



# Einführung in die Praktische Informatik

Prof. Björn Ommer      HCI, IWR  
Computer Vision Group





# Willkommen!

- An der Universität Heidelberg
- In der Informatik

# Das Team

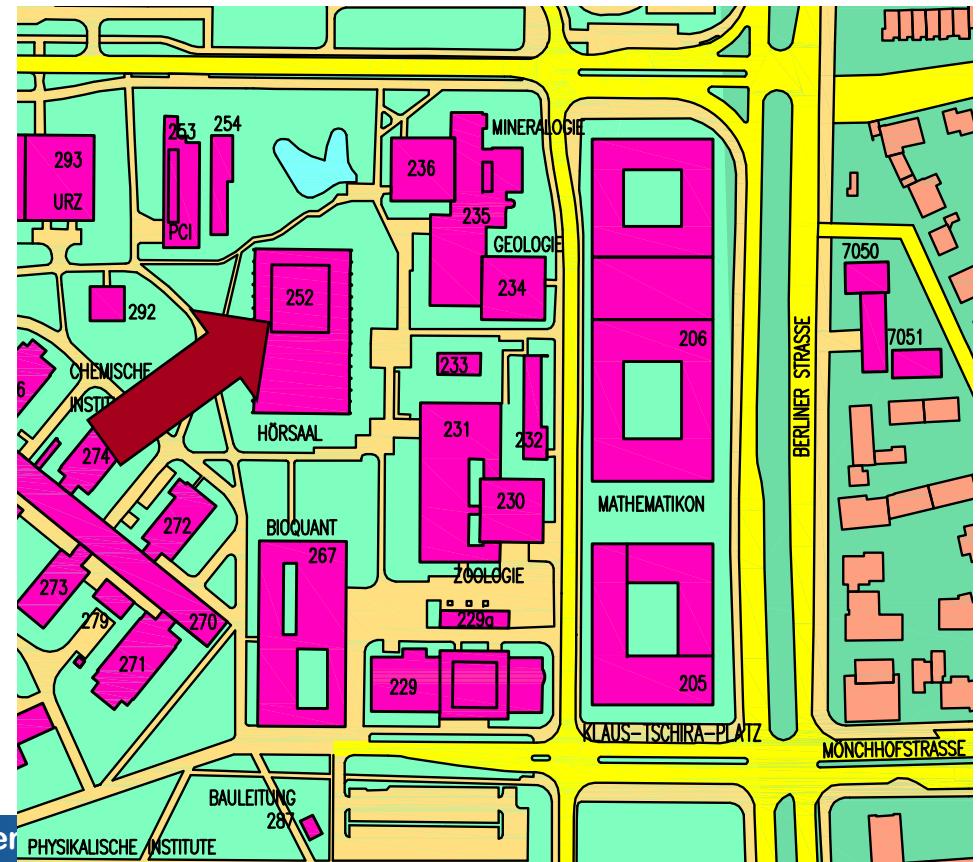
- Vorlesung: Prof. Dr. Björn Ommer
- Organisation: Uta Büchler & Timo Milbich
- TAs: Studierende  
höherer Semester

Vorname	Name
Patrick	Dammann
Julien	Stern
Thomas	Darr
Simeon	Scheib
Jacob	Schnell
Markus	Boehling
Rene	Snajder
Dominic	Gargya
Adrian	Danish
Christopher	Bednorz
Nils	Friess

- Webseite zur Vorlesung:  
[hci.iwr.uni-heidelberg.de/CompVis\\_Teaching](http://hci.iwr.uni-heidelberg.de/CompVis_Teaching)
- Skript aus Vorjahren: siehe Moodle

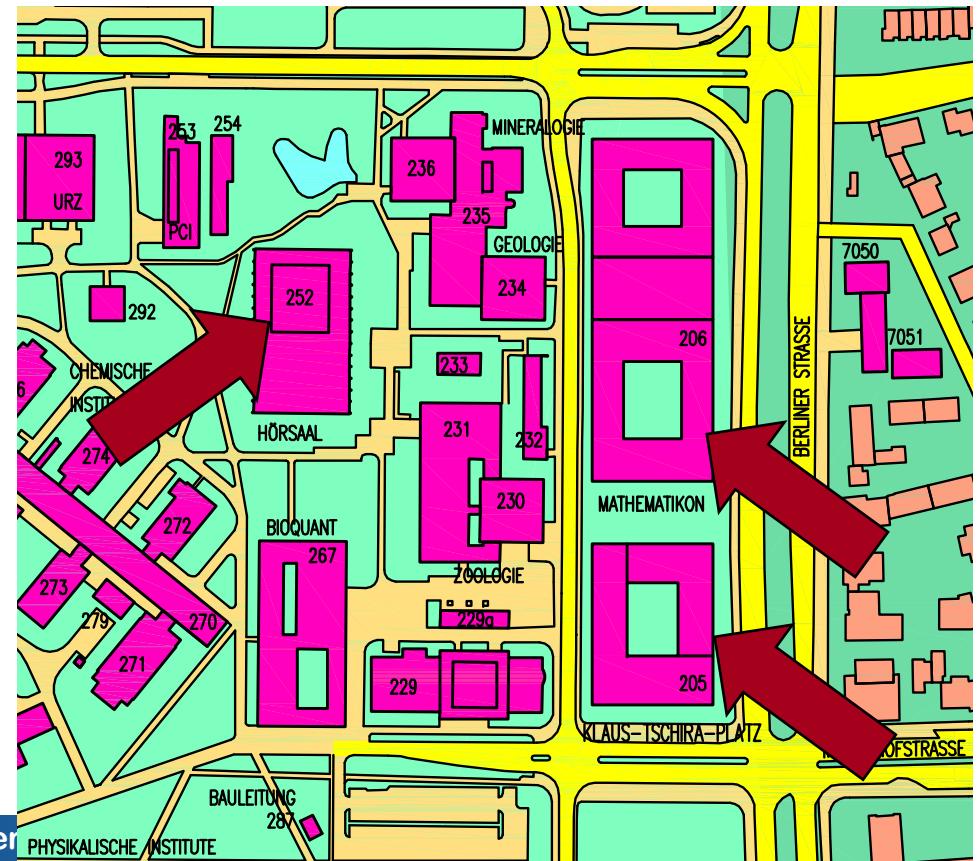
# Vorlesung

- Dienstags und Donnerstags, 14:15-15:45  
Großer Hörsaal Chemie, INF 252



# Übungen

- Anmeldung und Eintragen der zeitlichen Präferenzen in Moodle **bis Freitag (19.10.) 15 Uhr!**  
First come first served, ab 18.10. 17 Uhr!
- Zuteilung siehe Moodle
- ~20 Teilnehmer pro Übungsgruppe



# Prüfungsmodalitäten

- Die Modulnote ergibt sich aus dem Ergebnis einer zweistündigen Klausur.
- Es werden zwei Klausurtermine angeboten:
  - – 1. Klausur: Donnerstag, 07.02.2019 ~15-17 Uhr, Anmeldung im Moodle im Januar 2019, TBA
  - – 2. Klausur Dienstag, 09.04.2019 ~14-16 Uhr, Anmeldung im Moodle im März 2019, TBA
- Für den Schein ist das Bestehen einer der beiden Klausuren notwendig.
- Teilnehmer an der 1. Klausur haben die Möglichkeit, die 2. Klausur mitzuschreiben, sofern sie die 1. Klausur nicht bestanden haben
- Ein Mitschreiben der 2. Klausur zur Notenverbesserung - nach Bestehen der 1. Klausur - ist nicht möglich

# Zulassungsvoraussetzung

Zulassungsvoraussetzung zur Klausur ist eine erfolgreiche Teilnahme an den Übungen, d.h.

- mindestens 50% der Punkte in den Übungen
- regelmäßige Teilnahme an den Übungen  
(Feststellung liegt im Ermessen des Tutors)
- mindestens einmal Vorrechnen in den Übungen  
(Tutor kann Studenten zum Vorrechnen auffordern)
- Die Übungsaufgaben sollen in **Teams zu zweit** bearbeitet werden (Abgabe einer Lösung pro Team, an der Tafel vorrechnen und Lösung der Computerübungen erklären muss aber jeder alleine).

