioned, include similar testing that has been done here but on different corpora over larger domains using different HRGs. Such experiments will be made easier as the number of AMR corpora grows. Moreover, the generation of correct AMRs from HRGs should also be investigated.

References

15. Linz, P.: An Introduction to Formal Languages and Automata. 5th edn. Jones and Bartlett Learning, LLC. (2012)
Model-based RSA Using NURBS Models

Heba Shehabeldin
Department of Computing Science
Umeå University, Sweden
mcs11hdb@cs.umu.se

Abstract. Model-based Roentgen Stereophotogrammetric Analysis techniques (RSA) have been developed to overcome the drawback of the conventional marker-based RSA which is that artificial markers must be inserted into orthopaedic implants. In model-based RSA, a 3D geometric model of an implant is used instead of the markers to determine the 3D position and orientation of the implant from its projections on two radiographs (x-ray images). The pose of the implant is obtained by finding the best fit between an actual projection of the implant which appears on the radiographs and a virtual projection of the 3D geometric model of the same implant. This paper proposes a model-based RSA algorithm that uses Non-Uniform Rational B-Splines (NURBS) models. We tested the precision of the developed algorithm on determining the center of a femoral head and compared it with an algorithm from the literature which uses Elementary Geometrical Shape (EGS) model of the femoral head. The precision is the same but by using NURBS models, it is possible to model complex shapes that cannot be described by EGSs.

1 Introduction

Roentgen Stereophotogrammetric Analysis (RSA) [1][2][3] is an accurate method for determining micromotions in the skeletal system. In RSA, the 3D pose (position and orientation) of a skeletal segment, e.g., bony segment or artificial implant is computed from the projection of the segment on two radiographs. By localizing two segments in one 3D coordinate system, their relative positions can be computed. In this sense, RSA has been used to measure migration of implants over time. The migration of implants is the change of their positions with respect to their surrounding bones.

RSA considers any implant as a rigid body in which the distance between any two points in the implant remains constant through motion. If at least three noncollinear points in a rigid body are known, the pose of the whole body can be determined. Hence, for accurate measurements, tantalum beads are inserted into prostheses and their surrounding bones. A patient together with a reference cage is then x-rayed (exposed to two synchronized roentgen tubes) resulting in two radiographs. The cage is used for calibration. It contains two kinds of radiopaque markers (control and fiducial markers) held in fixed, well-defined positions. The projections of the cage markers are measured on the radiographs. Using the
known 3D positions of the cage markers and their measured 2D projections on the radiographs, a 3D coordinate system, usually called the laboratory coordinate system, can be defined. The 3D positions of the two focal spots are assessed in a similar way. For each patient marker, its 2D projections are measured on the radiographs then its 3D position is computed as the intersection point between the two lines connecting the focal spots and the measured projections on the radiographs.

The process of inserting markers into implants is expensive, time consuming, and reduces the strength of the implants. Hence many techniques have been investigated to avoid the insertion of artificial markers. One technique that avoids inserting markers is called model-based RSA [4] [5]. Model-based RSA uses a 3D geometric model of an implant and matches a projection of that model with the actual projection on the radiographs. The pose of the implant is found by minimizing the difference between the two projections.

To date, model-based RSA techniques have been performed on Elementary Geometrical Shape (EGS)-based and triangle-based models [6] [7] [8]. Implants that have distinctive canonical shapes (e.g., sphere, cylinder, and cone) can be modeled using EGS models, while those that do not have distinctive landmarks can be modeled using triangle-based models.

The disadvantages of these models are that EGS-based models can not be used to model complex shapes, and triangle-based models are resolution dependent [9] and require a huge amount of memory. Instead we propose to use Non-Uniform Rational B-Splines (NURBS) models which are compact, much smoother, resolution independent, and invariant under affine and projective transformations [10] [11] [12]. Most importantly, they provide a unified representation for both standard and complex free form shapes.

As the proposed NURBS model-based RSA algorithm relies on a 3D NURBS model of the implant, generating NURBS models of implants is of great importance. Fortunately, modelling of human parts (bio CAD modelling) [13] has become easy due to developments in imaging and reverse engineering techniques. Two common techniques for modelling human parts using NURBS are found in the literature [14] [15]. They are based on either knot insertion or knot removal techniques [10].

The aim of this study is, therefore, to develop a NURBS model-based RSA algorithm and to ensure that its precision is acceptable.

2 Preliminaries

2.1 NURBS Curves

A NURBS curve of degree $d$ is a piecewise curve, where each piece is a degree $d$ rational curve. A knot vector $U$ defines where these pieces start and end. The shape of the curve is controlled by a set of $n + 1$ weighted control points. Hence, a NURBS curve may be defined as

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,d}(u)W_i P_i}{\sum_{i=0}^{n} N_{i,d}(u)W_i}, \quad (1)$$
where \( C(u) \in \mathbb{R}^3 \) is a point on the curve at the parameter value \( u \), \( d \) is the degree of the curve, \( P_i \in \mathbb{R}^3 \) is a control point, \( W_i \) is the weight associated to \( P_i \), and \( N_{i,d}(u,U) \) is a B-Spline basis function defined recursively by the Cox-deBoor recursion formula [10]:

\[
N_{i,0}(u) = \begin{cases} 
1 & u_i \leq u \leq u_{i+1}; \\
0 & \text{otherwise,}
\end{cases} \tag{2a}
\]

and

\[
N_{i,d}(u) = \frac{u - u_i}{u_{i+d} - u_i} N_{i,d-1}(u) + \frac{u_{i+d} - u}{u_{i+d+1} - u_{i+1}} N_{i+1,d-1}(u). \tag{2b}
\]

Fig. 1 shows a semicircle represented with a NURBS curve of degree \( d = 2 \). Five 3D points \( P_0, P_1, \ldots, P_4 \) with weights \( 1, 0.7071, 1, 0.7071, 1 \) control the shape of the curve. A knot vector \( U = [0, 0, 0, 0.5, 0.5, 1, 1, 1] \) divides the parameter domain into two actual knot spans \([0, 0.5)\) and \([0.5, 1]\), which correspond to the two periods \([-1, 0)\) and \([0, 1]\) on the horizontal axis of Fig. 1, respectively. Therefore the semicircle consists of two curve segments. Each segment is controlled by at most \( d + 1 \) control points.

Fig. 1: Example of a NURBS curve of degree 2. The shape of the curve is controlled by five control points \( P_0, P_1, \ldots, P_4 \) with weights \( 1, 0.7071, 1, 0.7071, 1 \). A knot vector \( U = [0, 0, 0, 0.5, 0.5, 1, 1, 1] \) divides the parameter domain into two knot spans \([0, 0.5)\) and \([0.5, 1]\) and hence two curve segments. One segment is controlled by control points \( P_0, P_1, P_2 \) and the second segment is controlled by control points \( P_2, P_3, P_4 \).

### 2.2 NURBS Surfaces

NURBS surfaces are functions in two parameters \( u \) and \( v \). They can be viewed as the limit of a set of NURBS curves that we have by holding \( u \) (or \( v \)) constant and varying \( v \) (or \( u \)) between 0 and 1. This scheme is called a tensor product.
scheme [16]. Hence, a NURBS surface may be defined by the rational expression
\[
C(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,d_u}(u)N_{j,d_v}(v)W_{i,j}P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,d_u}(u)N_{j,d_v}(v)W_{i,j}},
\]
where \(C(u, v)\) is a 3D point on a NURBS surface at the parameter values \(u, v\); \(d_u\) and \(d_v\) are the degrees of the curves in \(u\) and \(v\) directions; \(P_{i,j}\) is a control point, \(W_{i,j}\) is the weight associated to \(P_{i,j}\), and \(N_{i,d_u}(u), N_{i,d_v}(v)\) are B-Spline basis functions and are functions in knot vectors \(U\) and \(V\), respectively. Fig. 2 shows an example of a NURBS surface.

3 Method

During the RSA procedure, an implant of interest, i.e., inserted into a patient, is projected on two x-ray images. Model-based RSA techniques use these images to determine the pose of the implant with the aid of a 3D geometric model of the implant. In this study NURBS models of implants are used. The NURBS model-based RSA algorithm consists of three major steps:

1. A NURBS model of the implant that we want to determine its pose is obtained.
2. Points \(\tilde{q}\) on the projected contour of the implant are measured on the two radiographs.
3. The pose of the implant is computed by solving a constrained optimization problem to minimize the distance between the actual contour measured on the radiographs and a computed contour of the 3D NURBS model.

In this study, the center of a femoral head is computed. The actual model of the femoral head should be a hemisphere [1] but as we only compute the center and ignore the orientation we can assume that the model is a full sphere.

3.1 NURBS Model of the Femoral Head

A NURBS model of the femoral head is assumed to be a scaled and translated version of a NURBS model of a unit sphere. Fig. 2 shows a NURBS model of the unit sphere. The model is defined by a 9x5 array of control points in homogeneous coordinates, two knot vectors \(U = [0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1]\) and \(V = [0, 0, 0, 0.5, 0.5, 1, 1, 1]\); and the degrees of the curves in \(u\) and \(v\) directions \((d_u = d_v = 2)\).

3.2 Problem Formulation

The problem of minimizing the distance between the measured points \(\tilde{q}\) on the projected contour of the femoral head and the virtual projection \(q\) of its 3D
Fig. 2: A NURBS surface model of a unit sphere. It is defined by 45 weighted control points of which some points are repeated, two knot vectors \(U = [0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1]\) and \(V = [0, 0, 0, 0.5, 0.5, 1, 1, 1]\), and the degrees of the curves in \(u\) and \(v\) directions (\(d_u = d_v = 2\)).

NURBS model may be formulated as a constrained non-linear least squares optimization problem of the form [17]:

\[
\text{min}_x \sum_i ||q_i - \tilde{q}_i||^2 + \sum_j ||q_j - \tilde{q}_j||^2
\] (4a)

subject to:

\[
\frac{A^T_1 p_i + b_1}{v_1^T p_i + 1} = q_i, \quad \frac{A^T_2 p_j + b_2}{v_2^T p_j + 1} = q_j,
\] (4b)

\[
t + sC(u_i, v_i) = p_i, \quad t + sC(u_j, v_j) = p_j,
\] (4c)

\[
(p_i - f_1)^T \hat{n}(u_i, v_i) = 0, \quad (p_j - f_2)^T \hat{n}(u_j, v_j) = 0,
\] (4d)

where \(x = (p_i, q_i, p_j, q_j, u_i, v_i, u_j, v_j, s, t)\) is the vector of unknowns; points related to images 1 and 2 have subscripts \(i\) and \(j\), respectively; \(p\) is a 3D point on the NURBS model of the femoral head; \(q\) is the 2D projection of \(p\); \(u\) and \(v\) are the values of the parameters where the point \(p\) is assumed; \(C(u, v)\) is a 3D point on the NURBS model of the unit sphere; \(f_1, f_2 \in \mathbb{R}^3\) are the positions of the X-ray focal spots; \(\hat{n}(u, v) \in \mathbb{R}^3\) is a normal vector of the femoral head’s model at the point corresponding to \(u\) and \(v\); \(A, v, f,\) and \(b\) are known from the reconstruction of the projection geometries as described in [18]; \(s \in \mathbb{R}\) is a scaling factor and corresponds to the radius of the femoral head; \(t \in \mathbb{R}^3\) is a translation vector and corresponds to the center of the femoral head.

