

Skizzenerkennung mit tiefen neuronalen Netzen

Hagen Tarner kontakt@hagen-tarner.de, Kristian Hildebrand khildebrand@beuth-hochschule.de

KURZFASSUNG

Zur visuellen Kommunikation behelfen sich Menschen seit prähistorischen Zeiten einfacher Strichzeichnungen. Wurden diese früher auf Höhlenwände und später auf Papier gezeichnet, verwendet der moderne Mensch ein Gerät mit Touchscreen. Die Skizze als Eingabemedium ist vielseitig einsetzbar. Immer dann, wenn geschriebene oder gesprochene Sprache keine Mittel zur Kommunikation sind, können zeichenbasierte Eingaben als Vereinfachung dienen. Mit der vorliegenden Arbeit haben wir ein System entwickelt, mit dem Computer die Eingaben des Menschen verstehen und interpretieren können. Basierend auf Ansätzen aktueller Forschungsarbeiten im Bereich des Deep Learnings erzielen wir damit Erkennungsraten von 80,22 % und übertreffen damit den Stand der Technik und die gemessene Leistung des Menschen (73,10 %).

ABSTRACT

Sketching enables users to depict the visual world and communicate when written or spoken language is not an option. In addition, mobile touchscreen devices support this communication ubiquitously and thereby provide users with an exceptional intuitive human-computer interface. In this work we present a system that has learned how to recognize and interpret human input sketches. Based on recent state-of-the-art machine learning research we achieve classification results of 80.22 % that outperform previous works and human recognition accuracy (73.10 %).

EINLEITUNG

Skizzenbasierte Eingabe bietet Menschen in vielen Situationen die Möglichkeit, auf einfache und verständliche Art und Weise zu kommunizieren. Mit dem allgegenwärtigen Einsatz von mobilen Endgeräten kann man diese Art der Kommunikation auch auf intuitive Interaktionen mit dem Computer übertragen und dem Benutzer damit eine ideale Interaktionsplattform im Alltag zur Verfügung stellen. Die Arten der Interaktion reichen dabei von skizzenbasierter Modellierung von 3D-Modellen [ZNA08], über die Suche in großen Bild- oder 3D-Modelldatenbanken [EHB+11] [ERB+12], der Manipulation von Bildern [ERH+11], bis hin zur Personalisierung von physisch realisierbaren Objekten durch 3D-Drucktechniken [HA13]. Diese komplexe Art der Mensch-Maschine-Kommunikation setzt ein immer weiterreichendes Verständnis der Eingaben voraus. Die Maschine muss also zunächst verstehen, was der Mensch zeichnet. Diese Art der Interaktion und Erkennung skizzenbasierter Eingabe ist Gegenstand dieser Arbeit.

Basierend auf Ansätzen aktueller Forschungsarbeiten haben wir ein System entwickelt, das skizzenbasierte Benutzereingaben aus 250 Objektkategorien klassifizieren kann. Die Erkennung basiert auf dem Einsatz von Convolutional Neural Networks (CNNs), die in den letzten Jahren sehr effektiv bei Bildklassifikationsproblemen eingesetzt wurden [KSH12]. Dabei nutzen wir einen Datensatz aus 20.000 Skizzen mit korrekten Kategoriebeschreibungen zum Training unseres Klassifikationsmodells, der im Zusammenhang mit der Arbeit von Eitz et al. [EHA12] entstanden ist.

Mit unserem trainierten Klassifikationsmodell erreichen wir Erkennungsraten von 80,22 % und übertreffen damit

den Stand der Technik. Dies gelingt insbesondere durch eine veränderte Vorverarbeitung der Bildinhalte und den Einsatz partieller Skizzen in den Trainingsdaten.