# Anerkennung von Klausurzulassungen aus dem Vorjahr

Entsprechend der allgemeinen Regel in der Informatik in Heidelberg (s. Webpage):

- Studierende, deren Prüfungsleistung in einer der Vorlesungen aus dem Pflichtmodulbereich nicht bestanden ist oder als nicht bestanden gilt, haben das Recht zur Prüfung (z.B. einer Klausur) **beim nächsten Angebot des Moduls** anzutreten, ohne vorher notwendigerweise am Übungsbetrieb teilzunehmen.
- Die Prüfungszulassung gilt also **max. 1 Jahr** (denn das ist der späteste nächste Prüfungstermin) und **nur, wenn man mindestens eine der Klausuren im Vorjahr mitgeschrieben** (und nicht bestanden) hat.
- Wenn Sie diese Regel in Anspruch nehmen möchten, kontaktieren Sie uns bitte im Voraus, um sicherzustellen, dass Sie auf der Liste der zugelassenen Studenten stehen.
- **Wir empfehlen trotzdem dringend die Teilnahme an unseren Übungen!**

# Moodle - E-Learning-Plattform

- <https://elearning2.uni-heidelberg.de/>
- Vorlesungsunterlagen
- Übungszettel
- Diskussions- und Tauschforen
- Einschreibeschlüssel für IPI: **IPIWS18\_compviz**

# Übungsgruppen

Montag	09:00 – 11:00	INF 205 / SR 9
Montag	09:00 – 11:00	INF 205 / SR B
Montag	09:00 – 11:00	Berliner Str. 43 / SR B128
Montag	11:00 – 13:00	INF 205 / SR 11
Montag	11:00 – 13:00	INF 205 / SR Statistik
Dienstag	09:00 – 11:00	INF 205 / SR 9
Dienstag	09:00 – 11:00	INF 205 / Sr 8
Dienstag	11:00 – 13:00	INF 205 / SR Statistik
Dienstag	16:00 – 18:00	INF 205 SR 10
Mittwoch	09:00 – 11:00	INF 205 / SR 9
Mittwoch	11:00 – 13:00	INF 205 / SR Statistik
Mittwoch	14:00 – 16:00	INF 205 / SR 10
Mittwoch	14:00 – 16:00	Berliner Str. 43 / SR B128
Mittwoch	14:00 – 16:00	INF 205 / SR 11
Donnerstag	09:00 – 11:00	INF 205 / SR 9
Donnerstag	11:00 – 13:00	INF 205 / SR 10
Donnerstag	16:00 – 18:00	INF 205 / SR 8
Donnerstag	16:00 – 18:00	INF 205 / SR 11
Freitag	09:00 – 11:00	INF 205 / SR 9
Freitag	11:00 – 13:00	INF 205 / SR 10
Freitag	14:00 – 16:00	INF 205 / SR 11
Freitag	16:00 – 18:00	INF 205 / SR 11

# Was können Sie machen, wenn die Übungsgruppe wechseln müssen/wollen?

- Selbst neu eintragen (falls noch Plätze frei sind)
- Wechselbörse (im Moodle) - bei Finden eines Partners mail an Uta Büchler & Timo Milbich
  
- In jedem Fall: Tutoren der alten und der neuen Gruppe kontaktieren!

# Ablauf der Übungen

- In der Regel dienen die Übungen zur Besprechung und Präsentation der Übungsaufgaben, die Sie vorher bearbeitet und abgegeben haben und die der Tutor korrigiert hat
- Möglichkeit Themen zu vertiefen & Fragen zu stellen
  
- Woche 1 (ab 22.10.) - Besprechung des Übungszettels 0
  - keine Abgabe und Korrektur, enthält wichtige Voraussetzungen
  - bitte versuchen Sie auf jeden Fall, eine Übung zu besuchen, auch wenn Sie noch nicht in der endgültigen Gruppe sind

# Abgabe der Übungsaufgaben

- Ausgabe neuer Übungszettel: Mittwochs
- Abgabe: Mittwoch 13h eine Woche nach Ausgabe, als Team zu zweit
- Besprechung/Korrektur: in der Übungsgruppe nach der Abgabe
- Abgabe der Übungszettel: per Moodle
  
- Übungszettel 0: Ausgabe heute (keine Abgabe erforderlich)
- ca. 11 Übungszettel insges.

# Bei Fragen zur Vorlesung / zu den Übungen

- sprechen Sie mich gerne direkt nach der Vorlesung an
- fragen Sie Ihren Tutor
- Forum auf Moodle ‚*generelle Fragen*‘ + separat pro Übung
- per Mail: Uta Büchler & Timo Milbich
- Vereinbarung von individuellen Sprechstundenterminen per Mail an [cvsec@iwr.uni-heidelberg.de](mailto:cvsec@iwr.uni-heidelberg.de)

# Praktisches Üben

- Programmieren ist ein wesentlicher (nicht alleiniger!) Inhalt der Vorlesung
- Beim Programmieren gilt: Übung macht den Meister! Programmieren ist eine Fertigkeit (Kunst). Eines der berühmtesten Bücher der Informatik von Donald E. Knuth heißt "[The Art of Computer Programming](#)"  
**Nutzen Sie alle gebotenen Möglichkeiten zum Üben!**
- In der Vorlesung/Übung benutzen wir eine UNIX-Programmierumgebung. Sie sollten Zugang zu so einem System haben um die Übungen durchführen zu können. Geeignet sind LINUX, Mac oder ein Windows-System mit WUBI  
<http://wiki.ubuntuusers.de/Wubi>
- Falls Sie Schwierigkeiten haben, melden Sie sich bei Ihrem Tutor

# Inhalte der Vorlesung

- Grundlegende Konzepte der Informatik kennenlernen  
z. B. Algorithmenbegriff, Komplexität, Abstraktion, . . .
- Algorithmisches Denken schulen  
Problem → Algorithmus → Programm
- Programmieren im Kleinen  
verschiedene Programmierstile (funktional, prozedural, objektorientiert, generisch), Erlernen der Programmiersprache C++  
Aber: Vorbereitung für Programmieren im Großen!
- Grundlegende Algorithmen und Datenstrukturen  
Suchen, sortieren, . . .  
Listen, Felder, Heaps, Stacks, Graphen, Bäume, . . .



# The Computer Vision Group Heidelberg

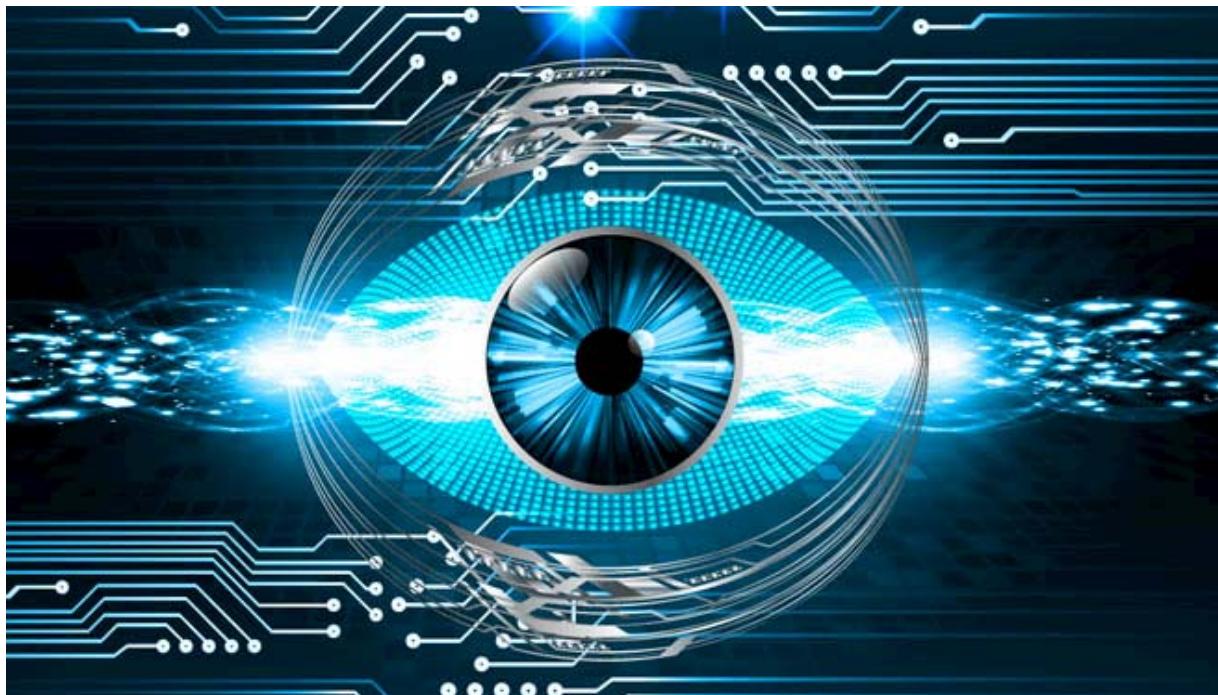


# Our Research: Computer Vision



Vision & Learning

Algorithms that teach machines to see



# Why Computer Vision



Cameras are  
all around us...