As illustrated in Fig. 3, Equations (4b) state that each 2D point \(q_i\), is a projection of a corresponding 3D point \(p_i\); Equations (4c) restrict each 3D point
to be on the surface of the femoral head’s model which is assumed to be a scaling and a translation of the NURBS model of the unit sphere. Equations (4d) ensure that the projection ray \((p_i - f)\) is the tangent to the model at point \(p_i\) and hence orthogonal to the normal vector \(\hat{n}_i(u,v)\) of the femoral head’s model at that point. The normal vector \(\hat{n}_i\) is computed as the cross product of the two vectors
\[
\hat{u}_i = \frac{\partial(t+sC(u,v))}{\partial u} \quad \text{and} \quad \hat{v}_i = \frac{\partial(t+sC(u,v))}{\partial v}.
\]

Fig. 3: Central projection of a femoral head. The femoral head is modeled by a NURBS sphere of radius \(s\) and center \(t\). Points on the surface of the sphere model are denoted \(p_i\). The 2D projection of these points are denoted \(q_i\). The focal spot \(f\) is the center of projection. \(\hat{n}_i\) is the normal vector of the sphere at point \(p_i\). The projection ray \((p_i - f)\) is tangent to the sphere at point \(p_i\).

We solved optimization problem (4) using the sequential quadratic programming (SQP) optimization scheme with Gauss-Newton approximation [19, 20].

3.3 Sequential Quadratic Programming

SQP is a successful technique for solving optimization problems with nonlinear constraints. It can be viewed as an application of Newton’s method for unconstrained optimization. In unconstrained optimization, Newton’s method is used to find a minimum of an objective function
\[
\min_x F(x).
\]

Newton’s method is an iterative method. It starts by an initial approximation of the variables \(x\) and at each iteration it approximates the nonlinear objective
function $F(x)$ by a quadratic function $F(x + h)$ using the first three terms of the Taylor series

$$F(x + h) = F(x) + h^T \nabla F(x) + \frac{1}{2} h^T \nabla^2 F(x) h,$$

(6)

where $\nabla F(x)$ and $\nabla^2 F(x)$ are the gradient (a vector containing the first order partial derivatives of $F(x)$) and Hessian (a matrix containing the second derivatives of $F(x)$) of $F(x)$, respectively. Then it computes a descent search direction $h$ by setting $\nabla F(x + h) = 0$, the first order necessary condition for a minimizer of $F(x + h)$, and solving for $h$.

$$\nabla^2 F(x) h = -\nabla F(x).$$

(7)

The new iterate is defined as

$$x_{k+1} = x_k + \alpha_k h_k,$$

(8)

where $\alpha_k$ is the length of a step along $h$ chosen to guarantee that $F(x + \alpha_k h_k) < F(x)$.

In the case of constrained problems, the optimization problem is generally formulated as

$$\min_x F(x)$$

subject to $c(x) = 0,$

(9a)

(9b)

where $c(x)$ is a vector of $m$ nonlinear constraint functions. Now the search direction $h$ is not only required to be a descent direction but also to be a feasible direction, i.e., moving a small step along that direction does not violate the constraints. The SQP method solves this constrained problem by converting it to an unconstrained problem and then applying Newton’s method for the unconstrained problem. A function called the Lagrangian function $\mathcal{L}(x, \lambda)$ is used to combine the objective function, $F(x)$, and the constraint functions, $c(x)$, in one single function

$$\mathcal{L}(x, \lambda) = F(x) - \lambda^T c(x).$$

(10)

SQP applies Newton’s method to the Lagrangian function as we do with the unconstrained problem. First, $\mathcal{L}(x, \lambda)$ is approximated using the Taylor series

$$\mathcal{L}(x_{k+1}, \lambda_{k+1}) = \mathcal{L}(x_k, \lambda_k) + h^T \nabla \mathcal{L}(x_k, \lambda_k) + \frac{1}{2} h^T \nabla^2 \mathcal{L}(x_k, \lambda_k) h.$$  

(11)

Second, a feasible descent direction is obtained by setting $\nabla \mathcal{L}(x_{k+1}, \lambda_{k+1}) = 0$ and solving for $h$

$$\nabla^2 \mathcal{L}(x_k, \lambda_k) h = -\nabla \mathcal{L}(x_k, \lambda_k),$$

$$\begin{bmatrix} \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k) & -A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} h_k \\ h_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(x_k, \lambda_k) \\ -c(x_k) \end{bmatrix}.$$  

(12)
where $A(x)$ is the Jacobian matrix of the constraints (i.e. the first order partial
derivatives of the constraints with respect to the vector of unknowns $x$). Finally,
the new Newton’s step is computed as

$$
\begin{bmatrix}
x_{k+1} \\
\lambda_{k+1}
\end{bmatrix} = \begin{bmatrix}
x_k \\
\lambda_k
\end{bmatrix} + \begin{bmatrix}
h_k \\
h_\lambda
\end{bmatrix}.
$$

(13)

For least squares problems with nonlinear constraints [19] [20]

$$\min_{x} \frac{1}{2} \|r(x)\|^2$$

subject to $c(x) = 0$,  

the Lagrangian function is

$$\mathcal{L}(x, \lambda) = \frac{1}{2} \|r(x)\|^2 - c^T \lambda.$$  

(15)

The partial derivatives of (15) with respect to $x$ are

$$\nabla_x \mathcal{L} = J^T r - \nabla c \lambda, \quad \nabla^2_{xx} \mathcal{L} = J^T J + \sum_i r_i(x) \nabla^2 r_i(x) + \sum_i \lambda_i \nabla^2_{xx} c_i,$$

(16)

where $J$ is the Jacobian matrix of the residual $r(x)$ (i.e. the first order partial
derivatives of $r$ with respect to the vector of unknowns $x$). By substituting (16)
into (12) and using the Gauss-Newton approximation which ignores the curva-
tures information $Q$ and $Q_c$, we get

$$\begin{bmatrix}
J^T J & -A^T \\
-A & 0
\end{bmatrix} \begin{bmatrix}
h_k \\
h_\lambda
\end{bmatrix} = \begin{bmatrix}
-J^T r + A^T \lambda \\
c
\end{bmatrix}.$$

(17)

3.4 Implementation

Points $\tilde{q}$ on the projected contour of the femoral head were extracted from the
two radiographs. In this paper, we used the points given in [17]. The points form
two arcs as shown in Fig. 4. Solving problem (4) using the SQP method requires
calculating the Jacobian $J$ of the residual function and the Jacobian $A$ of the
constraint functions in (17). We calculated them analytically, except the partial
derivatives with respect to the parameters $u$ and $v$ of constraint function (4d)
which were approximated using the finite difference method. The derivatives we
computed analytically were verified by the finite difference method.

An initial approximation of the unknowns was configured based on the mea-
sured points $\tilde{q}$ and on our knowledge of the problem at hand. The initial approx-
mation of $q$ was assumed to be equal to the measured 2D points $\tilde{q}$. The initial
approximation of the center $t_0$ was estimated from Equations (4b) by solving
the over determined system

$$t_0 = \begin{bmatrix}
m_{q1} v_1^T - A_1^T \\
m_{q2} v_2^T - A_2^T
\end{bmatrix}^{-1} \begin{bmatrix}
b_1 - m_{q1} \\
b_2 - m_{q2}
\end{bmatrix},$$

(18)
where \( m_{q1} \) and \( m_{q2} \) are the centers of gravity for points \( q_i \) and \( q_j \), respectively. The initial value of the radius \( s0 \) was given by the manufacturer of the prosthesis and it is 0.014 m. The initial values \( u0 \) of the \( u \) parameters were random. The initial values \( v0 \) of the \( v \) parameters were set to 0.5. The initial model of the femoral head was obtained by transforming the unit sphere model using the transformation parameters \( s0 \) and \( t0 \). The initial approximation of the 3D points \( p \) was the 3D coordinates of the points on the initial model of the femoral head which correspond to the parameters \( u0 \) and \( v0 \).

4 Results

When we used the sphere of Fig. 2 as the NURBS model of the femoral head, singularity at the poles of the sphere model was encountered. The singularity means that at some iterations of the SQP algorithm, the values of the \( u \) and/or \( v \) parameters go below 0 or over 1. In this case, points on the sphere which correspond to these values of the parameters are not defined. The singularity problem was solved in two steps:

1. The first step was to rotate the sphere in Fig. 2 around the y-axis to align the poles of the sphere with the z axis. By doing so, we removed the poles away from the two sphere arcs whose 2D projections \( \tilde{q} \) may appear on the radiographs. Fig. 5 shows the model of the femoral head after rotation.
2. The second step was to enforce some conditions on \( u \) and \( v \) when the singularity occurs. The \( u \) domain, as illustrated by the knot vector \( U \), is a full circle. Hence, it is periodic and a condition can be set to remedy the singularity. For example, if \( u \) becomes 1.1 we set it to 0.1, and if it becomes −0.1 we set it to 0.9. The \( v \) domain is a semicircle so it is not periodic. For solving
the singularity of the $v$ parameter, we set the $v$ parameter value between 0 and 1 and changed the $u$ parameter. For example, if $v$ becomes 1.1 or $-0.1$ we set it to 0.5, and change the corresponding $u$ parameter’s value to $u+0.1$.

By solving the singularity problem, the algorithm converged to a solution. We used a convergence tolerance of $10^{-8}$ for both the residual and the constraint functions. Fig. 6 shows the normalized objective function value at each iteration. The relative error $e$ between solution $x1$ of the proposed algorithm and solution $x2$ of the algorithm found in [17] was computed as

$$e = \frac{||x1 - x2||}{||x1||}.$$  \hspace{1cm} (19)

The relative error of the center is $2.6802^{-12}$ and of the radius is $2.8195^{-11}$. Fig. 7 shows the initial and the final 3D positions of the points on the femoral head which their 2D projections $\tilde{q}$ were shown in Fig. 4. The final positions form two arcs on the 3D model.

5 Conclusion

In this paper, a model-based RSA algorithm that uses NURBS models of implants was presented. We showed that the algorithm can estimate the position of the femoral head from its projections on two RSA radiographs with acceptable precision. Determining the implant’s orientation was not handled though. But, it can be handled in the future work by replacing the scaling parameter $s$ of
Fig. 6: Normalized objective function value at each iteration of the SQP algorithm.

Fig. 7: The initial and the final 3D positions of points $p$ on the femoral head component which their 2D projections $\tilde{q}$ were measured on the two RSA radiographs.
Equation (4) with a rotation matrix. Compared to model-based RSA techniques that uses EGS-based models, the NURBS model-based algorithm is generic and can estimate the pose of implants that can not be modeled by EGSs. Compared to techniques that use triangle-based models, the algorithm is resolution independent and does not need a huge storage requirements. In the future work, the algorithm will be used to investigate implants that have non canonical geometries.

References

Using Dependency Parse Trees as a Method for Grounding Verbal Descriptions to Perceived Objects

Alexander Sutherland
Department of Computing Science
Umeå University, Sweden
alsu@cs.umu.se

Abstract. This paper presents an approach towards the grounding of external entities to natural language utterances by representing physical objects as trees, sharing the same structural properties as trees that describe the syntax of sentences. The paper then presents a method of connecting these representations to natural language requests based on the shared structure between representations, defined as the similarity between representations. The test domain is limited to a set of cards with a simple set of attributes and possible spatial relations. Results show that, for single cards, it is possible to perform grounding with a similarity of 60%. In cases with multiple cards that have relational properties we are able to achieve between 75% and 25% similarity, depending upon the spatial relations between objects.

