

Aufgabenstellung für eine Abschlussarbeit

Optimierung von Mixed Reality Anwendungen durch Graph Codes

Eine Erweiterung und Anwendung des GMAF Frameworks.

Prof. Dr.-Ing. Stefan Wagenpfeil

Lehrbeauftragter für Multimedia Informationssysteme und Information Retrieval
Prüfer für Informatik und Wirtschaftsinformatik (Bachelor / Master)

am Lehrgebiet Multimedia und Internetanwendungen

Prof. Dr.-Ing. Matthias Hemmje

Fernuniversität Hagen

Universitätsstrasse 1, D-58097 Hagen

Raum: 1 F09, Gebäude 3 (Informatikzentrum, Universitätsstr. 1)

Tel.: +49 2331 987-304

Fax: +49 2331 987-4487

Internet: <http://www.fernuni-hagen.de>

Aufgabenstellung für: Bachelor und Masterarbeiten, Fachpraktika

Bearbeitungszeitraum: 2023

Betreuer: Prof. Dr.-Ing. Stefan Wagenpfeil

E-Mail: stefan.wagenpfeil@fernuni-hagen.de

Skype: stefan_wagenpfeil

Lehrgebiet Multimedia und Internetanwendungen:

- Daten- und Dokumentenmanagement
- Informations- und Wissensmanagement
- Multimedia Informationssysteme und Datenbanken
- Informationsvisualisierung
- Information Retrieval
- Visuelle Mensch-Maschine-Interaktion
- Content- und Wissensmanagement im Internet
- Semantic Web
- Digitale Langzeitarchivierung
- Virtuelle Forschungsumgebungen
- Big Data Analyse
- Analyse natürlicher Sprache
- Berufliche Weiterbildung und E-Learning
- Industrie 4.0 und "Factories of the Future"

Die nachfolgenden Beschreibungen für die Abschlussarbeit im Bereich Informatik benennen die nötigen Aufgaben und Teilaufgaben und sollen ebenfalls als Übersicht und Leitfaden für die zu erbringenden Leistungen dienen. Zur Bearbeitung der nachfolgenden Aufgaben werden folgende Fähigkeiten und Kenntnisse vorausgesetzt:

- **Wissenschaftliches Arbeiten:** strukturiertes, analytisches Vorgehen auf Basis der gültigen Konventionen für Aufbau, Zitierweise und inhaltliche Bearbeitung von forschungsrelevanten Themen sind die Voraussetzung für das Schreiben einer Bachelor- oder Master-These. Hierzu gehören die eigenständige Literaturrecherche, die selbstständige Aneignung von Wissen und die Umsetzung von theoretischen Konzepten in die Praxis.
- **Programmiersprachen / Beschreibungssprachen:** Java, XML. Von Vorteil sind Kenntnisse im Umgang mit Eclipse und ein grundlegendes Verständnis von Konzepten aus dem Multimedia Information Retrieval, Multimedia Informationssystemen und Semantik.
- **Sonstige Systeme oder Konzepte:** Objektorientierte Systementwicklung, Design Patterns, Dokumentation, Umgang mit Repositories, Dependency Management (z.B. Maven), grundlegende Konzepte der semantischen Modellierung, RDF, Semantic Web.
- **Methodik:** Anwendung von Forschungsmethodiken zum strukturierten Lösen von Problemstellungen und zum Erstellen von wissenschaftlich-technischen Dokumenten (z.B. Nunamaker [29]).

Darüber hinaus müssen auch die formalen Voraussetzungen seitens der Universität für die Anmeldung einer Abschlussarbeit erfüllt sein. Diese entnehmen Sie bitte den Zulassungsvoraussetzungen für Ihren Studiengang.