# Images & Videos are Omnipresent



Personal photo albums



Movies, news, sports



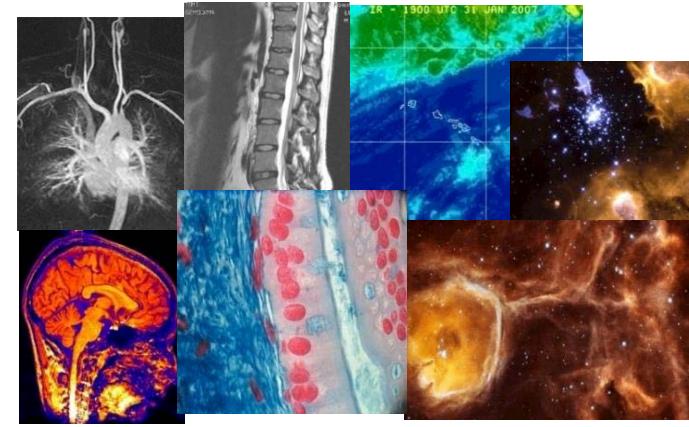
Internet services



Surveillance and security



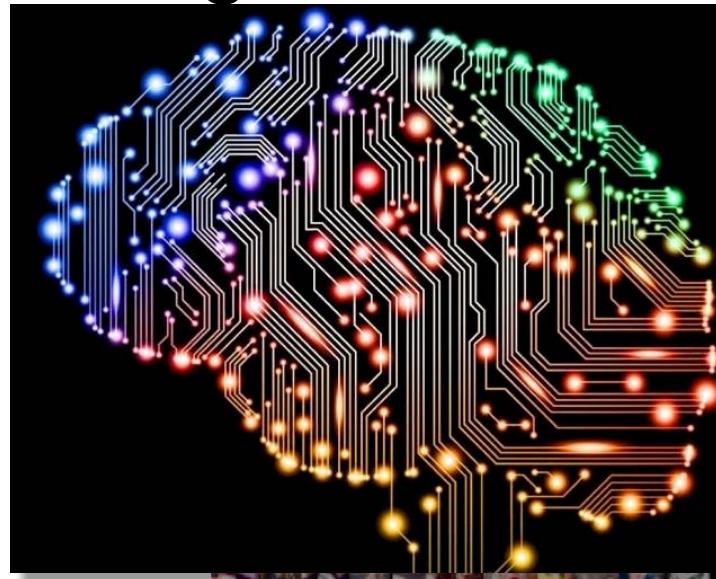
Mobile and consumer applications



Medical and scientific images

# The Grand Goal of Computer Vision & (Deep) Learning

by training an artificial intelligence



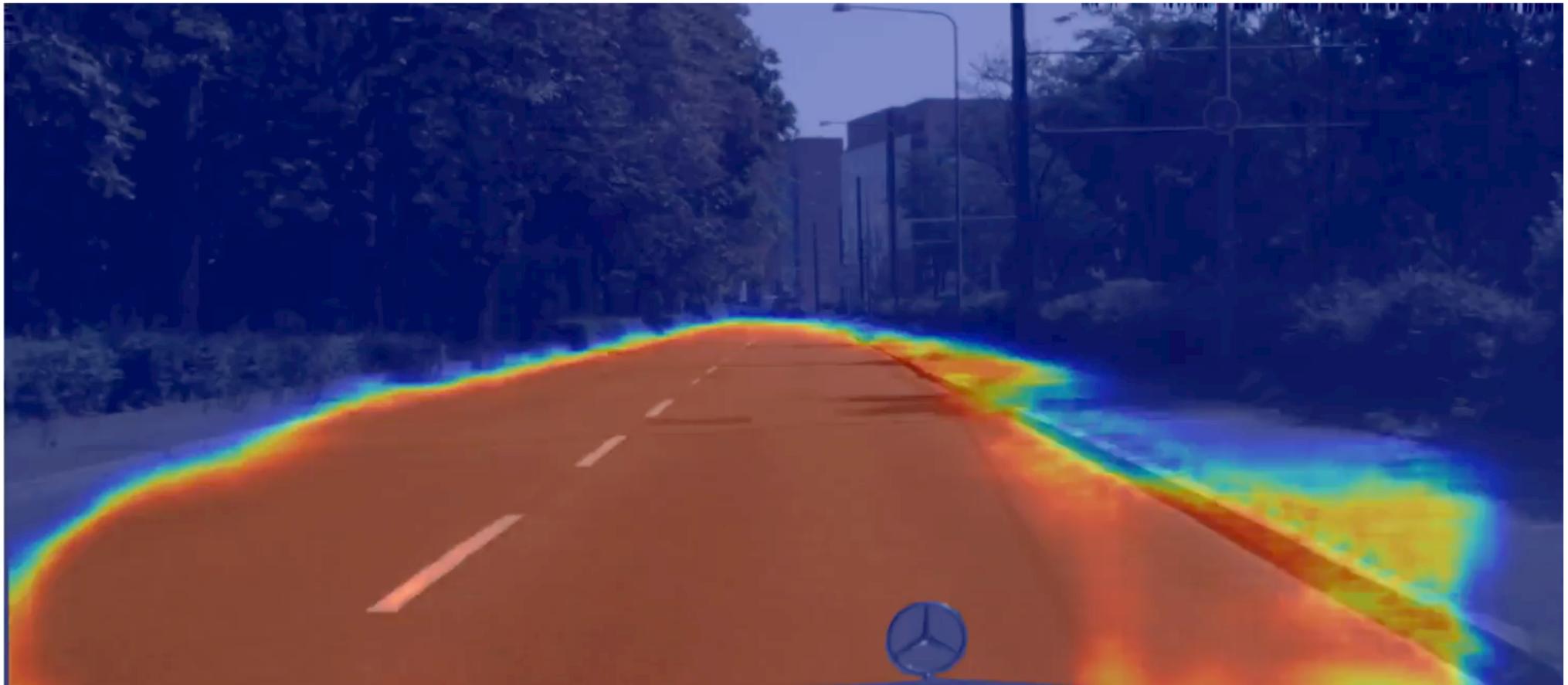
teaching machines to see

automatically using BIG data

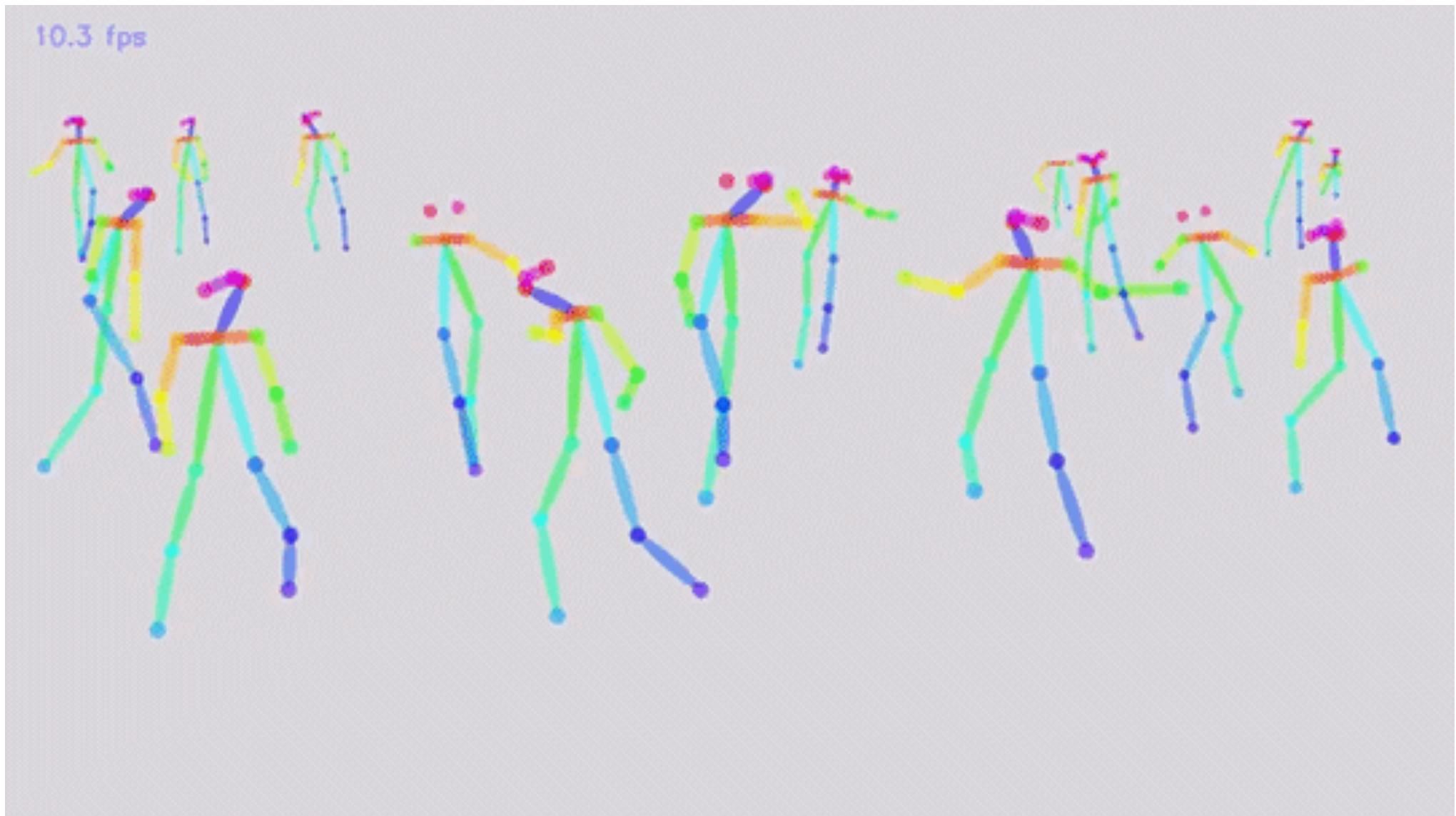
# Autonomous Driving



# Unsupervised Learning of Drivable Area



Bautista & Ommer,  