STAND DER TECHNIK

Unsere Arbeit basiert auf einer Reihe kürzlich veröffentlichter Arbeiten aus dem Bereich des Maschinellen Lernens. Die von Eitz et al. [EHA12] entwickelte Arbeit untersucht als erste, wie Menschen Objekte skizzieren und wie der Computer diese auf Basis von lokalen Bildmerkmalen und unter der Nutzung eines Klassifikators erkennen kann. Dabei wird eine Klassifikationsgenauigkeit von 56 % erreicht. In diesem Zusammenhang entsteht auch ein Datensatz aus 20.000 handgezeichneten Skizzen in 250 Kategorien, die den Folgearbeiten als Grundlage dient.

Yu et al. [YYL17] konstruieren ein neuronales Netz Sketch-a-Net speziell für die Skizzenerkennung. Um die Erkennungsrate noch weiter zu verbessern, werden außerdem zwei verschiedene Arten der Aufbereitung vor dem eigentlichen Training des Netzwerks verwendet. Die Autoren deformieren und entfernen einzelne Striche aus den Zeichnungen. So erreichen sie im Endeffekt eine Erkennungsrate, die die eines Menschen um etwa 4 % übertrifft.

Seddati et al. [SDM15] entwickeln 2015 mit DeepSketch ein weiteres tiefes neuronales Netz zur Skizzenerkennung. Die Erkennungsrate liegt hier bei 75,42 %.

2016 veröffentlichen sie eine Erweiterung von DeepSketch [SDM16] mit der es möglich ist, Skizzen bereits während der Entstehung zu erkennen. Hierzu untersuchen sie die Möglichkeit, Skizzen in Stufen verschiedener Vollständigkeit zu zerlegen, um so zeitlich basierte Sequenzinformationen zu generieren. Diese dienen danach als Eingabedaten für ein Netz. Mit 77,69 % wird so die bis dato höchste Erkennungsrate auf dem von Eitz zur Verfügung gestellten Datensatz erreicht.

CNNs ZUR SKIZZENERKENNUNG

Die Ausgangsbasis unserer Arbeit bildet die Netzwerkarchitektur von DeepSketch [SDM15] (siehe Tabelle 1). Auffällig bei dessen Architektur sind die geringe Filtergröße in Layer 1, der häufige Einsatz von Zero-Padding und die hohe Anzahl an Filtern und deren Größe in Layer 12. Die Filtergröße in Layer 1 begründen die Autoren damit, dass Skizzen im Gegensatz zu Fotos häufiger klare Kanten besitzen, die so einfacher zu detektieren seien. Der Kantenerkennung im ersten Layer kommt des Weiteren zugute, dass die genutzten binären Skizzen maximalen Kontrast haben.

Die Verwendung von Zero-Padding soll der Translations-

#	Typ	Filtergröße	-anzahl	Stride	Aktivierung
1	Convolutional	7x7	64	2	ReLU
2	Maxpool	3x3	-	2	-
3	Zero-Padding	2x2	-	-	-
4	Convolutional	5x5	128	2	ReLU
5	Maxpool	3x3	-	2	-
6	Zero-Padding	2x2	-	-	-
7	Convolutional	3x3	256	1	ReLU
8	Convolutional	3x3	512	1	ReLU
9	Maxpool	3x3	-	2	-
10	Convolutional	5x5 (3x3)	4096	1	ReLU
11	Dropout	-	-	-	-
12	Convolutional	1x1	250	1	ReLU
13	Flatten	-	-	-	-
	LSTM	-	512	-	tanh
14	Dense	-	250	-	Softmax

Tab. 1: Die Architektur von BeuthSketch entspricht der von DeepSketch [SDM15], lediglich Layer 13 und 14 wurden für die Implementierung in Keras ergänzt. BeuthSketch Mini verkleinert die Filter in Layer 10 auf 3x3 und BeuthSketch LSTM fügt zwischen Layer 13 und 14 einen weiteren LSTM Layer ein.