1. Themenbeschreibung

Smart Multimedia Information Retrieval (MMIR) führt eine Reihe von Konzepten ein, die über klassisches Information Retrieval hinaus gehen. Hierzu gehören erweiterte Metriken für die Berechnung von Ähnlichkeit oder Empfehlungen, Möglichkeiten der Merkmalsintegration und -fusion, sowie Erklärbarkeit von MMIR Prozessen und Inhalten. Unter Erklärbarkeit versteht man in diesem Zusammenhang menschen-verständliche Antworten auf Fragestellungen wie "was ist in dem Element zu sehen?", "welche Anfrage erzeugt das System aus meiner Frage?", "warum ist das Element in der Ergebnisliste?", "was ist die wichtigste Eigenschaft des Elements?", oder "warum ist das Element vor dem anderen?". Die Berechnung der Antworten auf derartige Fragen werden durch den Einsatz diverser Metriken auf Basis von MMIR Merkmalen möglich und in Form von Indexstrukturen repräsentiert. Im Rahmen von Smart MMIR sind dies sog. *Graph Codes* [41]. Ein *Graph Code* ist im Wesentlichen die Adjazenzmatrix eines Merkmalsgraphen. Eine Vielzahl von Berechnungen auf Basis von *Graph Codes* wurde in [40][42][43] beschrieben, die mathematisch bereits Antworten in Form von *Graph Codes* auf o.g. Fragen liefern können.

Der Bereich "Augmented oder Mixed Reality" stellt einen speziellen Nutzungskontext für MMIR Anwendungen dar. Hier werden Elemente der realen Welt durch Multimedia Elemente angereichert oder überlagert. Typischerweise werden für die Erkennung der anzureichernden Elemente sog. Anker definiert, die dann bspw. ein bestimmtes Multimedia-Objekt in der realen Welt erkennen und an das AR/MR-Framework weitergeben. Betrachtet ein Benutzer die reale Welt bspw. durch ein Smartphone oder mittels einer AR/MR-Brille, so können auf diesen Ankern zusätzliche Multimediale Inhalte angezeigt werden. Somit kann sowohl die Merkmalerkennung (im Anker) als auch das Finden des passenden Multimedia-Inhalts zur Darstellung auf / neben dem Anker als MMIR-Prozess betrachtet werden. Da AR/MR-Anwendungen typischerweise auf spezieller Hardware ausgeführt werden, ist die Performance der zugehörigen MMIR-Prozesse für bestimmte Anwendungsfälle von großer Wichtigkeit.

Im Rahmen dieser Aufgabenstellung wird untersucht, in wie weit sich Graph Codes MMIR Prozesse auf Android-Smartphones einsetzen lassen. Hierzu wird eine AR/MR-Anwendung entwickelt, die als Ankerpunkt bestimmte Bilder verwendet. Wird eines dieser Bilder erkannt, so wird mittels MMIR und Graph Codes automatisch eine Slideshow aus ähnlichen Bildern, die auf dem Smartphone gespeichert sind, berechnet und über dem Anker angezeigt. Für den Benutzer sieht es dann so aus, als würde sich das ausgedruckte und an der Wand aufgehängte Bild bewegen.

Rein technisch sind derartige Möglichkeiten bereits in den Standards für MR/AR-Anwendungen enthalten. Daher liegt der Fokus der Arbeit darauf, die zugehörigen MMIR-Prozesse zu modellieren und durch den Einsatz von Graph Codes deren Laufzeit zu untersuchen und ggfs. zu optimieren.

Sie beschäftigen sich also unter anderem mit den folgenden Fragestellungen:

- wie lassen sich AR/MR-Anwendungen unter Android realisieren?
- welche Möglichkeiten der Merkmalsextraktion von Fotos gibt es unter Android (Personen, Tiere, Farben, Objekte, Ortsmarken, usw.)?
- wie können die extrahierten Merkmale auf einem Android-Smartphone in Graph Codes umgewandelt werden?
- wie können Zugriffsmechanismen auf die Foto-Bibliothek mit Hilfe von Graph Codes gestaltet werden?
- wie lässt sich die Qualität der Ergebnisse messen und vergleichen?

Im Rahmen Ihrer Implementierung entwickeln Sie einen Android-Prototypen für o.g. Anwendungsfall.

2. Theoretischer Teil

Der theoretische Teil der Arbeit wird auf den Konzepten von Norman & Draper (User centered system design) [28] und Nunamaker [29] aufgebaut und methodisch hergeleitet.