GCPR'17

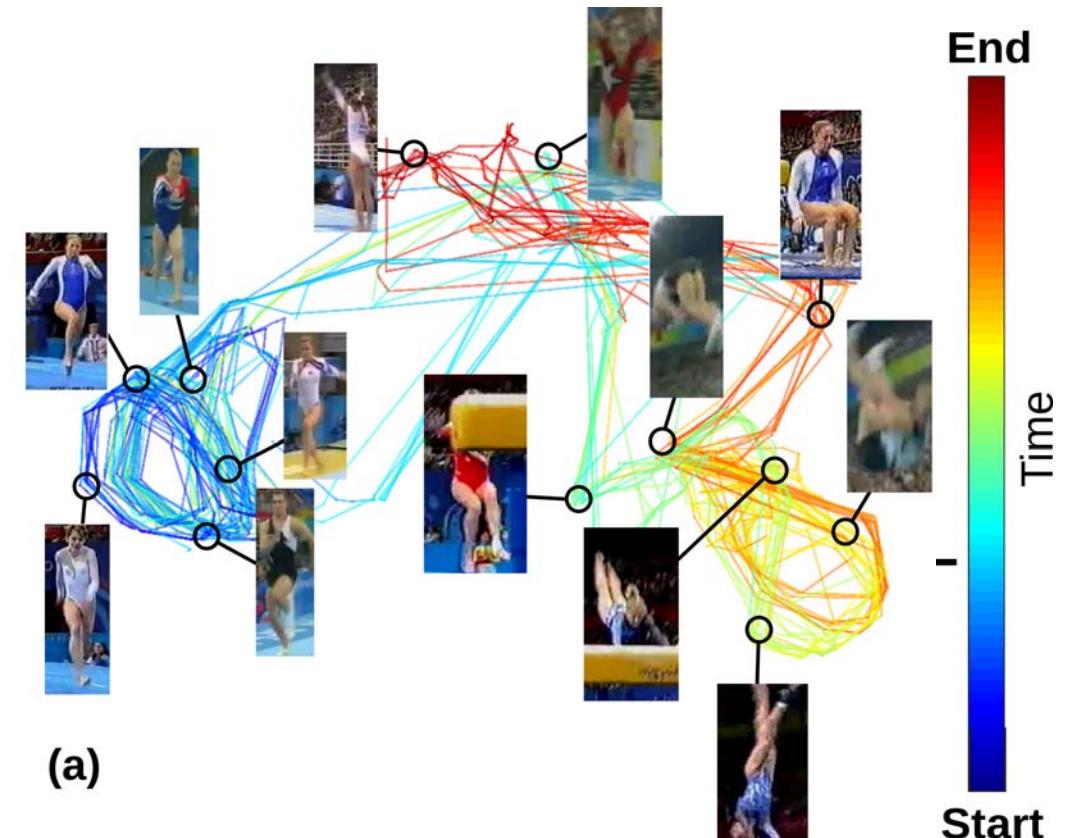


# Detecting, Analyzing, and Understanding Posture & Behavior



[Milbich et al., ICCV'17]

# Detecting, Analyzing, and Understanding Posture & Behavior

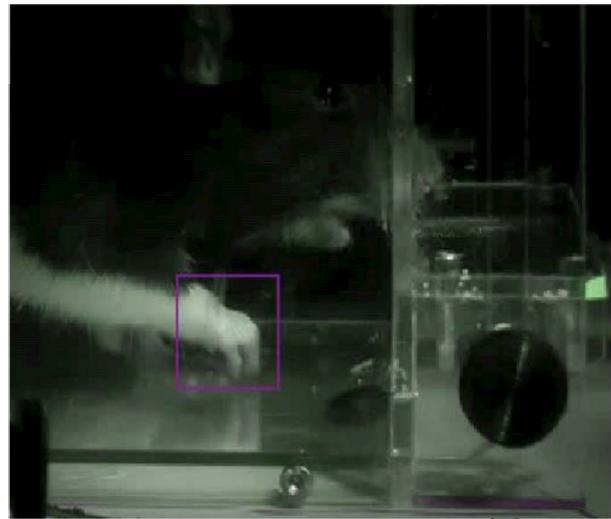


[Milbich et al., ICCV'17]

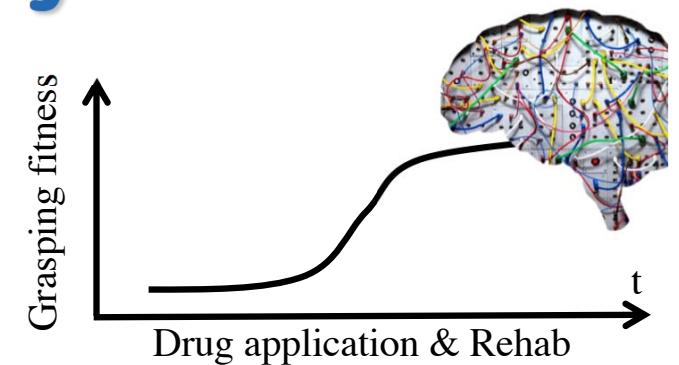
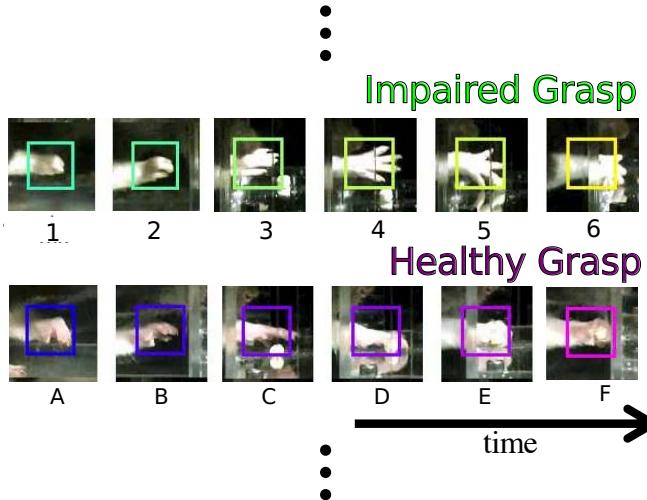
[https://hciweb.iwr.uni-heidelberg.de/compvis/research/tmilbich\\_iccv17](https://hciweb.iwr.uni-heidelberg.de/compvis/research/tmilbich_iccv17)

Björn Ommer | [ommer@uni-heidelberg.de](mailto:ommer@uni-heidelberg.de)