invarianz dienen. So werden Merkmale, die sich außerhalb der Mitte befinden, auch nach mehrmaligem Falten und Downsampling durch die Convolutional und Max Pooling Layer noch erfolgreich erkannt. Die erhöhte Anzahl an Filtern und deren Größe (5×5) in Layer 12 sorgt dafür, dass das Netzwerk in der Lage ist, die erkannten Merkmale gut in eine räumliche Beziehung zueinander zu setzen. Anders als bei Fotos, wo Informationen wie Texturen oder Farbe weiteren Kontext liefern, fehlt dieser bei den Skizzen. Daher ist die räumliche Bezugnahme von hoher Bedeutung. Um verbesserte Klassifikationsgenauigkeiten zu erreichen, haben wir einige Veränderungen an der Architektur, den Vorverarbeitungsschritten und der Aufarbeitung der Trainingsdaten durchgeführt, die im Folgenden erläutert und in Tabelle 1 deutlich werden.

VORVERARBEITUNG

Die Aufbereitung bzw. Vorverarbeitung des Datensatzes hat einen wesentlichen Einfluss auf die Erkennungsrate. Die Vorverarbeitung haben wir durch folgende drei Schritte optimiert, die jede Skizze im Training durchläuft (siehe Abbildung 1):

- > Verkleinern des Sketches auf die vom Netzwerk erwartete Eingangsgröße. Diese unterscheidet sich je nach verwendetem Netz und umfasst entweder 224×224 Pixel (BeuthSketch) oder 168×168 Pixel (BeuthSketch Mini und BeuthSketch LSTM).
- > Reduktion von Grauwertschizze auf Binärschizze (Schwellwert: Farbwert > 0). Danach erfolgt eine Invertierung.

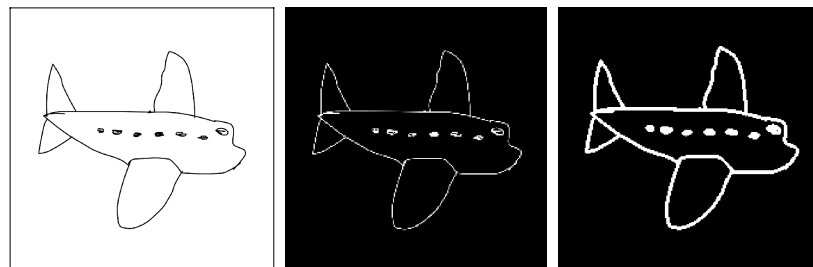


Abb. 1: Die drei Schritte der Vorverarbeitung: Verkleinern (links), Invertieren (Mitte) und Dilatation (rechts)

- > Dilatation mit einem (2, 2) -Kernel, damit nach der Verkleinerung in Schritt 1 keine Details verlorengehen. Ein beispielhaftes Ergebnis der Vorverarbeitung ist in Abbildung 1 zu sehen.

ENTFERNEN EINZELNER STRICHE ZUR SEQUENZGENERIERUNG

Der ursprüngliche Datensatz von Eitz et al. [EHA12] enthält 80 Skizzen pro Kategorie. Um die Variationen der Trainingsdaten für das Netz zu erhöhen, haben wir an verschiedenen Möglichkeiten gearbeitet, den bestehenden Datensatz zu vergrößern, ohne Skizzen hinzuzufügen. Eine Möglichkeit bestand darin, die zeitliche Abfolge, in der die Skizzen gezeichnet wurden, als Einzelskizzen zu übernehmen. Technisch ist das möglich, da [EHA12] ihren Datensatz u. a. auch als Scalable Vector Graphics (SVG)¹ zur Verfügung stellen. Dieses Textformat beinhaltet pro Sketch eine Reihe einzelner, der Reihenfolge nach sortierter Pfadelemente, die sich anteilig entfernen lassen. So sind pro Sketch vier neue Sketches mit jeweils unterschiedlichen Vollständigkeitsstufen des Originals entstanden (siehe Abb. 2).

¹ http://cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/sketches_svg.zip, zuletzt abgerufen am 13.03.2017.