1 Introduction

Symbol grounding, originally described in [1], pertains to the act of connecting meaning to symbolic representations, such as words. In doing so, systems are able to perform more advanced tasks as they are in possession of a more complex understanding of their environment. As such, the construction of a reliable grounding procedure is required in order to boost the performance of any systems that rely on symbol grounding, such as natural language interpreting systems.

Natural language commands can be structured and interpreted in a variety of ways, an example being Stanford’s universal dependency parse trees [2]. When using a dependency parse tree representation, sentences are structured as directed acyclic graphs. Nodes in the graph represent words in a sentence and edges represent how words modify and relate to each other. This manner of describing syntax is presented as being easily interpretable even for non-professional linguists, as seen in [3].

In this paper, we present a method that potentially helps solve the symbol grounding problem by extending a solution to the similar anchoring problem, introduced in [4]. The anchoring problem is similar to the symbol grounding problem in that it has to do with attributing meaning to symbolic representations. What sets the anchoring and symbol grounding problems apart is that...
anchoring only deals with the connecting of sensory data, such as color or position, to symbols. Symbol grounding should in addition be capable of adding higher level conceptual and contextual meanings to symbols, such as possession or status. As such, symbol anchoring can be seen as a simplification of the symbol grounding problem. It should be noted that we do not present a solution to the symbol grounding or symbol anchoring problems, rather we present a method of structuring internal representations based on the data given by a symbol anchoring solution. This can be seen as an intermediate step between the two problems.

As an example of a symbol grounding problem let us consider a scenario that we will use in this paper. In our hypothetical scenario a human and robotic agent are sitting across from each other, at a table. A glass wall separates the two and the only way they can interact is through a small slit in the wall. The robot is given a set of random cards from a traditional deck of cards and the humans task is to request that specific cards be handed to them, through the slit. In order to determine which card to give to the human the robot would have to associate the meaning of the humans request with one of the cards it has available. We assume that the robot does not automatically know the name of the card, i.e. the Ace of Spades, the Jack of Hearts, as these values are inherent to our human interpretation of the cards.

Symbol anchoring in our example scenario would be the robot associating some sensory input with some symbolic denotation, such as some RGB value being red, some detected shape being that of a card, some detected number being associated with a value, etc. What we consider symbol grounding conversely is the robots ability to map meaning to more complex entities, i.e. the robots ability to map the Ace of Spades to the card representation or the ability to determine if the symbolic representation of an object is equivalent to another representation.

For the purpose of simplicity we assume that our robotic agent is capable of connecting sensory data to appropriate symbols as described by an anchoring table, see Table 1, that designates sensor data-symbol pairings. We assume that our robot is in possession of a visual sensory system that is able to fulfill the input requirements listed in the anchoring table. The anchoring table is a specification of how sensory input correlates to an internal representation. Every time an input requirement is fulfilled a certain structure is added to a physical objects representation, as specified by the generation column.

An approach to Symbol grounding is performed in this paper by calculating a similarity metric, see Sec 5, between the structured representations of the verbal utterance and a perceived object. Both representations are structured in the form of dependency parse trees (DPTs), see Sec. 3. While the structuring of verbal input is given by the Stanford parser, presented in [2], the generation rules for DPTs of visual input need to be defined as seen in Sec. 4. Identifiable objects in an environment will have these so-called visual dependency parse trees (VDPTs), generated for them and each VDPT will be compared to the input verbal DPT. The VDPT that is most similar to the input verbal DPT, according to our metric, is selected as the intended object. In the case of our example, the
intended card can be considered grounded through the selection of that cards VDPT based on the input verbal DPT.

2 Related Work

Over a decade of research, as outlined in [5], has been spent examining the Symbol Grounding problem and still no consensus has been reached. A clear example of this is the work by Taddeo et al. in [6] wherein the authors present a method of symbol grounding through the usage of an agent architecture that allows agents to ground symbols semantically without breaching the rule set presented by the authors. The approach used was based on observations in their previous work [5]. Even this solution was refuted by Bielecka et. al in [7] as Taddeo et al. failed to meet their own requirements for symbol grounding. This has left the problem of symbol grounding unsolved.

The indeterminate state of symbol grounding solutions causes obstacles in the development of embodied agents. Embodied agents, such as robots, are physical entities comprising hardware and software, that perform tasks in the physical world. Obstacles can manifest as an inability to interpret natural language to a satisfactory degree, leading to agents being unable to fulfill given commands due to a lack of understanding. An example scenario of when a robot may fail to perform a task due to insufficient symbol grounding capabilities is when dealing with possessions. If we ask a robot to retrieve “John’s ball” we expect to receive a specific ball. However if it cannot properly ground the ball as being a possession of Johns the robot may fruitlessly search after a ball with this property forever, as the robot cannot see possession, or may simply return with the first ball it can find. In either case we have failed to adequately communicate our request through the medium of natural language. This facilitates the necessity of a system capable of forming a rudimentary chain of connections between symbols and their meaning.

What constitutes symbol grounding is rather diffuse in nature, as the problem itself is somewhat philosophical. Many methods of Symbol Grounding have been presented but ultimately fail to pass the non-semantic requirements laid out by Taddeo et al. [5]. In the work by Yu et al. [8] we are introduced to an attempt at symbol grounding that locates real world objects by using verbal and visual inputs alongside a learned categorization of objects. The integration of multiple sources of sensory data in Yu et al. acts as an example of how to approach solving the symbol grounding problem. The difference between our approaches is that Yu et al. focus on learning and categorization of sensory data, whereas the method we introduce is primarily directed toward connecting internal and external representations.

Similar work that attempts to apply higher level cognitive concepts can be found in [9] by Hudelot et al. In the paper, the authors perform semantic image interpretation and in this manner attempt a variant of symbol grounding that relies on two staged classification. The first stage interprets sensory data as a set of descriptive attributes drawn from their image concept ontology. In the
second stage the attributes are interpreted by a visual concept ontology which then performs higher level semantic structuring. The paper not only introduces a potentially useful two-stage anchoring method but also introduces semantic structure. These semantic relations can potentially help to reinforce symbolic representation through a particular symbols relationship to other symbols. Our methods used differ in the structuring of semantic data, as we use VDPTs to structure relations and Hudelot et al. make use of a separate ontology.

Another method that also attempts to use the structure of sentences in symbol grounding is presented in [10] and is then used to discuss their presented framework. The generative model based upon the structure of sentences can perform spoken commands and can locate environmental objects but was highly corpora dependent and as such hard to generalize. Their new framework attempts to solve this by using probabilistic models that could dynamically take into account natural language structuring. The approach used is more general and focuses on learning from corpora, in contrast to our method that focuses on the connection of external and internal representations.

The papers by Yu et al. [8] and Tellex et al. [10] both attempt to tackle symbol grounding while struggling with semantic interpretation of the entire sentence. While semantics is without a doubt a necessity in natural language processing its involvement in symbol grounding quickly becomes parasitic as symbolic denotations of objects become defined by some external database, as mentioned in [5]. In order to avoid the semantics of the sentence influencing the symbol grounding procedure we refer to our earlier work, seen in [11], where we regulate semantic interpretation to higher level action classification of sentences via shallow semantic parsing. Shallow semantic parsing [12] is the division of words in a sentence based upon their semantic function in regard to some primary word. The entities specified by these semantic roles will often be expressed as noun phrases. This leaves us with the task of grounding symbols within noun phrases, a simpler yet still non-trivial task.

Our proposed method relies on the ability to generate what is essentially structured representations, similar to DPTs, of a visual space as perceived by a robotic agent. An example where a similar approach has been taken is in [13] where the authors present a system capable of creating descriptive annotations of images. Our approaches are similar in that they seek to describe an environment using natural language properties. In contrast we choose to further extend this by attempting to correlate the generated environmental representations with some verbally requested object. Many of the chosen references focus on internalization of external objects, which is a key aspect of symbol grounding.

As is evident by the variety of approaches described, there is no obvious correct way to perform symbol grounding and the structures used are often tailored to a specific task. This is also done in the present paper, as we make use of representations that are syntactically oriented. Tailoring how a representation is structured acts as a means of bridging the gap between theoretical concepts and practical applications. This simplification that allows for motivation around whether symbol grounding has occurred or not based on the representation used,
as there is no clearly defined metric for when “meaning” has been attributed to a symbol. In the present paper a form of grounding occurs between the aural representation and the visual representation with the highest similarity.

3 Universal Dependency Parse Trees

Stanfords Universal Dependency parse trees (DPTs), presented in [2], are trees that describe the syntactic structure and relationships of a sentence. Examples of such relationships are adjectives (colors, sizes, speeds etc) and prepositions (above, inside of, on, etc) related to a root noun, where a noun is the name of some specific thing. As described in [3], the Stanford typed dependencies were introduced in order to create an easy-to-understand representation of the syntactic relationships between words that can be found in sentences. The design philosophies behind DPTs are not only meant to be accessible but the rule-set that governs the construction of the DPTs is also predictable within certain limits.

Dependencies typically exist as triples, describing a governing word and dependent word as well as the relationship between them. An example is that for the noun phrase “the red ball” the governing word “ball” has a dependent word “red” and there is an adjectival modifier relation between the two, e.g. amod(ball,red). A sentence can therefore be described as a set of triples that, when viewing the relationships between all words in a sentence, generate a syntactic graph. In the case of noun phrases, a group of words that are related to a noun, these relationships can potentially be useful as a means of determining what words modify the core noun and therefore what attributes a visual system should attempt to find in an environment. An example DPT can be seen in Fig 1.

![Fig. 1. Example DPT that represents the sentence “the red card with a nine”.

4 Visual Dependency Parse Tree Generation

This section describes the concept of VDPTs and motivates why it is beneficial to attempt to mimic Stanfords Universal DPTs. Furthermore, the conditions for
4.1 Motivation of the VDPT Representation

This paper makes use of the simplicity of DPT design principles by attempting to recreate the same structure from visual sensory input data. What this leads to is a system “seeing through sentences” as objects in the environment are represented in structured graphs that resemble DPTs. Representing data in this manner leads to the creation of a consistent and predictable mapping scheme that can describe complex spatial relationships between physical objects.

Furthermore, it should simplify the task of locating specific objects in an environment that are described in natural language. The reasoning behind this is that objects in the environment should, to some degree, be similar to their requested natural language counterpart, such that a relationship can be created between the two.

4.2 Descriptive Attribute Classification

In general, attributes in noun phrases commonly fall into the categories of either adjectival modifiers or nominal modifiers. As an example, examine the sentence “the red card with a heart on it”. In this sentence, “red” is an adjectival modifier, i.e. an adjective that modifies the noun, in this case the card. “Heart”, alternatively, is a noun that acts as a property of the card, and is a nominal modifier.

As attributes can be expressed in either modifier, the syntactic tree can in accordance also take multiple configurations. Given that the visual system attempts to identify the most likely structure of the syntactic tree, this causes issues when considering how to structure the visual representation. An example of this being the noun phrases “the red card” and “the card with the color red”. Both phrases can refer to the same object but the former has a tree structure that includes an adjectival modifier while the latter contains a nominal modifier.

In short, a single VDPT can have multiple valid DPTs.