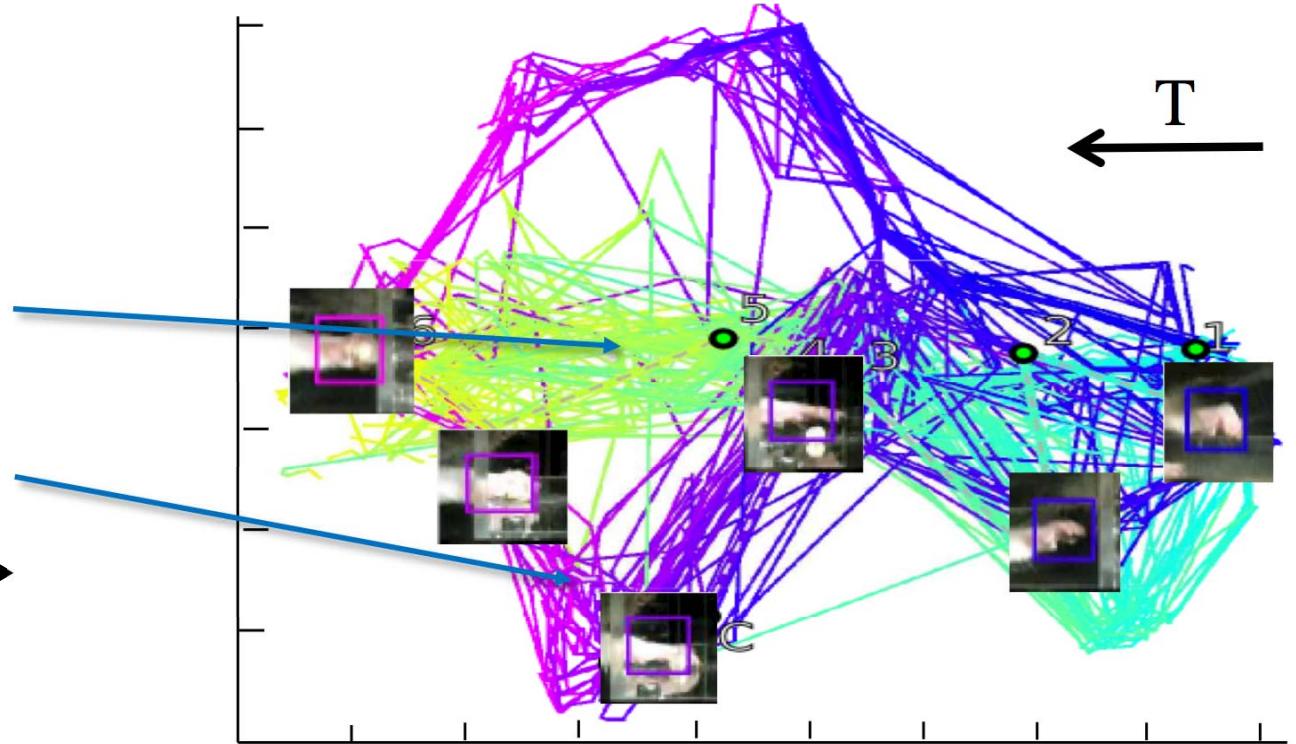
# Behavior Analysis $\Rightarrow$ Therapy



grasping sequences:



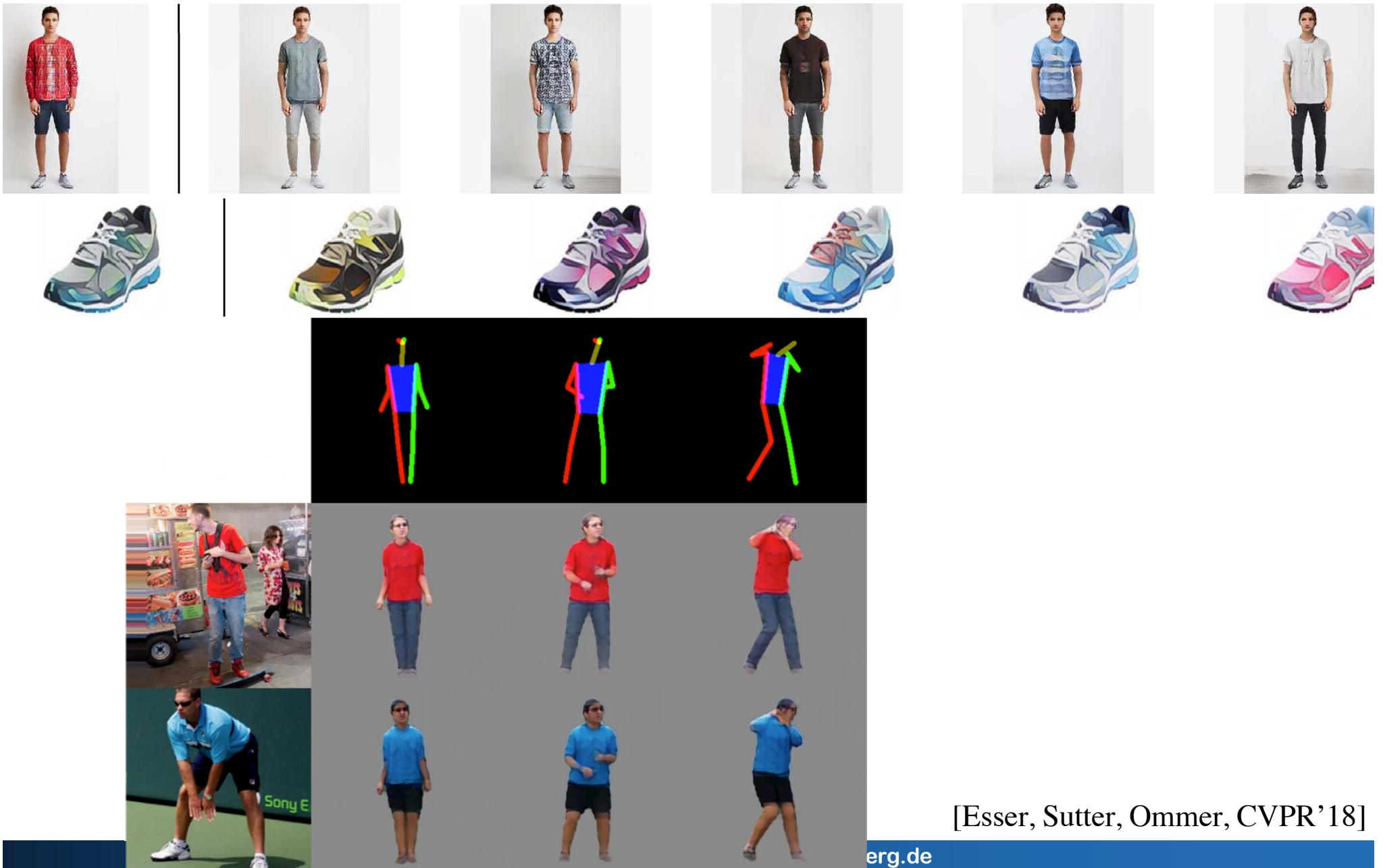
Representation of hand posture deformation



[Wahl et al., Nature Comm '17]

[Antic et al., MICCAI 2015]

# Image Generation



[Esser, Sutter, Ommer, CVPR'18]