ANSÄTZE ZUR SKIZZENERKENNUNG

Mit der veränderten Vorverarbeitung und der Möglichkeit, einzelne Striche eines Sketches zu entfernen, ergeben sich drei verschiedene Ansätze für das Training der Netzwerkarchitektur:

- > Den Datensatz vorverarbeiten und das Netzwerk lediglich mit vollständigen Sketches trainieren. Insgesamt stehen hierbei 20.000 Dateien zur Verfügung (BeuthSketch).
- > Die Bilder auf 168×168 Pixel verkleinern, einzelne Striche entfernen und mit den so entstandenen 80.000 Datenpunkten das Netzwerk trainieren (BeuthSketch Mini).
- > Aus den Skizzen Sequenzen anfertigen und diese einer LSTM als Eingabedaten zuführen (BeuthSketch LSTM).

Das Ergebnis dieser drei Ansätze ist in Abbildung 3 zu sehen. Die Abbildung zeigt die Top1-Genauigkeit der Kandidaten. Zum direkten Vergleich sind die Erkennungsraten eines Menschen (Quelle: [EHA12]), die der von [EHA12] vorgestellten SVM und die von DeepSketch [SDM15] und Sketch-a-Net [YYL17] angegebenen hinzugefügt. Da sich Letztere um weniger als ein Prozentpunkt unterscheiden, wurden diese aus Gründen der Übersicht zusammengefasst.

Abb. 2: Aufteilung eines Sketches in eine Sequenz aus vier Vollständigkeitsstufen: (A) 25 %, (B) 50 %, (C) 75 %, (D) 100 %

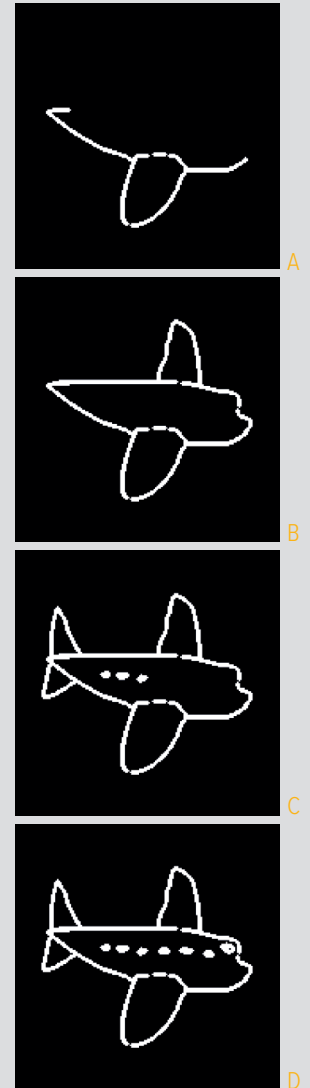
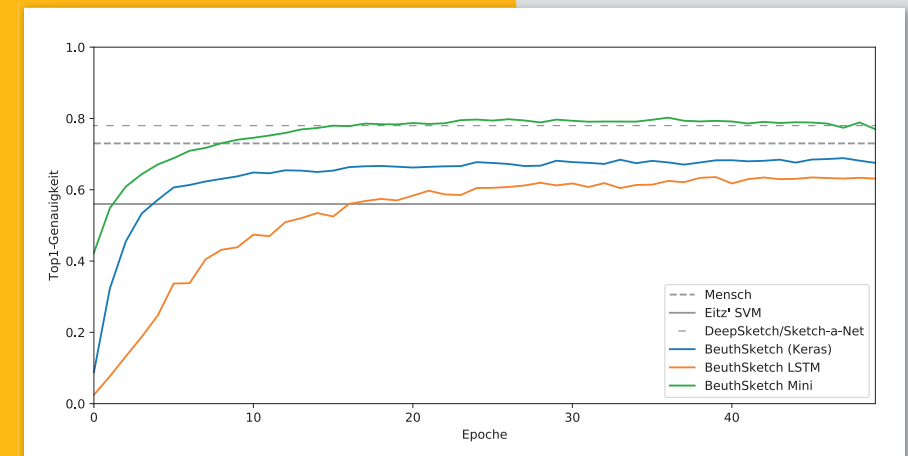


Abb. 3: Ergebnis der Evaluation: BeuthSketch erreicht maximal 68,87 %, BeuthSketch Mini 80,22 % und BeuthSketch LSTM 63,56 % Top1-Genauigkeit.