Als Modellierung wird die UML [30] verwendet. Dies bedeutet, dass sich auch die Dokumentstruktur von Exposé und Abschlussarbeit auf die Nunamaker Methodik stützt und entsprechende Forschungsziele der Nunamaker-Typen (Beobachtung, Theoriebildung, Implementierung und Experiment) definiert werden. Im einem Exposé formulieren Sie zunächst Ihre Forschungsfragen und leiten die zugehörigen Forschungsziele ab. Nachdem dieses Exposé freigegeben wurde und somit auch das erste Kapitel Ihrer späteren Abschlussarbeit darstellt, führen Sie in den Stand der Wissenschaft und Technik ein und stellen die für das Thema relevanten Konzepte vor. Auf dieser Basis kann dann die Modellierung stattfinden. Weitergehende Informationen zum Thema "Aufbau von wissenschaftlich-technischen Dokumenten" finden Sie unter [35]. Das Kennwort, um die Inhalte ansehen zu können lautet **FUH-SW!**.

3. Praktischer Teil

Der praktische Teil der Arbeit enthält die programmiertechnische Umsetzung der Modellierung und die experimentelle Überprüfung. Als Programmiersprache kommt typischerweise Java zum Einsatz. Die Verwendung von Standard-Entwicklungstools wie Eclipse, GIT, Maven, Javadoc wird vorausgesetzt. Im Rahmen des praktischen Teils erfolgt auch die Erstellung von Test-Cases und deren Dokumentation, eine Evaluierung der Ergebnisse inklusive der Wahl und Begründung einer geeigneten Evaluierungs-Methodik, sowie ein Ausblick auf zukünftige Ergänzungen. Im Rahmen der Implementierung folgen Sie den GoF Design Patterns und orientieren Ihre Arbeit an den bestehenden Strukturen. Hierbei sind insbesondere das *Command Pattern*, das *Abstract Factory Pattern*, *Singleton Strukturen* und das *Model-View-Controller Prinzip* zu berücksichtigen. Den Einsatz und die Verwendung der Patterns stimmen Sie mit Ihrem Betreuer ab.

4. Betreuung

Die inhaltliche und fachliche Betreuung der Arbeit wird von Dr.-Ing. Stefan Wagenpfeil übernommen. Hierzu steht Ihnen jederzeit die Kommunikation per Skype, Telefon, E-Mail zur Verfügung, regelmäßige Abstimmungstermine mit dem Betreuer helfen Ihnen, die Arbeitsergebnisse strukturiert voranzutreiben und geben Ihnen die Gelegenheit, durch unmittelbares Feedback zügig und effizient die Ziele der Arbeit zu erreichen. Diverse "Quality-Gates" stellen sicher, dass die Abschlussarbeit auch die Anforderungen der Universität und des Lehrgebietes erfüllt und somit zu einem erfolgreichen Ergebnis führen wird.

5. Aufgaben im Vorfeld

Bevor Sie sich dafür entscheiden, das Thema zu bearbeiten, sollten Sie sich einen groben Überblick zu den benötigten Voraussetzungen oder Hintergründen im Rahmen einer Recherche verschaffen. Im Literaturverzeichnis dieses Dokuments sind eine Reihe von Quellen hierzu aufgeführt. Anschließend beginnen Sie mit der Erstellung eines Exposés, welches den genauen Rahmen Ihrer Arbeit absteckt. Die Gliederung des Exposés sieht hierbei wie folgt aus:

1. Einleitung (Motivation der Arbeit)

1.1. **Forschungsthema** (Beschreibung des Themas, Einordnung in ein übergeordnetes Themengebiet, Relevanz der Arbeit)

1.2. **Problembeschreibung** (Formale Beschreibung des Problems durch die Definition von Problemfeldern)

1.3. **Forschungsfragen** (Definition der Forschungsfragen zu den Problemfeldern)

1.4. **Methodik und Ableitung der Forschungsziele** (z.B. nach Nunamaker)

1.5. **Ansatz** (Beschreibung, wie die Forschungsziele gruppiert werden und wo sie im Rahmen der späteren Arbeit behandelt werden)