# Generated Images



BERG

*source pose*



*synthesized*



*synthesized*



*synthesized*

<https://compvis.github.io/vunet/>

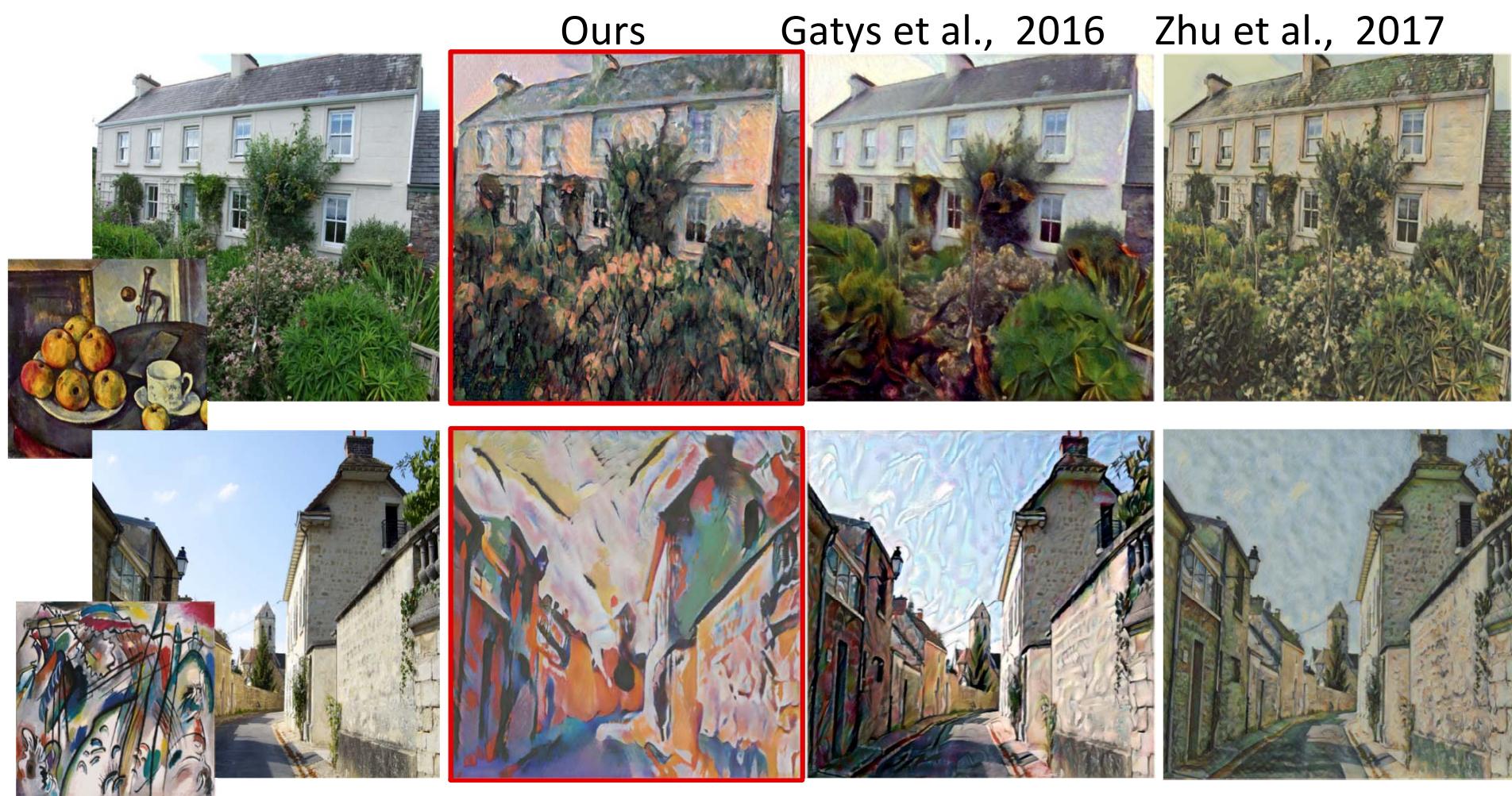
Source: [Gatys et al. 2015]

A



Input: photo

# Image Stylization





# HD Styliization



LBERG

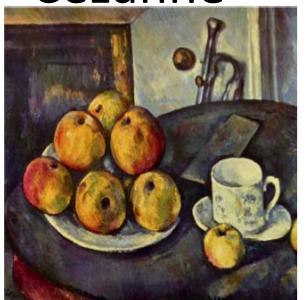
# HD Video Style Transfer

@ 1920x1280pix

Content



Cézanne



Kandinsky

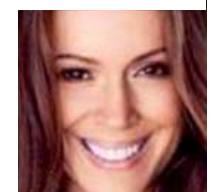
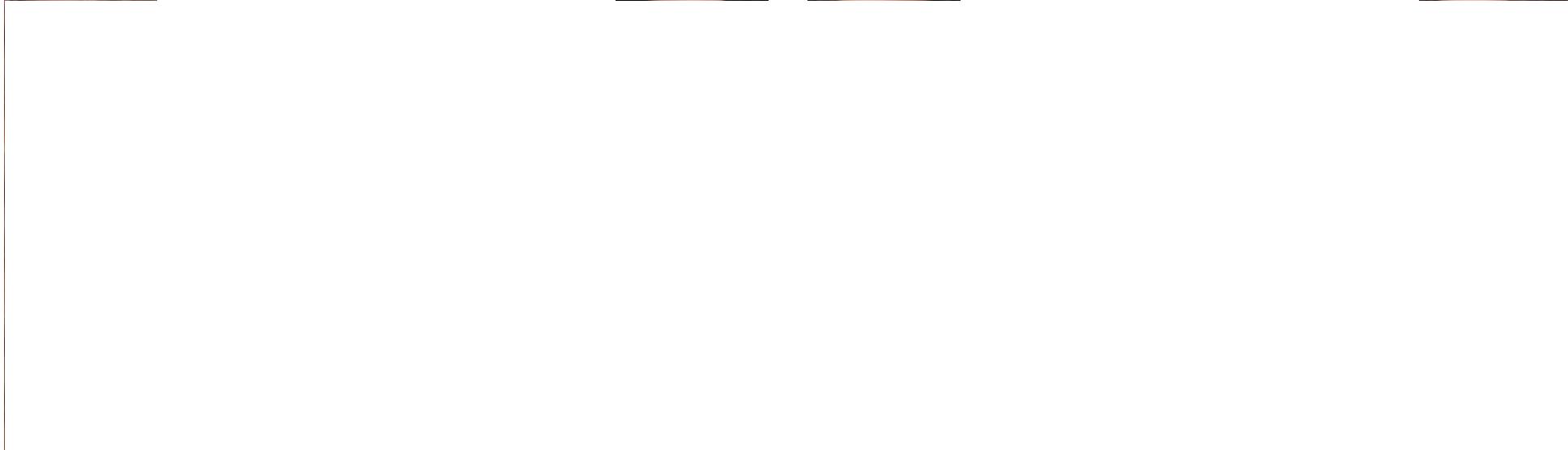


Picasso



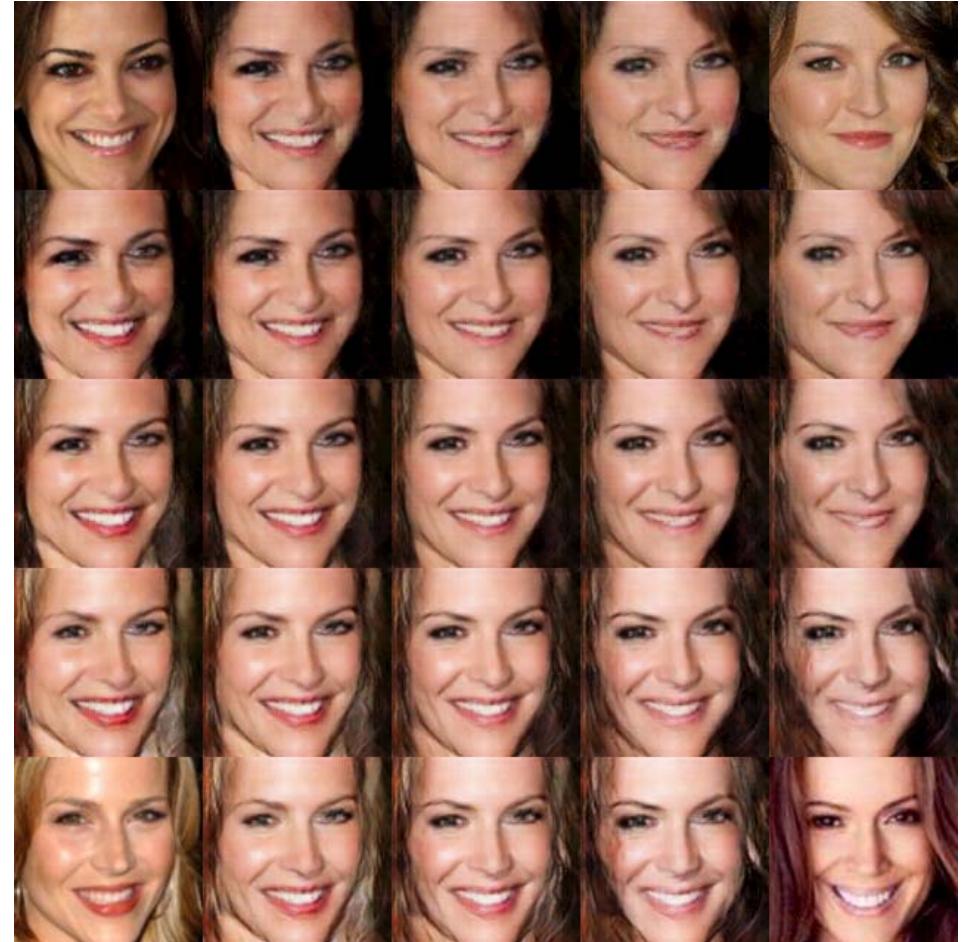
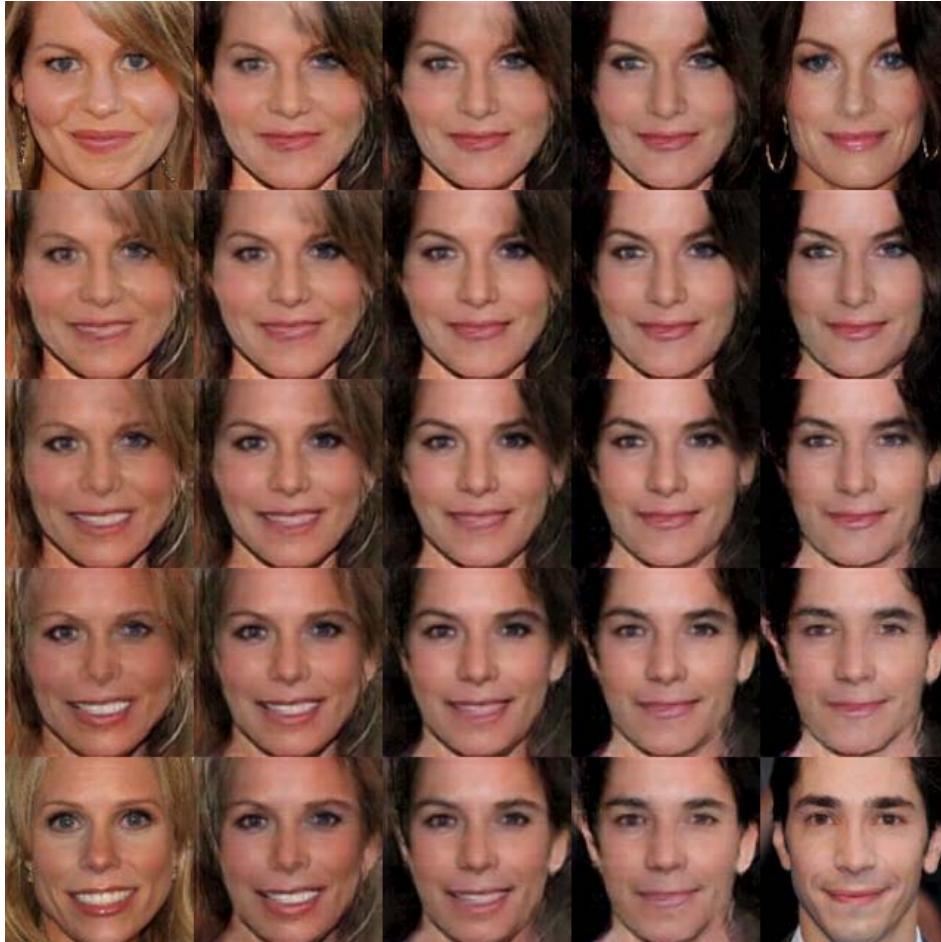
# Image Synthesis & Morphing