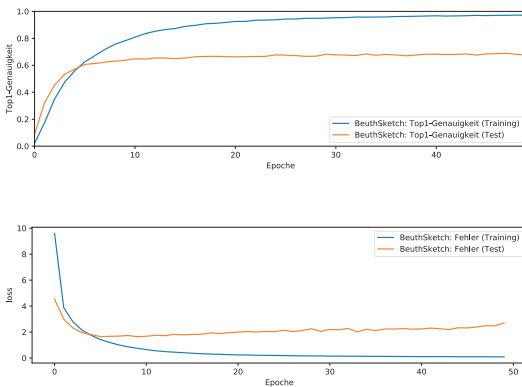
ERGEBNISSE UND AUSWERTUNG

Die drei vorgestellten Ansätze wurden alle mithilfe des Frameworks Keras [C17] implementiert und ausgewertet. Dabei ergab sich aus der Hyperparametersuche folgende optimale Konfiguration: Eine batch-size von 64, 4096 Filter in Layer 10 und Adadelata als Optimierungsfunktion. Sofern nicht anders angegeben, wurde für 50 Epochen mit einem dreigeteilten Datensatz (Trainingsdatensatz doppelt so groß wie Testdatensatz) trainiert.

BeuthSketch

Der erste Ansatz (im Folgenden BeuthSketch genannt) war, erst die Originalbilder vorzuverarbeiten und dem Netz zum Training zu zeigen. Hierfür mussten die Sketche auf 224×224 Pixel (von ursprünglich 1111×1111 Pixel) verkleinert werden. Die Anzahl der Skizzen wurde durch die Verwendung randomisierter Transformationen vervierfacht, sodass die Epochenlänge für dieses Training auf 52.800 für den Trainingsdatensatz und 27.200 für den Testdatensatz angestiegen ist. Das Ergebnis sind 68,87 % Top1-Genauigkeit in der Klassifizierung des Testdatensatzes. Damit liegt BeuthSketch knapp 4 % unter der Leistung eines Menschen und knapp 9 % unter der von [SDM15] und [YYL17].

Bei der Betrachtung der Ergebnisse fällt auf, dass die Fehlerfunktion für die Testdaten ab Epoche 15 leicht steigt, was ein Indikator für Overfitting ist. Um dem



vorzubeugen, haben wir versucht, den Datensatz komplexer zu machen, indem wir partielle Skizzen hinzugefügt haben (siehe BeuthSketch Mini).

BeuthSketch Mini

Für den zweiten Ansatz (im Folgenden BeuthSketch Mini genannt) wurden aus jedem Sketch vier Vollständigkeitsstufen generiert, sodass bei diesem Training ebenfalls Epochenlängen von 52.800 bzw. 27.200 Samples entstehen. Hierbei wird der Datensatz um ein Vierfaches vergrößert. Aus praktischen Gesichtspunkten haben wir dafür die Sketche zusätzlich auf 168×168 Pixel verkleinert.

Damit das Netzwerk die verkleinerten Sketches entgegennehmen konnte, musste die Architektur verändert werden. Dazu ist die Filtergröße des Convolutional Layers an Stelle 10 (siehe Tab. 1) von 5×5 auf 3×3 verringert worden.

Das so trainierte Netzwerk erreicht mit 80,22 % Top1-Genauigkeit die höchste Erkennungsrate der hier gezeigten Ansätze. Diese übertrifft die der von [EHA12] für einen Menschen angegebenen 73,10 % und die Klassifikationsgenauigkeiten von [SDM15] und [YYL17].