4.3 VDPT Generation Rules

The generation rules are designed in such a way as to create VDPTs, that partially mimic a syntactic DPT counter-part. The anchoring table, see Table 1,
Fig. 2. Example of two DPTs that both denote a red card. The bidirectional arrow indicates semantic equivalency of the graphs.

describes how sensory inputs are bound to certain low-level symbols, such as denotations of color, size, solidity, etc. In this paper we use a set of cards as our test domain. The attributes these cards can assume are located in the following sets:

\[ C = \{\text{red, black}\} \]
\[ V = \{\text{two, three, four, five}\} \]
\[ P = \{\text{above, under, beside, to the right of, to the left of}\} \]

In order to generate the VDPT we assume a recognized object has a color from the set of colors in \( C \) and a value from the set of values in \( V \). The VDPT can possibly also have some preposition from the spatial relationships in \( P \) to another card. We generate VDPTs according to the rules described below. The structure of the VDPT can be described in the form of a set of triples:

\[ nmod : Q(X, Y) \text{ or } amod(X, Y) \]

where \( X \) and \( Y \) are unique labeled symbols and \( Q \) is some relational entity, specifying the relationship between the two. These values are specified in Table 1 and the sub-trees to be added to the VDPT in association with certain anchored values are specified in the “VDPT generation” column.

<table>
<thead>
<tr>
<th>Rule nr</th>
<th>Symbol</th>
<th>VDPT generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>card</td>
<td>root(ROOT,card)</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>amod(card,c)</td>
</tr>
<tr>
<td>3</td>
<td>v</td>
<td>nmod:with(card,v), case(v,with)</td>
</tr>
<tr>
<td>4</td>
<td>card</td>
<td>nmod:p(card,card), case(card,p)</td>
</tr>
</tbody>
</table>

Table 1. Anchoring table used to determine VDPT rule activation.

It should be noted that classification is the ability of a robotic agent to categorize a detected item according to some set of metrics. On the contrary, identification is the robots ability to determine if some visual input data is an object that it can manipulate and interact with, i.e. not just a part of the back-
The tree generation rules for the VDPT that make use of the anchoring table are as follows:

1. If an object can be identified in the environment a VDPT is created containing only a ROOT node.
2. If the new object can be classified in the environment it is set as the ROOT node with its classification as a label.
3. If sensory data exists such that it coincides with one of the entries in the anchoring table add the corresponding subtree with either an amod edge or a nmod:X edge as specified by the anchoring table.
4. If other identifiable items exist in the environment and the relationship between items can be described by the anchoring table, connect to that objects representation with a nmod:p edge. The $p$ is a relational subtree specified by the anchoring table.

5 Testing Methods

This section describes the testing setup and procedure. Here we will also specify the limitations placed upon the system in order to keep the work within the scope of the paper, the scope being able to represent the possible set of cards generated by the sets of colors, values, and positions, for our robot in the example scenario.

5.1 Similarity of Graphs

As VDPTs and DPTs can be defined as sets of triples, similarity can be considered the set of triples shared between the two parse trees. Assume a set of VDPTs $G_i$, $1 \leq i \leq n_v$, generated from observable objects within the robots field of view. For a given DPT $H$, the most similar $G_i$, is denoted $G_T$, where $T$ is given by:

$$ T = \arg \max_{1 \leq i \leq n_v} \left( \text{sim}(G_i, H) \right) $$

The similarity function, $\text{sim}(G, H)$, is described by the pseudo-code in Algorithm 1. In the algorithm $G$ is a set of triples describing the generated VDPT representation of an object in the environment. $H$ is a set of triples describing the DPT representation of a spoken sentence. The $\text{gov}$ and $\text{dep}$ functions return the governing and dependent word of a triple in one of the $j$ triples of $H$. A triple can only be considered similar if the governing word of that triple has either been visited before or acts as a ROOT node of the graph, so as to ensure we do not calculate similarity on unrelated representations.

In layman's terms this algorithm gives the similarity between a VDPT and a DPT by counting the percentage of triples that the two trees have in common.
Algorithm 1 Similarity function that calculates the structural similarity between two graphs, $G$ and $H$.

1: function $\text{sim}(G, H)$
2: for $j = 1$ to $\text{sizeof}(H)$ do
3:   if $G$ contains $H[j]$ then
4:     if $\text{gov}(H[j])$ is a ROOT then
5:       nodes $\leftarrow$ $\text{gov}(H[j])$
6:     end if
7:     if nodes contains $\text{gov}(H[j])$ then
8:       nodes $\leftarrow$ $\text{dep}(H[j])$
9:       $k \leftarrow k + 1$
10:    end if
11: end if
12: end for
13: $s_{(G,H)} \leftarrow k/\text{sizeof}(H)$
14: return $s_{(G,H)}$

A visual example of similarity in graphs can be seen in Fig 3, where two trees are presented and the thick lines are indicative of similar elements between the trees.

![Fig. 3. Example of a DPT and a VDPT that have a similarity of 4 out of 6 triples. Note that although we do not specifically show the connection between the root node and card we assume it to belong to the top node in the graph.](image_url)

5.2 Experiments

In the following tests we assume that we have a robotic agent with a visual system capable of classifying sensory inputs according to Table 1. The robot is given the task of sending a card through the slit. We assume that our robot is capable of identifying different cards and a set of attributes associated with those cards. We also assume that these cards are located against such a backdrop that these cards are the only objects being analyzed by robot.

Two scenarios are examined over the course of the testing procedure. The first “single card” scenario two cards are separately presented to the robot and it is required to pick a requested card associated with a natural language request by a human. In the second scenario two pairs of cards are separately presented.
to the robot. Each pair of cards have a spatial relation that is describable by a preposition. As in the first scenario a natural language sentence is spoken to the robot and it is required to pick a single card based on similarity between generated VDPT and DPT representations.

5.3 Test Procedure

For each test a number of cards are presented according to the sets $C$ and $V$. Every card after the first is assigned a randomly generated spatial relation to the first in accordance with the relations in $P$. For each card generated the possibility to draw that card again is removed, which entails that all cards generated will be different from a previously generated card. Thereafter, a sentence is spoken by the human with the format:

\[ \text{the } c \text{ card with a } v \ ( p \ 	ext{the } c \ 	ext{card with a } v \ ... ) \]

where $c \in C$, $v \in V$ and $p \in P$. The generated sentence is then processed by the Stanford parser and we receive a syntactic representation of the sentence in the form of a DPT. The number of VDPTs generated is equal to the number of cards that exist within the robot's field of view. The similarity between each VDPT and the DPT is calculated by the robot and the card that has a VDPT with the highest similarity to the DPT is selected and sent through the slit.

It is entirely possible that a sentence can have multiple DPT representations. This occurs when a sentence is ambiguously phrased, an example of this being the phrase “a glass of water”. DPTs can be generated representing a glass containing water or a glass made out of water. In these cases, where phrase syntax is ambiguous, the DPT with the highest probability is chosen.

6 Results

This paper presents a method that can structure visual input information from objects in the environment, given that we have a system capable of performing symbol anchoring. We also present how we can compare a generated structure to noun phrases from English natural language allowing us to perform low level symbol grounding of a natural language request to a physical entity.

Preliminary results indicate we are able to correctly correlate verbal and visual representations. For single cards we are able to correctly determine the eight possible cases with a 60% similarity metric for all generations. For two cards we are able to correctly determine a referred card in all 660 cases albeit with a 75% similarity metric for the “above”, “under” and “beside” prepositions. In the cases “to the left of” and “to the right of” we are only capable of reaching a 25% similarity, the reason being that syntactic structure changes in a way currently not defined by our generation rules.
7 Discussion

The results highlighted an important requirement for our system to make reliable connections, the ability to create spatial relations. It was observed that in the cases of 25% similarity the section of the representation reached only belonged to the first object and none of the objects it was spatially related to. As such, in certain scenarios, the system can break down if attributes are excluded. For example in a scenario with two red cards, on either side of a black card with a two, the sentence “The card to the right of the black card with a two” would fail, as the VDPT representations for both of the red cards would be equally similar to the DPT. Although this case was not within the scope of our test domain it should be an important facet of future work.

The possibility of being unable to determine a correct item if it cannot interpret the environment highlights the importance of covering a wide range of spatial relations, so as to ensure reliable interaction in robotic agents. Furthermore, as we have pointed out this method can fail if information is omitted and future work should expand the domain, ideally also using an embodied system, in order to identify other possible failure scenarios.

The system and the generative graph grammar used is quite limited in its current form. Ideally we would like to extend the proposed method by taking a more thorough look at the various syntactic configurations and the generative rules that would be required to cover the general case. If we examine the results from an embodied system that was capable of applying this method we may discover additional properties that would aid in the grounding procedure. Furthermore, the analysis of an embodied system should allow for the construction of a minimum similarity mechanic that would prohibit the system from drawing incorrect conclusions. However, without such an embodiment any attempt at designing such a mechanic for the method would be arbitrary, as we lack the feedback required from real world scenarios.

Whether the method we have proposed fulfills the definition of symbol grounding specified by Taddeo et al. [5] is arguable. The zero semantic commitment argument is rather hard to fulfill without having some kind of learning capability. Furthermore, the system is dependent on the graph grammar used and this can be interpreted as use of external semantics. However, we believe that the method we introduce in this paper does not perform semantic generation per se, but rather is a required precursor to proper symbol grounding.

In the future we would like to expand the concept presented in this paper even further by adding contextual information to the symbolic graph representation. As we are currently unaware of how well the system will scale given an extended corpora it is an important facet of any future work to increase the size of the test domain. Feasibly the syntactic constraints placed upon noun phrases should ensure a reasonably flexible format, with the ability to replace nouns, verbs, prepositions, and adjectives freely so long as the core syntactic structure remains the same. However, no clear conclusions can be about the systems generalization capabilities without access to a fully developed grammar and an
embodied system. By including these additional aspects we believe we would be able to finally accomplish proper symbol grounding in some form.

References

Evaluating Negotiation Approaches with Opponent Models for Multi-Agent Systems

Dawit Kahsay Weldemariam
Department of Computing Science
Umeå University, Sweden
mrc14kwt@cs.umu.se

Abstract. It has been previously shown that argumentation-based negotiation, as a means of multi-agent system interaction, outperforms proposal-based negotiation under a specific negotiation domain and using selected evaluation criteria. This empirical evaluation was conducted for agents that do not model the preferences of offers of their opponents. In this paper, we use the same negotiation domain and similar evaluation criteria, and show that both types of negotiation perform almost equally well if the agents learn the preferences of their opponents and make offers that not only increase their own utilities but also their opponents’ utilities.

1 Introduction

A multi-agent system, MAS, is composed of several autonomous agents. Faratin et al. [1] emphasize that while agents in MAS have a high degree of autonomy, they usually need to interact with other agents to achieve their objectives. The rationale for the interaction between agents can be because they do not have sufficient capabilities or resources to solve their problems alone, they want to optimize their objectives by exchanging resources or distributing tasks, or they simply have interdependencies between them. Automated negotiation is one way in which software agents communicate or interact in order to reach a mutually acceptable agreement among themselves [2]. It can be viewed as a distributed search through a space of potential agreements [3]. In bilateral negotiation [4], two agents aim at reaching a joint agreement by exchanging various proposals and arguments.

Several negotiation approaches have been studied as a means of facilitating inter-agent communication and coordination. These approaches are classified [5] into proposal-based negotiation (PBN) and argumentation-based negotiation (ABN). PBN is characterized by the exchange of offers between agents with conflicting interests and typically use game-theoretic approaches or heuristics to automate the negotiation. ABN, on the other hand, enables agents to exchange additional meta-information called arguments in the form of logical sentences during negotiation to justify or critique an offer. For example, an offer from an agent can be some resources that the agent can give in exchange for
another resources that the agent wants from another agent. In PBN, the agent which receives an offer can either agree, or disagree (reject) the offer and then make a counter-offer if it has any. But in ABN, the agent can additionally argue why it disagrees with the offer prior to making a counter-offer. The argument can be a justification message stating that the agent does not have the resources requested by the other agent or a critique message stating that the agent does not want the resources offered by the other agent.