[Blum, Brattoli, Ommer, GCPR'18]



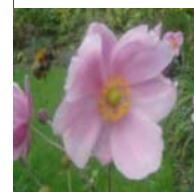
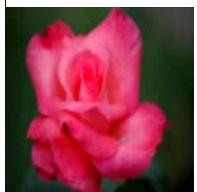
# Image Synthesis & Morphing

[Blum, Brattoli, Ommer, GCPR'18]



# Image Synthesis & Morphing

[Blum, Brattoli, Ommer, GCPR'18]

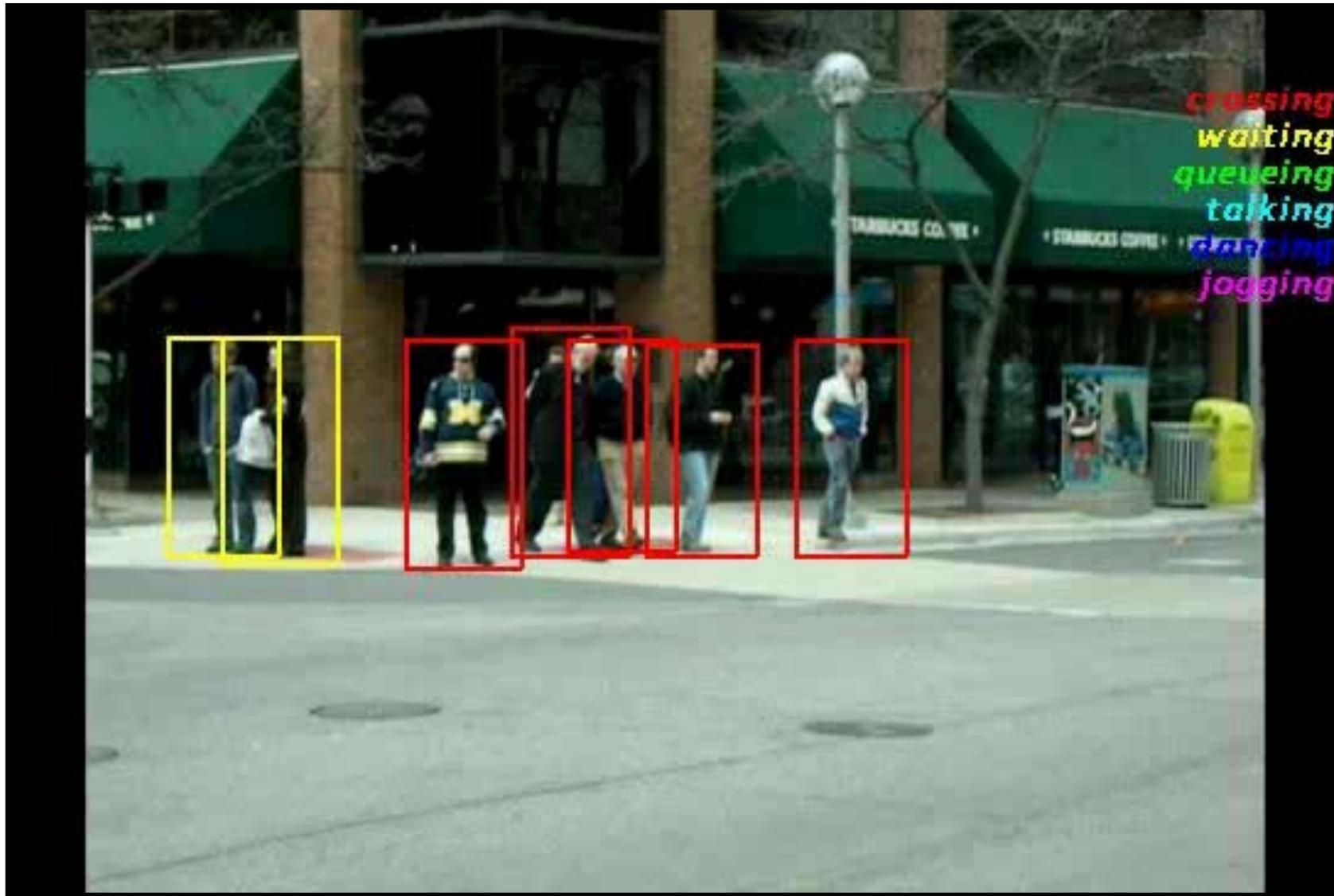




# Search

# Activity Recognition

[Antic & Ommer, ECCV'14]