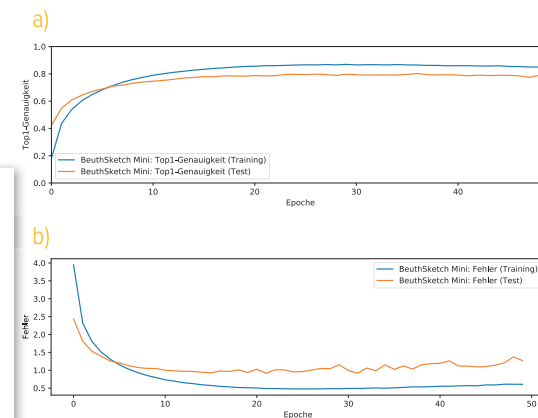


Abb. 4 und 5: BeuthSketch Erkennungsraten (a) und loss (b) für einen Durchlauf mit 50 Epochen.

BeuthSketch LSTM

Für den dritten Ansatz (im Folgenden BeuthSketch LSTM genannt) wurde versucht, die Merkmalsbestimmung eines CNN mit den zeitlichen Zusammenhängen, die man in einem Recurrent Neural Network (RNN) abbilden kann, zu vereinen. In der Fachliteratur ist dieser Ansatz als „Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Network“ (CLDNN) bekannt [SVS+15]. Um aus den eigentlichen Skizzen Zeitinformationen zu generieren, wurde wieder auf das Löschen einzelner Striche zurückgegriffen. Dies wurde dahingehend erweitert, dass jede Vollständigkeitsstufe einen klar definierten Sequenzindex bekam und so einem konkreten Zeitpunkt² zugeordnet werden konnte. So sind insgesamt 20.000 Sequenzen (pro Bild im Originaldatensatz eine Sequenz) mit jeweils vier Bildern entstanden. Für das Training standen so 13.200 Sequenzen (=52.800 Einzelbilder) und für den Test weitere 6.800 Sequenzen (=27.200 Einzelbilder) zur Verfügung.

Damit das Netzwerk die neue Datenstruktur verarbeiten kann, musste dessen Eingabevolumen um eine Dimension erweitert werden. Diese neue Dimension enthält den Sequenzindex jedes Sketches. Für diese Extrainformation stellt Keras den Layer Typen TimeDistributed zur Verfügung. Alle Layer, bis hin zum LSTM Layer, werden in einen Time Distributed Layer eingebettet. Der LSTM Layer selber arbeitet mit 512 internen Neuronen und wird über eine Tangens Hyperbolicus Funktion aktiviert. Der anschließende Dense Layer deckt sich mit denen in den anderen Netzen und muss nicht in einen Time Distributed Layer eingebettet werden, da der vorangehende LSTM Layer die Sequenzinformationen nicht im Output beibehält. Die Klassifizierung findet

weiterhin im letzten Layer mit einer Softmax Funktion statt. Die für diesen Versuch verwendete Architektur ist Tabelle 1 zu entnehmen.

BeuthSketch LSTM erreicht mit 63,56 % Top1-Genauigkeit den schlechtesten Wert im Test und liegt damit zwischen der SVM (56 %) [EHA12] und dem Ergebnis von BeuthSketch.

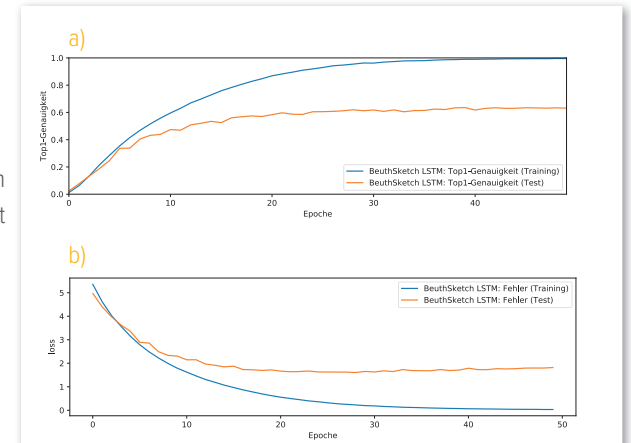


Abb. 6: BeuthSketch Mini Erkennungsraten (a) und loss (b) für einen Durchlauf mit 50 Epochen.