An empirical comparison of PBN versus ABN have been conducted in [6] on a specific negotiation domain. The negotiation domain used for the evaluation was a partially observable environment in which the state and the preferences of one agent is not known to the others apart through the negotiation objects — proposals in PBN or proposals and arguments in ABN. Furthermore, the subsequent state of agents was not solely dependent on the actions they have previously carried out as the negotiation domain incorporated some stochastic elements due to allocation of randomly generated resources to the agents. Under these settings, Först et al. [6] show that ABN performs better than PBN in terms of the optimality of agreements reached between agents and the overall time to reach the agreements.

A major challenge in automated negotiation [7] is that agents usually keep their preference information private to avoid exploitation. When the agents have limited knowledge of the preferences of other agents, they may fail to reach an optimal agreement as they cannot take their opponents’ desires into account. Hindriks et al. [4] argue that two facts about negotiation define the basic dilemma each negotiating agent has to face. On one hand, an agent aims to satisfy its own objective or utility as best as possible, while on the other hand it has to take its opponents’ preferences into account in order to reach an agreement. In order to improve the efficiency of the negotiation and the quality of the outcome, Baarslag et al. claim that agents need to construct a model of their opponents’ preferences to aid them in estimating the information that is kept private [7].

A number of opponent models have been proposed based on different learning techniques and underlying assumptions, and the performances of some of these opponent models have been evaluated on various negotiation problem domains. Baarslag et al. [7] provide two measures of the quality of an opponent model: using an agent’s performance as a benchmark for the model or by directly evaluating the accuracy of the model using similarity measures. The authors investigate the relation between the two methods by considering a large set of state-of-the-art opponent models in different negotiation domains and pinpoint the measures for accuracy that best predict the performance gain when constructing an opponent model. A generic algorithm based on Bayesian learning to model an opponent’s preferences is provided and experimentally analyzed in [8]. The algorithm makes some general assumptions about the preference structure and rationality of the negotiating agents. Hindriks et al. [4] propose a negotiation strategy that learns opponent preferences to avoid exploitation and maximize the chance of an agreement. The performance of the proposed negotiation strategy is then experimentally analyzed in different negotiation domains.
The opponent modeling proposed in [9] uses Bayesian update rule to learn an opponent’s reservation point—the threshold utility of an opponent below which it disagrees. Radu et al. [2] propose a multilateral automated negotiation in which agents adapt their preferences using different bargaining strategies. However, to the best of our knowledge, all of these evaluations were limited to only proposal-based negotiation approach although on different problem domains. The purpose of this research is to evaluate the performance of proposal-based negotiation approach versus argumentation-based negotiation approach on the same problem domain as in [6], but for agents that learn the preferences of their opponents. The rationale for selecting the same problem domain, described in the next section, is to compare and contrast the evaluation results found in [6] which does not use opponent modeling with our work which includes opponent modeling.

2 Problem Domain

We adopt the negotiation problem domain used in [6] which is a production game called Queue-Stack game. The game is a type of general-sum stochastic games. In contrast to zero-sum games where agents’ rewards are always negatively related or coordination games where agents’ rewards are always positively related, general-sum games allow agents’ rewards to be arbitrarily related. The game is stochastic because the subsequent state of agents is not solely dependent on the actions they carry out but also on allocation of randomly generated resources to the agents. Furthermore, the game is a multi-player game. For simplicity, we restrict our work to two agents. Each agent (player or producer) in the game has a queue-based resource store and a stack-based production unit. Each round of the game has two phases: allocation phase and negotiation phase. The goal of the game is to assemble a product and earn a reward by using allocated resources from the resource store and resources acquired from another agent in exchange to some other resources during the negotiation phase. A product is made when the stack is full and contains only one type of resources; otherwise, some of the resources are wasted and no reward is gained. The two phases of the game are described in detail as follows.

2.1 Allocation Phase

During the allocation phase, a fixed but random sequence of new resources of size $M_e$ are en-queued to each agent’s resource store. The resources allocated to each player are independently generated, and are uniformly drawn from the available resource types. We consider only two types of resources represented by 0 and 1. Subsequently, each agent is forced to remove (de-queue) the first $M_d$ elements of its queue and push them onto its stack, maintaining their ordering. Let the state of each agent’s queue and stack after the allocation phase be represented by $Q_i$ and $S_i$ respectively, where $i = 1, 2$ for first agent and second agent respectively.

As a running example, let us consider the following scenario. Let $M_e = 6$, and the resources allocated to the queue of agent 1 be [100101] and to the queue of
agent 2 be [111010] where the first element of each resource sequence represents the head of the queue of the corresponding agent. Furthermore, let \( M_d = 3 \) and the stack capacity of both agents be \(|S| = 5\). Next, the agents push the first three elements of their queues to their stacks. This is done as follows for agent 1. The agent removes the first element (head) of its queue (1) and pushes it on to the top of its stack which is initially empty, then the second element (0) and finally the third element (0). The queue and stack states of agent 1, after the allocation phase, becomes \( Q_1 = [101] \) and \( S_1 = [001] \) respectively where the first element in \( S_1 \) is the top of the stack which is the last (third) element pushed from the queue. Similarly, the queue and stack states of agent 2 becomes \( Q_2 = [010] \) and \( S_2 = [111] \) respectively.

2.2 Negotiation Phase

During the negotiation phase, the agents take turns to make an offer which consists of a set of resources that a proposing agent is willing to give to its opponent in exchange for another set of resources it expects to receive. Prior to making an offer, each agent generates a set of all possible offers, \( O_i \), that it can make during each step of the negotiation process. The generation of \( O_i \) is done in three steps as described below.

First, the set of all possible resources that agent \( i \) is willing to give, \( R_{i,give} \), is generated by removing each possible subset of resources from its queue \( Q_i \). For our example, \( R_{1,give} = \{[0], [1], [01], [11], [101]\} \) which means agent 1 can give one 0, one 1, one 0 and one 1, two 1s, or one 0 and two 1s from its queue \( Q_1 \). The order of the resources to be given is irrelevant; consequently, \([01]\) can be written as \([10]\), and \([101]\) as \([011]\) or \([110]\). Similarly, \( R_{2,give} = \{[0], [1], [00], [10], [010]\}\) for agent 2. Let the state of an agent’s queue after removing one of the resources from \( R_{i,give} \) be denoted by \( Q_i^* \). For our example, \( Q_1^* = [0] \) if agent 1 gives \([11]\) and \( Q_2^* = [1] \) if agent 2 gives \([00]\).

Then, the set of resources that an agent expects to receive, \( R_{i,recv} \), is generated. Since an agent does not know how many and which type of resources it might receive from the other agent, all possible combinations of resource types up to an arbitrary total amount of \( N_r \) are considered to generate the set \( R_{i,recv} \). If we assume \( N_r = M_d = 3 \) in our example, then \( R_{1,recv} = R_{2,recv} = \{[0], [1], [00], [01], [11], [000], [001], [011], [111]\}\) in which the first two elements are possible if the agents receive 1 resource, the next three elements if they receive 2 resources and the last four elements if they receive 3 resources. The resources received from an opponent can be pushed to an agent’s corresponding stack in any desired order, and thus all possible orderings (permutations) of the received resources are considered. For example, if an agent receives \([01]\), it can push the resources as \([01]\) or \([10]\). Let the state of an agent’s stack after pushing one of the resources from \( R_{i,recv} \) be denoted by \( S_i^* \). For our example, after agent 1 receives the resources \([00]\) from agent 2, the received order is the only possible ordering of these resources. The state of the stack of agent 1 after pushing these resources becomes \( S_1^* = [00001] \). Similarly, after agent 2 receives the resources
from agent 1 and pushes them on to its stack, the state of its stack becomes $S_2^* = [11111]$. Finally, the set of all possible offers, $O_i$, of agent $i$ is computed by combining each element in $R_{i,\text{give}}$ with every element in $R_{i,\text{recv}}$. Mathematically, $O_i$ is given by the Cartesian product of $R_{i,\text{give}}$ and $R_{i,\text{recv}}$, $R_{i,\text{give}} \times R_{i,\text{recv}}$, as follows.

$$O_i = \{ \delta_i = (r_{i,\text{give}}, r_{i,\text{recv}}), r_{i,\text{give}} \in R_{i,\text{give}} \text{ and } r_{i,\text{recv}} \in R_{i,\text{recv}} \} \quad (1)$$

For our example, one possible offer that agent 1 can make is $\delta_1 = ([11], [00]) \in O_1$, which means agent 1 is willing to give the resources $[11]$ to agent 2 in exchange for the resources $[00]$. An agent selects an offer that maximizes its utility—a measure of the quality of the agent’s state after making the selected offer, where the state of an agent is defined by the combination of its queue state ($Q_i^*$), stack state ($S_i^*$) and the amount of rewards that it might earn. We adopt the utility function developed in [6] to evaluate the quality of an agent’s state, but slightly change it such that it becomes within the range $[0, 1]$. Accordingly, the utility function of an offer $\delta_{i,s}$ made by agent $i$ in negotiation step $s$ is given by

$$U(\delta_{i,s}) = \frac{1}{k(|Q_i^*| + |S_i^*| - 1)} \sum_{j=1}^{k} n(b_j(r_i)), \quad (2)$$

where $r_i = Q_i^* | S_i^*$ is the sequence of resources of agent $i$ if it makes the selected offer and is the concatenation of the resources in the resulting stack and queue states where the concatenation is done as if the resources of the queue were pushed to the stack starting from the element at the head of the queue; $b_j(r_i)$ represents the $j^{th}$ block of the sequence $r_i$ where a block is defined as a sequence of resources in $r_i$ containing identically typed resource; $k$ is the total number of such blocks in $r_i$; $n(b_j(r_i))$ is the number of resources in block $b_j; |S|$ is the capacity of the stack which we assume to be the same for both agents; and $|Q_i^*|$ is the number of resources agent $i$ has in its queue after making the offer which is not a fixed value. For our example, if agent 1 selects the offer $\delta_{1,s} = ([11], [00])$ in some negotiation step $s$, then $r_1 = Q_1^* | S_1^* = [0]||[00001] = [000001]$; $|Q_1^*| = 1$ and there will be $k = 2$ blocks: $b_1(r_1) = [00000]$ with $n(b_1(r_1)) = 5$ and $b_2(r_1) = [1]$ with $n(b_2(r_1)) = 1$; and the utility of offer $\delta_1$ will be $U(\delta_{1,s}) = \frac{1}{2(1+5-1)(5+1)} = 0.6$. The utility function in equation 2 computes the average reward that can be achieved if agent $i$ makes the selected offer. It should be noted that the utility function is only defined if $|r_i| \neq 0$ where $|r_i|$ is the number of resources in $r_i$; otherwise, the value of the utility function is assumed to be zero which is reasonable since agent $i$ does not have any resource left in its queue and stack. Moreover, the utility function captures interesting features of the game as it gives more weight to offers made by an agent that results in giving as few resources as possible to another agent, retaining longer blocks of identically typed resources in the agent’s queue especially near the head, and receiving as many resources as possible which matches most with what the agent has near the top its stack.