1.6. **Aufbau** (Festlegung der Struktur Ihrer Arbeit)

1.7. Zeitplan (inklusive Meilensteinplanung und Gantt-Diagramm auf Basis der Forschungsziele)

1.8. Zusammenfassung

Die Ergebnisse fassen Sie im Rahmen eines Dokumentes (Exposé) in ca. 10 bis 15 Seiten zusammen. Das Exposé ist die Basis für die spätere Abschlussarbeit und gibt einen für alle Beteiligten definierten Rahmen der Arbeit vor. Dadurch werden unter Anderem folgende Punkte sicher gestellt:

- Überprüfung der formalen und schriftlichen Fähigkeiten des Studenten zur Erstellung und Ausarbeitung einer Abschlussarbeit
- Vermittlung der Herangehensweise für die Strukturierung des wissenschaftlichen Diskurses
- Entwicklung eines Grundverständnisses für die zu erstellende Abschlussarbeit

Nach der Freigabe und Besprechung des Exposé mit Ihrem Betreuer und dem Lehrstuhlinhaber, beginnen Sie damit, die eigentliche Arbeit zu schreiben und die Planung, Konzeption und Implementierungen, die im Exposé bereits vorgeschlagen wurden, umzusetzen. Das Exposé wird direkt als erstes Kapitel Ihrer Arbeit verwendet.

6. Die Abschlussarbeit

Durch die Erstellung des Exposé haben Sie bereits einen wichtigen Teil der Abschlussarbeit fertig gestellt: die saubere wissenschaftliche Verankerung, die daraus resultierende Ableitung der Forschungsziele, sowie den Ansatz und die Struktur der Arbeit. Deshalb können Sie die im Rahmen des Exposé erstellten Kapitel direkt in Ihre Abschlussarbeit übernehmen (bis einschl. 1.6). Nun erfolgt:

- eine weitergehende, selbständige Literaturrecherche
- eine Beschreibung des aktuellen Stands von Wissenschaft und Technik
- das Konzeptionelle Design der Architektur für die Umsetzung und Implementierung
- die praktische Einarbeitung (Proof-Of-Concept Implementierungen, Kennenlernen der APIs und Frameworks)
- das Erstellen von Test-Szenarios, ggfs. auch die Organisation von Test-Daten
- die Implementierung
- eine Evaluierung des Ergebnisses mit Vergleich und Ausblick

Die jeweiligen Ergebnisse halten Sie in Ihrer Abschlussarbeit fest und folgen dabei analog zu den definierten Forschungszielen einer logischen Struktur:

1. Einleitung (folgt dem Exposé bis einschl. 1.6)

2. Stand der Wissenschaft und Technik (Für jedes Forschungsziel vom Typ Beobachtung wird ein Kapitel vorgesehen und der zugehörige Stand der Wissenschaft beschrieben. Am Ende werden "remaining challenges" definiert und zusammengefasst.)

3. Modellierung (folgt dem Aufbau von Kapitel 2 und sieht für jedes Forschungsziel vom Typ Theoriebildung ein Kapitel vor)

4. Implementierung (folgt dem Aufbau von Kapitel 2 und bearbeitet Forschungsziele vom Typ Implementierung)

5. Evaluierung (folgt dem Aufbau von Kapitel 2 und bearbeitet Forschungsziele vom Typ Experiment)

6. Diskussion (Wie kann das Gesamtergebnis eingeordnet werden? Welche "remaining challenges" verbleiben? Was kann Teil von "future work" sein? Zusammenfassung)

Die Arbeit ist so ausgelegt, dass der Anteil von Theorie und Praxis jeweils 50% beträgt. Durch die hier bereits vorgeschlagene Herangehensweise werden Sie zu einer effizienten, wissenschaftlich korrekten und nachvollziehbaren Arbeitsweise angeleitet. Im Folgenden finden Sie eine Reihe weitergehender Links und Referenzen, die Ihnen einen Einstieg

und Überblick über das Thema erleichtern. Sollten Sie Interesse an diesem Thema haben, kontaktieren Sie bitte einen der genannten Ansprechpartner.