ZUSAMMENFASSUNG

Abbildung 7 illustriert einige Beispielergebnisse und die ersten fünf wahrscheinlichsten Ergebniskategorien. Anhand dieser Beispiele wird ersichtlich, wie präzise das trainierte BeuthSketch Mini Klassifikationsmodell die handgezeichneten Benutzerskizzen erkennen kann. <<

Abb. 7: Eingabe in das Netzwerk (obere Zeile) und die fünf vom Netzwerk vorhergesagten Kategorien, sortiert nach Konfidenz.

crown	99.99%	hat	95.52%	knife	71.29%	pizza	99.58%	wheel	98.65%
ashtray	0.01%	speed-boat	4.65%	carrot	19.86%	wheel	0.29%	tire	1.34%
castle	0.00%	cake	0.88%	sword	10.53%	ashtray	0.06%	pizza	0.01%
cat	0.00%	sailboat	0.84%	feather	4.66%	crown	0.03%	pretzel	0.00%
radio	0.00%	ship	0.67%	pen	4.19%	sailboat	0.02%	sailboat	0.00%

² Mit Zeitpunkt ist hier kein konkretes Datum, sondern der Zeitpunkt der Erstellung relativ zu dem der anderen Vollständigkeitsstufen gemeint. Beispiel: Die 25 % Vollständigkeitsstufe wird mit Index 1 versehen und ist damit zeitlich klar vor der 75 % Vollständigkeitsstufe (Index 3) einzuordnen.

LITERATUR

- [ZNA08] J. Zimmermann, A. Nealen und M. Alexa, „Sketch-based Interfaces: Sketching contours,“ *Comput. Graph.*, Bd. 32, pp. 486-499, 10 2008.
- [EHB+11] M. Eitz, K. Hildebrand, T. Boubekeur und M. Alexa, „Sketch-based image retrieval: Benchmark and bag-of-features descriptors,“ *IEEE transactions on visualization and computer graphics*, Bd. 17, pp. 1624-1636, 2011.
- [ERB+12] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand und M. Alexa, „Sketch-Based Shape Retrieval,“ *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, Bd. 31, pp. 31:1--31:10, 2012.
- [ERH+11] M. Eitz, R. Richter, K. Hildebrand, T. Boubekeur und M. Alexa, „Photosketcher: interactive sketch-based image synthesis,“ *IEEE Computer Graphics and Applications*, Bd. 31, pp. 56-66, 2011.
- [HA13] K. Hildebrand und M. Alexa, „Sketch-Based Pipeline for Mass Customization,“ *Rethinking Prototyping: Proceedings of the Design Modeling Symposium Berlin*, pp. 465-477, 2013.
- [KSH12] A. Krizhevsky, I. Sutskever und G. Hinton, „Imagenet classification with deep convolutional neural networks,“ in *Advances in neural information processing systems*, 2012.
- [EHA12] M. Eitz, J. Hays und M. Alexa, „How Do Humans Sketch Objects?,“ *ACM Trans. Graph. (Proc. SIGGRAPH)*, Bd. 31, pp. 44:1--44:10, 2012.
- [YYL17] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang und T. M. Hospedales, „Sketch-a-net: A deep neural network that beats humans,“ *International Journal of Computer Vision*, Bd. 122, pp. 411-425, 2017.
- [SDM15] O. Seddati, S. Dupont und S. Mahmoudi, „DeepSketch: Deep convolutional neural networks for sketch recognition and similarity search,“ in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2015.

- [SDM16] O. Seddati, S. Dupont und S. Mahmoudi, „DeepSketch 2: Deep convolutional neural networks for partial sketch recognition,“ in *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016.
- [C17] F. Chollet, *Keras: Deep Learning library for Theano and TensorFlow*. 2017.
- [SVS+15] T. N. Sainath, O. Vinyals, A. Senior und H. Sak, „Convolutional, long short-term memory, fully connected deep neural networks,“ in *Acoustics, Speech and Signal Processing (ICASSP)*, 2015 IEEE International Conference on, 2015.