After one of the agents, say agent 1, selects an offer $\delta_{1,0} \in O_1$ with maximum utility, it initiates negotiation with agent 2 by proposing the offer. Upon receiving
the offer, agent 2 has three available locutions (response messages) to choose from: *agree*, *reject* or *propose*. If agent 2 agrees with the offer, it utters an *agree* message to agent 1. The outcome of a successful negotiation is a deal. The agents then exchange the sets of resources specified in the offer as part of the deal, push the acquired resources on to their stack in any desired order and the negotiation terminates. If agent 2 does not agree with the offer from agent 1 because it has a better offer $\delta_{2,1} \in O_2$, it makes a *propose* message with a counter-offer to agent 1 in the next step of the negotiation process. If agent 2 does not agree with the offer from agent 1 because it has previously received a better offer from agent 1, it sends a *reject* message prior to proposing a counter-offer. Furthermore, in the case of ABN, we consider two more response messages: *justification* and *critique*. In such case, if agent 2 cannot give the resources requested by agent 1 because it does not have them all, agent 2 sends a *justification* message to agent 1 before making a counter-offer. If agent 2 does not want to receive the resources offered by agent 1 because it does not improve its current state under any circumstances, it utters a *critique* message to agent 1 prior to making a counter-offer. Upon receiving a *justification* message from agent 2, agent 1 modifies its set of all possible offers, $O_1$, by removing all offers which contain the same $r_{1,recv}$ as in the offer it has previously made. Similarly, if agent 1 receives a *critique* message from agent 2, it modifies its set of all possible offers, $O_1$, by removing all offers which contain the same $r_{1,give}$ as in the offer it has previously made.

If any of the agents do not want to negotiate because they are better off with the randomly allocated resources or if agents cannot find an agreement, the default deal, $\delta_{i,\text{default}}$, is enforced and the resource situations of the agents remain unchanged. In such cases, the negotiation terminates and the rewards that the agents can earn in the subsequent game round depends on the random resource sequence allocated to them. If the agents choose to negotiate and the negotiation ends in a deal, the agents exchange the corresponding resources and push them on to their stack in an order that maximizes their utility. An agent earns a unit reward for each product it successfully assembles. Then, the next round of the game starts with the allocation phase and the game continues. For our example, if agent 2 agrees on the offer $\delta_{1,s} = ([11],[00])$ from agent 1 in some negotiation step $s$, the offer becomes a deal, and the agents exchange the resources and push them on to their corresponding stacks. Consequently, the state of the stack of agent 1 becomes [00001]. Since the stack of agent 1 is full at this stage and it contains different resources, the stack is reset by retaining only similar resources near the top of the stack 0000 and removing (wasting) the remaining ones 1. Similarly, the stack of agent 2 becomes [11111]. But in this case, the stack is full and consists of only one type of resource, and thus a product is considered to be fully assembled or produced. As a result, agent 2 earns a unit reward for the product it has made and its stack is reset by removing all of the elements. Therefore, the state of the stack of agent 1 after pushing the resources [00] is $S_1^* = [00000]$, and the stack of agent 2 after pushing the resources [11] becomes $S_2^* = []$ which is empty. Agent 2 has earned one reward, though. During each negotiation phase, agents can learn to model the resource preferences of
their opponents from the locutions they exchange so as to make offers which have higher chances of reaching in agreements, and this is the topic of the next section.

3 Learning an Opponent Model

For an agent to learn about the preference of resources of its opponent, it has to estimate the queue and stack states of the opponent. These states can be estimated from the offers made by the opponent. Moreover, ABN agents can refine their opponent models using argument messages they receive. In this section, we discuss the main contributions of our work for the Queue-Stack game. First, we develop heuristic algorithms to model an opponent’s queue and stack states. Then, using these models, we provide a learning algorithm that can be used by an agent to estimate its opponent’s utility of preferred offers during each step of the negotiation process.

3.1 Estimating an Opponent’s Queue State

The following simple heuristic algorithm has been used to estimate the queue state of an opponent, \( Q_o \). In each round of the game, when an agent receives the first offer from its opponent, it evaluates the non-dominant resource type in the sequence of resources that the opponent is willing to give, \( r_{o,\text{give}} \). For example, the non-dominant resource type in \([011]\) or \([11]\) is 0 while in \([01]\) or \([\_]\) there is no non-dominant resource type. If there is a non-dominant resource type in \( r_{o,\text{give}} \), the agent initializes \( Q_o \) to resources of size \( N_r \) all of which have the same type as the non-dominant resource type. The intuition behind this heuristic is that the non-dominant resource type in \( r_{o,\text{give}} \) is the dominant resource type that the opponent is trying to keep in its queue. If there is no non-dominant resource type in \( r_{o,\text{give}} \), the agent will not be able to learn from the offer and it will initialize \( Q_o \) to resources of size \( N_r \) which consists of the same type where the type is selected randomly. Furthermore, when an ABN agent receives a justification argument from its opponent in any step of the negotiation process, the agent updates the model of its opponent’s queue by changing the resource type of all resources in \( Q_o \) to the non-dominant resource type, if any, in \( r_{o,\text{give}} \) that the opponent said it cannot give. If there is no non-dominant resource type, then the agent keeps \( Q_o \) intact.

3.2 Estimating an Opponent’s Stack State

In a similar fashion to estimating the state of an opponent’s queue state, the following heuristic algorithm has been used to estimate the stack state of an opponent, \( S_o \). In the first round of the game, when an agent receives the first offer from its opponent, it evaluates the dominant resource type in the sequence of resources that the opponent wants to receive, \( r_{o,\text{recv}} \). If there is a dominant resource type in \( r_{o,\text{recv}} \), the agent initializes \( S_o \) to resources of size \(|S| - 1\) all of
which have the same type as the dominant resource type. The rationale behind
this heuristic is that the dominant resource type in $r_{o,recv}$ is the resource type
that opponent has near the top of its stack. If there is no dominant resource type
in $r_{o,recv}$, the agent will not be able to learn from the offer. In such case, the agent
will initialize $S_o$ to resources of size $|S| - 1$ and of type opposite to the resource
type in $Q_o$ in order to avoid overestimating the resource situation of its opponent
and thus making worse offers. In subsequent rounds of the game, however, $S_o$ is
simply updated to have the same resource type as in $Q_o$ of the previous round.
The reason for this is that resources near the head of the queue of the opponent
in the previous round of the game are pushed to its stack during the allocation
phase of the next round. Moreover, whenever an ABN agent receives a critique
argument from its opponent during the negotiation process, the agent updates
the model of its opponent’s stack by changing the resource type of all resources
in $S_o$ to the non-dominant resource type, if any, in $r_{o,recv}$ that the opponent said
it does not want to receive. If there is no non-dominant resource type, then the
agent keeps $S_o$ intact.

### 3.3 Estimating an Opponent’s Utility of Preferred Offers

An agent can learn about its opponent’s preference of resources by updating its
belief about the expected utility of offers that it’s opponent might be willing to
accept. Then, the agent can make offers that provide the opponent at least as
good utility as the expected value using the estimated queue and stack states, $Q_o$
and $S_o$. Hindriks et al. [8] and Zeng et al. [9] propose generic Bayesian learning
algorithms for problem domains different than ours. We adopt these algorithms
to estimate an opponent’s utility of offers during each step of the negotiation
process of the Queue-Stack game in subsequent discussions.

An agent’s partial belief about the utility value of the offers that its opponent
prefers can be represented by a set of $m$ hypotheses, \{$h_k$, for $k = 1, 2, \ldots, m$\}. Since no a priori knowledge about the probability distribution of each hypoth-
esis $P(h_k)$ is available, a uniform distribution can be initially assigned to each
hypothesis. For example, we can consider ten hypotheses as follows: $h_1 = 0.1$, $h_2 = 0.2$, $h_3 = 0.3$, $h_4 = 0.4$, $h_5 = 0.5$, $h_6 = 0.6$, $h_7 = 0.7$, $h_8 = 0.8$, $h_9 = 0.9$ and
$h_{10} = 1.0$, initially with a uniform probability distribution of $P(h_k) = \frac{1}{10} = 0.1$ for each hypothesis. The meaning of each hypothesis is that the utility of pre-
ferrfered offer of an opponent has a value specified by the hypothesis with its
responding probability (belief). Since the utility function in equation 2 has
values in the range $[0, 1]$ and since agents use concession-based strategy which
is a subtype of rational strategy, the predicted utility of an offer $\delta_{j,s}$ can be
estimated using a linear function as follows:

$$U'(\delta_{j,s}) = 1 - c * s,$$

where $s$ is the current negotiation step and $c$ is some integer constant $0 < c < 1.$
This function monotonically decreases from 1 (maximum utility) to 0 (mini-
mum utility) as negotiation between the agents continues, and thus is applicable
for concession-based negotiation strategy which is discussed in section 4. For example, when agent 1 receives the first offer $\delta_{2,s}$ from agent 2, $s = 1$ and if we assume $c = 0.05$, the utility of the offer as predicted by agent 1 would be $U'(\delta_{2,1}) = 1 - (0.05 \times 1) = 0.95$. That is, agent 1 predicts that the utility of the first offer made by agent 2 has a utility value of 0.95 for agent 2. The likelihood $P(\delta_{j,s}|h_k)$ that an offer $\delta_{j,s}$ made by agent $j$ to agent $i$ in negotiation step $s$ might have been proposed using hypothesis $h_k$ (or the likelihood that the utility of the offer is the value given by $h_k$) can be defined as a probability distribution over the values of the hypotheses with $U'(\delta_{j,s})$ as the mean value. Mathematically, this conditional probability modeled as a Gaussian distribution is given by

$$P(\delta_{j,s}|h_k) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(h_k - U'(\delta_{j,s}))^2}{2\sigma^2}},$$

(4)

where $\sigma$ is the spread of the probability distribution and defines the certainty of an agent about its opponent’s negotiation strategy. Lower values of $\sigma$ means higher level of certainty and increases the learning speed since hypotheses predicting an incorrect utility value of an offer would get assigned an increasingly lower probability. Higher values of $\sigma$ have the opposite effect. When agent $i$ receives a new offer from agent $j$, agent $i$ can update the probability of its hypotheses by calculating the posterior belief of each hypothesis $h_k$ using Bayes’ rule as follows.

$$P(h_k|\delta_{j,s}) = \frac{P(h_k)P(\delta_{j,s}|h_k)}{\sum_{k=1}^{m} P(h_k)P(\delta_{j,s}|h_k)}$$

(5)

Then, the probability of each hypothesis $P(h_k)$ is updated to the value of $P(h_k|\delta_{j,s})$. Agent $i$ uses the updated probability distribution of its hypothesis set to compute the expected utility value of its opponent’s preferred offers using equation 6.

$$\bar{U}_{s+1} = \sum_{k=1}^{m} h_k P(h_k)$$

(6)

Agent $i$ then selects a counter-offer $\delta_{i,s+1}$, during the next step of the negotiation process, that maximizes its own utility while at the same time provides its opponent at least as good utility as the expected value. This is done by evaluating the utility of each of the offers agent $i$ considers (from the perspective of its opponent) using the resources it is willing to give ($r_{i,\text{give}}$) in each offer, and the estimated queue and stack states of its opponent ($Q_o$ and $S_o$). The utility function used for evaluating the offers of agent $i$, from the perspective of its opponent, is similar to the function in equation 2 but with the normalization term being replaced by $k(|Q_o| + N_r + |S| - 1)$. During each step of the negotiation process, agent $i$ is expected to gain more accurate estimate of its opponent’s preference of resources and make offers with increasingly higher chances of entering into an agreement. Furthermore, learning agents can minimize negotiation steps since they exclude offers that are below the expected utility value of their opponents’ preferred offers.
4 Negotiation Strategy

Each agent generates a set of possible deals, $O_i$, which it will propose to its opponent one after another, starting with the offer with highest utility, followed by offers with descending utilities. Such a negotiation strategy is called a concession-based strategy. In concession-based strategy [8], an agent makes its next offer that has a decreased utility compared to a previously proposed offer. Furthermore, we allow agents to defer agreeing to offers they receive from their opponents that improve the utility of their current states as long as the agents themselves are able to make their own offers that have higher utility than what they have received so far.