References

1. Beyerer J. (2017), *Pattern Recognition - Introduction*; Publisher: Walter de Gruyter GmbH & Co KG, Berlin; pp. 10-37, ISBN: 978-3-110-53794-9
2. Beierle C. (2019), *Methoden wissensbasierter Systeme - Grundlagen*; Publisher: Springer-Verlag, Berlin Heidelberg New York; pp. 89-157, ISBN: 978-3-658-27084-1
3. Goodfellow I. (2016), *Deep Learning*; Publisher: MIT Press, Cambridge; ISBN: 978-0-262-03561-3
4. Heaton J. (2015), *Deep Learning and Neural Networks*; pp. 1-25 ISBN: 978-1505714340
5. Google.com (2020), *Google Knowledge Search API*. Available online: <http://developers.google.com/knowledge-graph> (accessed 23.08.2020)
6. W3C.org (2020), *W3C Semantic Web Activity*. Available online: <http://w3.org/2001/sw> (accessed 23.08.2020)
7. Avola D. (2018), *Low-Level Feature Detectors and Descriptors for Smart Image and Video Analysis: A Comparative Study*; Journal: Intelligent Systems Reference Library; , pp 7-29; ISBN: 978-3-319-73890-1
8. Domingue J. (2011), *Introduction to the Semantic Web Technologies*. Available online: <https://doi.org/10.1007/978-3-540-92913-0> (accessed 2020)
9. Kwasnicka (2018), *Bridging the Semantic Gap in Image and Video Analysis*; Publisher: Springer, Berlin; pp. 97-118, ISBN: 978-3-319-73891-8
10. Mc Kevitt P. (2003), *MultiModal semantic representation*; Journal: In Proc. of the SIGSEM Working Group on the Representation of MultiModal Semantic Information, Editor Harry Bunt Kiyong Lee Laurent Romary and Emiel Kraemer
11. Spyrou (2017), *Semantic Multimedia Analysis and Processing*; Publisher: CRC Press, Boca Raton; pp.31-63, ISBN: 978-1-351-83183-3
12. Scherer R., *Computer Vision Methods for Fast Image Classification and Retrieval*; Publisher: Polish Academy of Science, Warsaw; pp. 33-134, ISBN: 978-3-030-12194-5
13. Mark Nixon, *Feature Extraction and Image Processing for Computer Vision*; Publisher: Academic Press Elsevier
14. Bhute B. (2012), *Multimedia Indexing and Retrieval Techniques: A Review*; Journal: International Journal of Computer Applications, Volume 58, pp 35-42
15. Smeulders A. (2000), *Content-based image retrieval at the end of the early years*; Journal: IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22, pp 1349-1380
16. Gurski F. (2019), *On Characterizations for Subclasses of Directed Co-Graphs*. Available online: <http://arxiv.org/abs/1907.00801> (accessed 24.08.2020)
17. Needham M., *Graph Algorithms, Practical Examples in Apache Spark and Neo4j*; Publisher: O'Reilly; ISBN: 978-1-492-05781-9
18. Robinson I., *Graph Databases*; Publisher: O'Reilly; ISBN: 978-1-491-93089-2
19. Jiezhong Q. (2017), *Network Embedding as Matrix Factorization: Unifying DeepWalk*. Available online: <http://arxiv.org/abs/1710.02971> (accessed 24.09.2020)
20. Bai et al. (2019), *SimGNN: A Neural Network Approach to Fast Graph Similarity Computation*. Available online: <https://doi.org/10.1145/3289600.3290967> (accessed 24.11.2020)
21. W3C.org (2013), *SPARQL Query Language for RDF*. Available online: <https://www.w3.org/TR/sparql11-overview/> (accessed 23.08.2020)
22. Nkgau T. (2017), *Graph similarity algorithm evaluation*; Journal: 2017 Computing Conference, pp 272-278
23. Sluzek A. (2013), *Local Detection and Identification of Visual Data*
24. J. S. Sevak and A. D. Kapadia and J. B. Chavda and A. Shah and M. Rahevar (2017), *Survey on semantic image segmentation techniques*; pp 306-313
25. Arndt et al. (2008), *COMM: A Core Ontology for Multimedia Annotation*; Journal of Combinatorial Theory - JCT; , pp 403-421
26. Ni J et al. (2017), *Research on Semantic Annotation Based Image Fusion Algorithm*; Journal: 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), pp 945-948
27. Gayathri N. (2019), *An Efficient Video Indexing and Retrieval Algorithm using Ensemble Classifier*; Journal: 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), pp 250-258;
28. Draper N. (1986), *User Centered System Design - New Perspectives on Human-computer Interaction*; Publisher: Taylor & Francis, Justus-Liebig-Universität; ISBN: 978-0-898-59872-8
29. Nunamaker J. (1991), *Systems Development in InformationSystems Research*; Journal of Management Information Systems 1991, pp. 89-106
30. Fowler M. (2004), *UML Distilled - A Brief Guide to the Standard Object Modeling Language*; Publisher: Addison-Wesley Professional, Boston; ISBN: 978-0-321-19368-1
31. Planche B. (2019), *Computer Vision with TensorFlow 2*; Publisher: Packt Publishing; pp. 77-99, ISBN: 978-1-78883-064-5
32. Schmitt I. (2005), *WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries*; Journal: 11th International Multimedia Modelling Conference, pp 222-229