KONTAKT

Prof. Dr. Kristian Hildebrand
Beuth Hochschule für Technik Berlin
Fachbereich VI / Informatik und Medien
Luxemburger Straße 10, 13353 Berlin
E-Mail: khildebrand@beuth-hochschule.de
Internet: <http://hildebrand.beuth-hochschule.de>

Multi-Channel-Anwendungen: MPEG-DASH-Standard und Transportstreams

Prof. Dr. phil. Jürgen Lohr

KURZFASSUNG

Für die Kommunikations- und Medientechnologien wurde ein neuer Standard entwickelt, die Dash-Technologie, welche viele Endgeräte an den unterschiedlichsten Netzwerken unterstützt. Die Beuth Hochschule forscht an zukünftigen interaktiven Medienbeiträgen und wie diese produziert, übertragen und dargestellt werden können. Kennzeichnend hierbei ist, dass die untersuchten interaktiven Medienbeiträge im Hypermedial-Verfahren verlinkt und im Multi-Channel-Verfahren mit vielen Audio- und Video-Spuren angereichert sind.

ABSTRACT

For communications and media technologies, a new standard, the Dash technology, has been developed. This standard supports many terminal devices in different networks. The Beuth University of Applied Sciences does research on future interactive media contributions and examines how they can be produced, webcasted and provided to the users. Characteristic of this is that the examined interactive media contributions are linked via hypermedia procedures and are enriched in the multi channel process with many audio and video tracks.

SCHLAGWÖRTER

Medienunternehmen, Kommunikations- und Medientechnologien, Endgeräte, Netzwerke, Dash-Technologie, Einsatzfelder, Beuth Hochschule, Forschung, Transportstreams, Multi-Channel-Verfahren, Audio- und Videospuren

EINLEITUNG

Im Rahmen des Forschungsprojekts „Medien-Systeme für Business-TV“ beschäftigt sich ein Team rund um die BeuthBox-Plattform mit der Entwicklung eines Systems zur Übertragung von mehrkanaligen Medienbeiträgen über öffentliche Netzwerke und unternehmensweites Intranet. Im Fokus des Projektes stehen Einsatzszenarien für Bildung, Unternehmenskommunikation und Unterhaltung. Ausgehend von den Aspekten des Forschungsprojekts wurde eine Erweiterung des Ansatzes zur technologischen Machbarkeit vorgenommen, sodass ein dauerhafter, qualitativ hochwertiger und prozess-optimierter Betrieb möglich wird.

Wichtige Prozesse bei der Bereitstellung von Mediensystemen sind Vorplanung, Aufzeichnung, Bereitstellung, Übertragung, Darstellung und Nachverfolgung von Medienlieferungen. Die Prozesse beschreiben die kreative Phase bis hin zur Nutzungsphase der Medienbeiträge in einem Portalsystem mit einer interaktiven Client-Applikation auf einem Endgerät. Dabei können Medienbeiträge, z. B. Vorlesungen, Schulungen, Präsentationen und weitere Anwendungsbereiche, im interaktiven Hypermedien-Verfahren oder Multi-Channel-Verfahren angeboten werden. In den Medienbeiträgen können dann beliebig viele virtuelle Elementarbeiträge kombiniert oder beliebig viele virtuelle Elementar-Videospuren- und Audiospuren synchron verarbeitet und übertragen werden.

Medienbeiträge, die im Multi-Channel-Verfahren übertragen und aufgezeichnet werden, bieten die Möglichkeit, individuellere Informationen bereitzustellen als es die bisherigen, verbreiteten Single-Channel-Verfahren können. Der Einsatz einer interaktiven Anwendung gestattet es dem Nutzer, die unterschiedlichen Medienbeiträge aus unterschiedlichen Kanälen zu selektieren.