After agent $j$ has proposed an offer $\delta_{j,s}$ to agent $i$ during the $s^{th}$ step of the negotiation process, agent $i$ updates its opponent model and computes the expected utility of its opponent’s preferred offer $\bar{U}_{s+1}$. It then computes the best offer it is able to propose $\delta_{i,s+1}$ which has not been offered yet and which provides its opponent at least as good utility as the expected value, as its counter-offer in the next step of the negotiation. If agent $i$ cannot find any counter-offer better than its default deal, it is forced to terminate the negotiation either by accepting the best offer $\delta_{j,best}$ of all offers received from agent $j$ in earlier steps of the negotiation or by accepting its default deal $\delta_{i,default}$. If agent $i$ finds a counter-offer $\delta_{i,s+1}$ which is better than its default deal but has lower utility than $\delta_{j,best}$, agent $i$ accepts the best previous offer from agent $j$. Otherwise, agent $i$ proposes its counter-offer $\delta_{i,s+1}$ to agent $j$, and the negotiation continues. Agent $i$ can precede its counter-offer by rejecting $\delta_{j,s}$, the latest offer from agent $j$, if the received offer is not the offer with the highest utility of all offers received so far. Moreover, in the case of ABN, agent $i$ can precede its counter-offer by making arguments, if any, on the latest offer from agent $j$. The negotiation strategy executed by agent $i$ upon receiving an offer from agent $j$ is shown in Algorithm 1.

5 Evaluation Criteria

We use criteria similar to the ones used in [6] to evaluate the performances of PBN and ABN agents with and without opponent modeling. These criteria are summarized below.

- Rewards: the total amount of rewards earned (products assembled) by both agents in one round of the game.
- Negotiation Steps: the total number of steps agents negotiate in one round of the game until they agree or disagree. This criterion determines the speed of negotiation.
- Communication Overhead: the total amount of communication data units exchanged during the negotiation phase in one round of the game. We defined reject, justification and critique messages to have one communication data unit while propose and agree carry as much communication units as the total resources contained in the proposed and agreed offers respectively.
Algorithm 1: Negotiation Strategy
1: receive offer $\delta_{j,s}$
2: update $Q_o$ and $S_o$
3: compute $\bar{U}_{s+1}$
4: $\delta_{i,s+1} \leftarrow$ next best offer from $O_i$ with $U^*(\delta_{i,s+1}) \geq \bar{U}_{s+1}$
5: $\delta_{j,best} \leftarrow$ best offer from agent $j$ so far
6: if $\delta_{i,s+1} = \emptyset$
7: if $U(\delta_{i,default}) \geq U(\delta_{j,best})$
8:-agree $\delta_{i,default}$
9: else
10: agree $\delta_{j,best}$
11: else
12: if $U(\delta_{i,s+1}) < U(\delta_{j,best})$
13: agree $\delta_{j,best}$
14: else
15: if $U(\delta_{j,s}) < U(\delta_{j,best})$
16: send reject message
17: if cannot give $r_{j,recv}$ in $\delta_{j,s}$
18: send justification message
19: if do not want to receive $r_{j,give}$ in $\delta_{j,s}$
20: send critique message
21: propose $\delta_{i,s+1}$

- Acceptance Rate: the percentage of negotiations per game which ended up in an agreement.
- Degree of Optimality: the percentage of deals per game which are the same as the optimal deals. The optimal deals are computed using an unbiased mediator which does not take part in the game as follows. During the negotiation phase of each round of the game, each agent reveals to the mediator, its queue and stack states as well as the set of all offers that it has generated but has not proposed yet. The mediator then uses these complete information about the agents and finds Pareto-optimal deal for both agents which maximizes the product of both agents’ utilities using the Nash bargaining solution [5]. A deal is Pareto-optimal if there is no other deal which improves the utility of one agent without decreasing the utility of at least one of the other agents.

6 Experiments
A total of 50 experiments or games have been conducted for each type of the negotiating agents: PBN agents without opponent model (OM), ABN agents without OM, PBN agents with OM and ABN agents with OM. All of the experiments have been performed under the same game parameters ($|S| = N_r = 3$, $M_d = 6$, $M_e = 9$) and OM parameters ($m = 10$, $c = 0.05$, $\sigma = 0.75$). Each game run for 20 rounds. In each round, two random sequence of resources of size $M_e$ are generated and then allocated to the queues of agents 1 and 2 respectively of each of the four negotiating agent types. Furthermore, in each round, the agent
which starts the negotiation process has been randomly selected for each agent type.

The rewards, negotiation steps and communication overhead have been computed per each game round and then averaged over the total number of rounds in the game. These resulting values are stored and then further averaged over the total number of experiments. The degree of optimality and acceptance rate are computed per each game and then averaged over the total number of experiments. That is, the final computed rewards, negotiation steps and communication overhead values are mean ($\mu$) values per game round while the degree of optimality and acceptance rate values are mean values per game. Moreover, the standard deviation ($\sigma$) of each evaluation criterion over the total number of experiments has been computed. The results of these experiments is presented in Tables 1 and 2 of section 7.

7 Results and Evaluation

It can be seen from Table 1 that the mean values of rewards earned by both agents, degree of optimality of deals and acceptance rate of offers of PBN agents are almost equal to the corresponding values of ABN agents when the agents do not use opponent models. These results are also similar when the agents use opponent models as it can be observed from Table 2. Nonetheless, when agents do not learn the preference of their opponents, ABN agents compared to PBN agents reduce the negotiation steps by about 35% (about 8 steps) and the communication overhead by about 29% (about 34 communication data units). That is, without opponent modeling, ABN agents reach in a deal or no-deal faster than PBN agents besides using fewer communication data units. However, these performance improvements of ABN agents over PBN agents is not significant when the agents learn the preference of offers of their opponents since the negotiation steps and communication data units are reduced by 0.3 steps and 2 data units respectively.

Moreover, the negotiation steps and communication overhead of agents that use OM compared to agents that do not use OM reduced by up to 84.5% and 78.9% respectively. However, the degree of optimality of deals and acceptance rate of offers of agents that use OM have reduced as well. Agents with OM earned slightly more rewards than agents without OM, though. Therefore, agents that learn the preference of offers of their opponents earned slightly more rewards by agreeing on suboptimal deals at very fast rate and thus were able to drastically reduce the communication overhead.

Fürst et al. [6] showed that argumentation-based negotiation outperforms proposal-based negotiation on the same problem domain using similar evaluation criteria when the agents do not model the preferences of their opponents. This is in accordance to the results observed in our work as discussed previously in this section. However, we have also shown that both PBN and ABN perform almost equally well if the agents learn the preferences of offers of their opponents. To make some sense out of these contrasting results, let us consider some real-world
analogy. In commerce, if buyers and sellers are only concerned with maximizing their own utilities, maximum possible profit for sellers and lowest possible price for buyers, the agents need to negotiate more before they can agree on the price of goods which are optimal to both. Furthermore, buyers who can argue better (analogous to ABN agents) will get better bargain prices. If, on the other hand, the sellers try to model the buying capacity and preferences of goods of previous buyers, and use that model to offer goods with agreeable prices to future buyers, the agents will negotiate less and agree more. In the latter case, the bargaining power of agents will play almost no role in reaching an agreement.

### Table 1. Experimental Results for Agents Without Opponent Models

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>PBN Agents without OM</th>
<th>ABN Agents without OM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewards</td>
<td>$\mu = 4.2$ $\sigma = 0.28$</td>
<td>$\mu = 4.2$ $\sigma = 0.27$</td>
</tr>
<tr>
<td>Negotiation Steps</td>
<td>$\mu = 22.0$ $\sigma = 3.85$</td>
<td>$\mu = 14.4$ $\sigma = 1.68$</td>
</tr>
<tr>
<td>Communication Overhead</td>
<td>$\mu = 117.0$ $\sigma = 17.41$</td>
<td>$\mu = 83.0$ $\sigma = 8.64$</td>
</tr>
<tr>
<td>Degree of Optimality</td>
<td>$\mu = 56.8$ $\sigma = 12.44$</td>
<td>$\mu = 56.5$ $\sigma = 11.67$</td>
</tr>
<tr>
<td>Acceptance Rate</td>
<td>$\mu = 99.2$ $\sigma = 1.83$</td>
<td>$\mu = 100.0$ $\sigma = 0.00$</td>
</tr>
</tbody>
</table>

### Table 2. Experimental Results for Agents with Opponent Models

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>PBN Agents with OM</th>
<th>ABN Agents with OM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewards</td>
<td>$\mu = 4.6$ $\sigma = 0.17$</td>
<td>$\mu = 4.6$ $\sigma = 0.18$</td>
</tr>
<tr>
<td>Negotiation Steps</td>
<td>$\mu = 3.4$ $\sigma = 1.03$</td>
<td>$\mu = 3.1$ $\sigma = 0.98$</td>
</tr>
<tr>
<td>Communication Overhead</td>
<td>$\mu = 24.7$ $\sigma = 6.05$</td>
<td>$\mu = 22.8$ $\sigma = 5.53$</td>
</tr>
<tr>
<td>Degree of Optimality</td>
<td>$\mu = 34.9$ $\sigma = 8.86$</td>
<td>$\mu = 35.4$ $\sigma = 11.7$</td>
</tr>
<tr>
<td>Acceptance Rate</td>
<td>$\mu = 88.7$ $\sigma = 3.72$</td>
<td>$\mu = 90.1$ $\sigma = 3.24$</td>
</tr>
</tbody>
</table>

### 8 Conclusion

In this work, four types of negotiating agents have been presented and the performance of each type of agents have been evaluated under identical experimental settings by letting two agents of each type play in a production game which is a type of partially observable, general-sum, stochastic game. Under these settings, we have shown that while ABN agents perform better in terms of decreased communication overhead and negotiation duration when compared to PBN agents without opponent models, both agent types perform almost equally well if they learn the preferences of offers of their opponents. Future work may include extensive evaluations of these agent types in different negotiation problem domains and using different learning techniques.
References

Evaluation of Color Association when Receiving a Mobile Notification

Victor Winnhed
Department of Computing Science
Umeå University, Sweden
id11vwd@cs.umu.se

Abstract. This study examines how people associate colors to mobile applications. The aim of the study was to find out whether we tend to associate the same colors to the same applications or not and to give the designers guidelines to help them when designing user-interfaces and application logos. A group of 15 students, all of them smartphone users, took part in the user tests. Six different colors showing on a smart watch through colored LED-lights were examined during the user tests, one color at a time. The users had to associate each color to a mobile application. The results show that we tend to associate some of the tested colors to mobile applications. Most of the participants associated the color blue to the Facebook application and the color green to the SMS-application.