33. Jung H. (2020), *Automated conversion from natural language query to SPARQL query*. Available online: <https://doi.org/10.1007/s10844-019-00589-2> (accessed 24.10.2020)
34. Bornschlegel M. (2016), *IVIS4BigData: A Reference Model for Advanced Visual Interfaces Supporting Big Data Analysis in Virtual Research Environments*; Journal: BDA@AVI
35. Wagenpfeil S. (2022), *Erstellen von wissenschaftlich-technischen Dokumenten*; Online: <http://www.stefan-wagenpfeil.de/nunamaker/>, Kennwort: FUH-SW!
36. Wagenpfeil S. (2020), *Github Repository of GMAF and MMFVG*. Available online: <https://github.com/stefanwagenpfeil/GMAF/> (accessed 25.09.2020)
37. Wagenpfeil S. (2021), *Towards AI-bases Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*; SMAP 2020 Conference paper
38. Wagenpfeil S. (2021), *AI-Based Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*; <https://www.mdpi.com/2078-2489/12/1/43>
39. Wagenpfeil S. (2021), *Fast and Effective Retrieval for Large Multimedia Collections*; <https://www.mdpi.com/2504-2289/5/3/33>
40. Wagenpfeil S. (2021), *Towards Automated Semantic Explainability of Multimedia Feature Graphs*; <https://www.mdpi.com/2078-2489/12/12/502>
41. Wagenpfeil, S. and McKeivitt P. and Hemmje, M. (2021), *Fast and Effective Retrieval for Large Multimedia Collections*. Big Data Cogn. Comput. 2021, 5(3), 33; DOI: <https://doi.org/10.3390/bdcc5030033>
42. Wagenpfeil, S. and McKeivitt P. and Hemmje, M. (2021), *Towards Automated Semantic Explainability of Multimedia Feature Graphs*. Information 2021, 12(12), 502; DOI: <https://doi.org/10.3390/info12120502>
43. Wagenpfeil, S. and McKeivitt P. and Cheddad, A. and Hemmje, M. (2022), *Explainable Multimedia Feature Fusion for Medical Applications*. Information 2021, 12(12), 502; DOI: <https://doi.org/10.3390/info12120502>
44. Wagenpfeil, S. and Engel, F. and McKeivitt P. and Hemmje, M. (2021), *AI-Based Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*. Information 2021, 12(1), 43; DOI: <https://doi.org/10.3390/info12010043>
45. Wagenpfeil, S. and Engel, F. and McKeivitt P. and Hemmje, M. (2021), *Graph Codes-2D Projections of Multimedia Feature Graphs for Fast and Effective Retrieval*. Available online: <https://publications.waset.org/vol/180> (accessed 10.11.2022)