1 Introduction

There are several wearable technology solutions on the market, like the Apple-Watch [1] and Moto-360 [2]. According to the study in [3], interaction designers struggle to downsize interactions originally designed for smartphones to fit wearable displays. Health and fitness trackers are the most used wearables [3] and some of these wearables do not have a display; there are smart watches and bracelets [4] [3] that only use colored LED-lights to present mobile notifications. These LED-lights indicate a specific color depending on the received notification. Due to the lack of a display the users are forced to identify a mobile notification by a single color. The study in [4] shows that receiving notifications by indicating LED-lights is both subtle and non-intrusive.

Some studies show that people tend to associate colors to emotions [5] [6]. The results from the study in [6] show that all humans associate colors to emotions at all ages. It also shows that colors and emotion-preferences changes with age. Furthermore, there are studies showing that colors can be associated to brands [7] [8], e.g. some people might associate the color red to Coca-Cola.

The aim of this study is to investigate if smartphone users tend to associate the same colors to the same mobile applications. Furthermore, according to the study in [7], people were asked to associate colors to brands by imagining the colors themselves and discussing the meaning of each color. The result showed that color is a powerful tool for identifying brands in general. In this study, the
test method is different to [7]. In this study the users will associate colors to mobile application brands with the help of a smart watch without a display. This study is intended to give the designers guidelines when choosing colors for their applications logos and user-interfaces.

The study is conducted by doing user-tests, where the users get to use a smart watch with indicating LED-lights. The tests are held inside a quiet room at the University of Umeå and consists of two tasks. The first task consists of users receiving a number of mobile notifications and the watch is indicating the notifications by showing different colors. When the smart watch is showing a color the users are asked to connect that color to a mobile notification. A mobile notification can either be a notification from a mobile application or a notification when receiving a text message, receiving an email or receiving a call. The second task consists of a form on which the users are asked to answer several questions about themselves like gender and age. According to the study in [6] color associations and emotions changes with age, that is why we want to know the age of those who participates. We want to have a homogeneous test group.

The test group consists of students at the University of Umeå in the age group of 20-30 and all of them are smartphone users. Another interesting aspect is to ask the users how often they use popular applications like Facebook and Instagram. This might impact how they associate colors to applications and is an interesting aspect to investigate further. We are also asking the users how often they use the applications SMS, Email and WhatsApp because these applications were the most popular according to the study in [9]. These applications do also generate a high amount of notifications [9]. The smart watch that is going to be used during the user tests is called a Nevo smart watch and more information about the watch can be found in the section below.

1.1 The Nevo Smart Watch

The Nevo smart watch is a fairly new innovation that was crowdfunded\textsuperscript{1} on the site www.indiegogo.com in the fall of 2014. This watch does not have a digital display like a regular smart watch, it looks like a regular analog watch, but when something happens on the phone e.g. a call, a text message, an email or any other mobile notification the Nevo vibrates and shows one of 6 colored LEDs that are placed around the watch face. There are a total of 6 colors: orange, red, blue, yellow, light green and dark green. Mobile notifications can be assigned to any one of these colors.\textsuperscript{2}

1.2 Difficulties

There are some pitfalls in our study that are kept in mind. There are brands that are easy to connect to the same color. For instance, both Twitter and Facebook

\textsuperscript{1} The practice of funding a project or venture by raising monetary contributions from a large number of people.

\textsuperscript{2} More information about the Nevo watch at www.nevowatch.com.
uses blue as their main color. If a user asks us during the user-tests if it is possible to connect multiple applications to the same color, we are letting them do that. During the user tests we are using a Nevo smart watch which only has six colors. So there are still a lot of colors and different shades of colors that can be tested. The tests are held inside because there might be a difference in how people chose if the tests were as an example held outside. All people are different, we all have different backgrounds but the test group is homogeneous in the aspects of age, employment and that all of the participants are smartphone users.

2 Method

In order to investigate how people associate colors to mobile notifications a user-test was conducted, which consisted of two main tasks. The user tests were held at the University of Umeå in a quiet and empty classroom.

We started the first task by explaining how it would be conducted. The users were told that the test was anonymous, that they could chose to leave at any time and that the study was made for educational purpose for the course Student conference. We told the users to read the following instructions carefully (the instruction has been translated from Swedish to English):

- “You are going to receive notifications, and when receiving a notification the Nevo Watch is going to indicate those notifications by showing colored lights.”
- “For every notification the Nevo Watch will be showing one indicating color three times while vibrating. Write down what application you associate the most to that specific color.”
- “If there are two applications that you associate just as much to one color, please write down both of them. Good Luck!”

Furthermore, we asked the users to put the Nevo smart watch on their wrist. After the users had put the smart watch on we started to send notifications. We told the users to write down what applications they thought about when an indicating colored LED-light was showing on the face of the Nevo watch. We sent one notification at a time and let the users write down what application they thought about. We tested a total of six colors which are all of the colored LED-lights on the Nevo smart watch. We always used the same order of colors when doing user tests; the order that was tested for each user was light green, yellow, orange, dark green, red and blue. There were no specific reason why we chose that order of colors other than that we wanted to be consistent.

The second task consisted of a form on which the users were asked to answer several questions. First of all the users had to write down their names, gender and what type of smartphone they are currently using. We asked about age and if the were smartphone users to get a homogeneous group. The gender aspect might be interesting if we can notice a difference in association to colors between men and women.
Fig. 1. A figure of the form given to the test subjects during the first task.

Fig. 2. The second part of the second task, where the users had to fill out how often they use specific applications. (The text shown in the figure is translated from Swedish to English.)
Then the form listed a number of popular application that usually sends a lot of notifications. The users were asked to write down how often they use all of these applications on a scale from never to every day. The aim of this task was to find out more about the users preferences, since this might have an impact in how they associate colors to applications. The applications the participants had to rank during the second part of the form were:

- Facebook-Messenger
- Facebook
- Twitter
- Instagram
- Snapchat
- Tinder
- Phone (how often they received a call)
- Text (how often they received a text message)
- Email-client
- WhatsApp
- KiK
- Jodel

Figure 2 shows an example of the outline from part two of the form. This is how the questions about the applications were asked.

All the answers from Task 1 were gathered and saved using Google Drive. The form used for Task 2 was created using Google Forms. All the data from the user tests were then gathered and analyzed.

3 Results

In total 15 people participated in the test, out of which 5 were women and 10 were men. All of the test participants were smartphone users, students and in the age between 23 and 30 years old.

<table>
<thead>
<tr>
<th>Color</th>
<th>Most popular applications</th>
<th>Votes in percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Green</td>
<td>SMS</td>
<td>66%</td>
</tr>
<tr>
<td>Yellow</td>
<td>Snapchat</td>
<td>40%</td>
</tr>
<tr>
<td>Orange</td>
<td>Jodel</td>
<td>20%</td>
</tr>
<tr>
<td>Dark green</td>
<td>SMS</td>
<td>40%</td>
</tr>
<tr>
<td>Red</td>
<td>Calender</td>
<td>26%</td>
</tr>
<tr>
<td>Blue</td>
<td>Facebook</td>
<td>66%</td>
</tr>
</tbody>
</table>

Table 1. A table showing the most common answer for each color. Evaluation of Task 1.

Overall, the colors with the most associations to the same applications were Light green and blue. As seen in Table 1, 66% of the participants associated the color light green to the SMS-application. Most of the participants also associated the color dark green to the SMS application. The Facebook application was
Table 2. A table showing the every day usage of the most popular applications from Table 1. Most of the participants uses the SMS-application every day.

The results show that most of the participants 66% associated the color light green to the SMS-application. 93% of the participants are using the SMS-application every day according to the results. The majority of the participants were iPhone users. As seen in Figure 3 both the SMS-icon and the message-bubbles are in a lighter green color in iOS. The participants also associated the color dark green to the SMS-application. However, 33% of the participants associated the color dark green to the Phone-application (When someone calls you). For Android users, the SMS-application has different colors depending on what phone and what theme they are using. Therefore, this might have an affect on the result.

66% of the participants in the user tests associated the color blue to the Facebook-application. Furthermore, 86% of the participants said that they were using Facebook every day. Other popular applications that the participants associated to the color blue were LinkedIn, Twitter and Instagram. All of these Use different shades of blue in their brand logo and/or in the application.

The color orange was the color that had the largest spread of applications. The most common was the application Jodel, 20% of the participants associated the color Orange to Jodel. That is only 3 out of 15 participants. Moreover, none of the participants are using Jodel on a daily basis.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Every day usage in percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>93%</td>
</tr>
<tr>
<td>Snapchat</td>
<td>66%</td>
</tr>
<tr>
<td>Jodel</td>
<td>0%</td>
</tr>
<tr>
<td>Calender</td>
<td>53%</td>
</tr>
<tr>
<td>Facebook</td>
<td>86%</td>
</tr>
</tbody>
</table>

Fig. 3. A figure showing the SMS icon and the message-bubbles in iOS.
There is a big difference in the daily usage of Jodel compared to the daily usage of the SMS-application. There is also a great difference in how many of the participants who associated the SMS-application to a specific color compared to Jodel. The percentage of the chosen colors seems to correlate with the usage of the applications. This means that the more an application is used the greater are the chances to associate that one application to a specific color. The color associated to the applications also seem to correspond to the color of the applications icon or/and a color used in the application.

According to the results, an application that produces a high amount of notifications and that are frequently used, have a higher change of being associated to a specific color. This result can help designers decide what color or colors to use when designing a user-interface or logo for a mobile application. Choosing a color that is not busy by another application that is frequently used is beneficial.

4.1 Limitations

One limitation of our study was the order of the colors. When doing the user tests we chose to show the yellow color before the orange color. Many of the participants thought that the yellow color looked orange and therefore these associations might not be correct. Right after the yellow color was showed, the orange color was presented to the participants. This made some of the participants confused and often associated the same application to both yellow and orange. Another limitation is that the user tests only had fifteen participants, which gave us a limited amount of answers. The fact that most participants were iPhone users is a limitation, the results might have been different if more participants were Android users.

5 Conclusion

Do most smartphone users associate the same colors to the same notifications? From this study we can determine that this apply to some of the colors that we have tested. This might be because applications that we frequently use influences us subconsciously and that color can be a powerful tool for identifying brands. In summary, our study study suggests that there is a correlation between the frequency of usage of the applications and the color associated to that application. We have to keep in mind that this study only consisted of 15 participants, therefore further studies are required before any general conclusions can be made.

6 Future Work

Suggested future work would be to investigate this further. Use different and more shades of colors to see if there are significant differences. To do user tests in different situations and environments are also a suggestion. People might
behave differently when tested outside as an example. According to the study in [6] color associations and emotions changes with age. One suggestion is to try this on other age groups. Also, when doing user tests, the colors can be presented differently to avoid confusion among the participants.

References

## Author Index

<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abedin, Md Reaz Ashraful</td>
<td>1</td>
</tr>
<tr>
<td>Bråne, Arvid</td>
<td>13</td>
</tr>
<tr>
<td>Englund, Lars</td>
<td>31</td>
</tr>
<tr>
<td>Gustafsson, Jonas</td>
<td>41</td>
</tr>
<tr>
<td>Hübsch, Albin</td>
<td>61</td>
</tr>
<tr>
<td>Holmgren, Johan</td>
<td>51</td>
</tr>
<tr>
<td>Jonsson, Anna</td>
<td>69</td>
</tr>
<tr>
<td>Shehabeldin, Heba</td>
<td>81</td>
</tr>
<tr>
<td>Sutherland, Alexander</td>
<td>95</td>
</tr>
<tr>
<td>Weldemariam, Dawit Kahsay</td>
<td>107</td>
</tr>
<tr>
<td>Winnhed, Victor</td>
<td>121</td>
</tr>
</tbody>
</table>