crossing

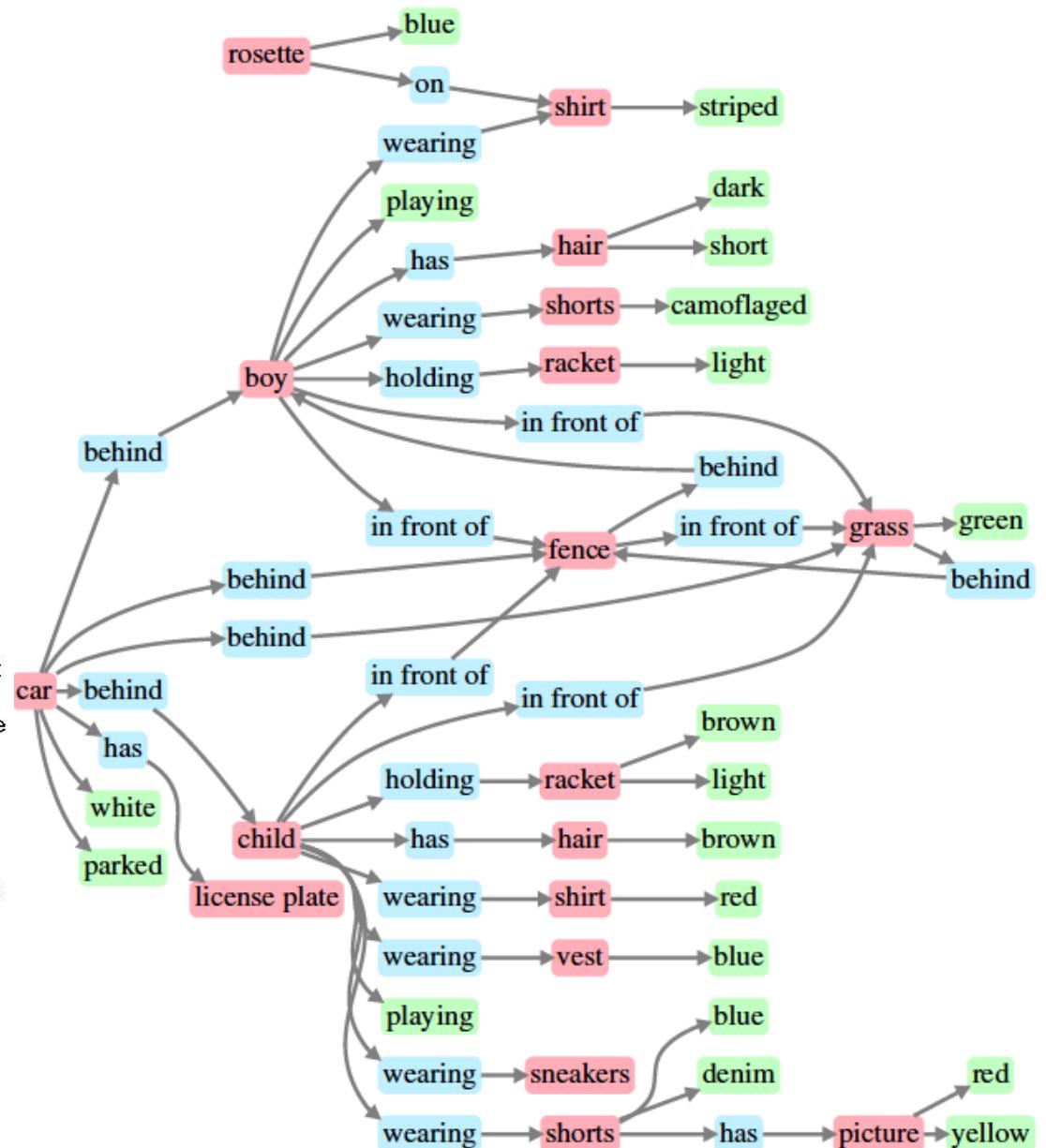
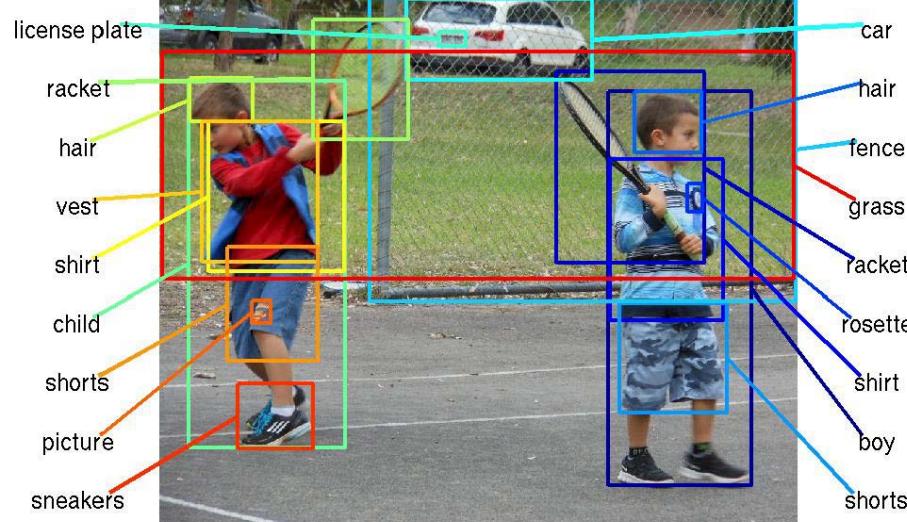
waiting

queueing

talking

dancing

jogging



# Scene Captioning



Human performance:  
**PT = 500ms**

Some kind of game or fight. Two groups of two men? The foreground pair looked like one was getting a fist in the face. Outdoors seemed like because i have an impression of grass and maybe lines on the grass? That would be why I think perhaps a game, rough game though, more like rugby than football because they pairs weren't in pads and helmets, though I did get the impression of similar clothing. maybe some trees? in the background. (Subject: SM)

Machine performance:



"little girl is eating piece of cake."



"baseball player is throwing ball in game."



"construction worker in orange safety vest is working on road."



"a cat is sitting on a couch with a remote control."

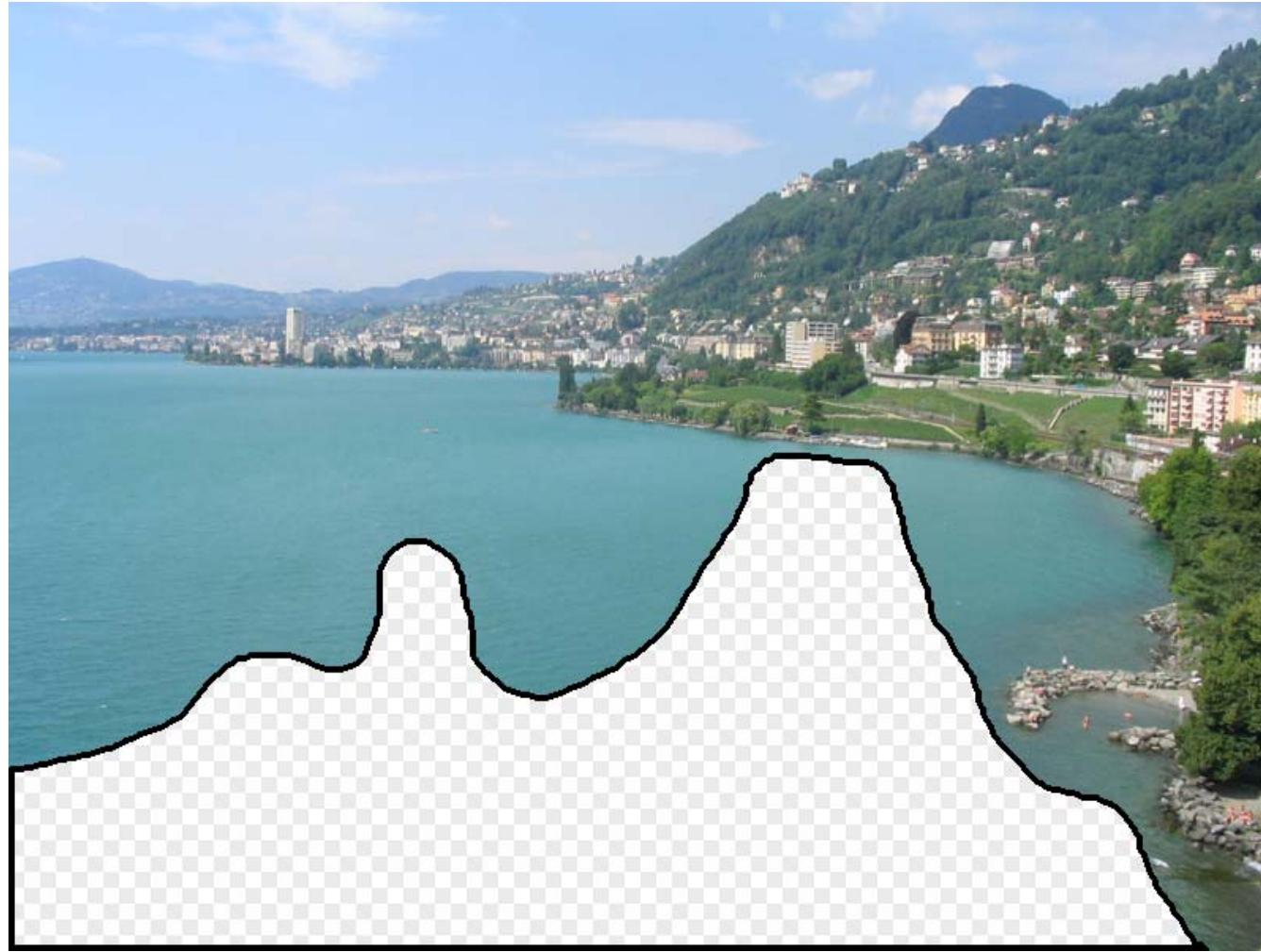


"a young boy is holding a baseball bat."

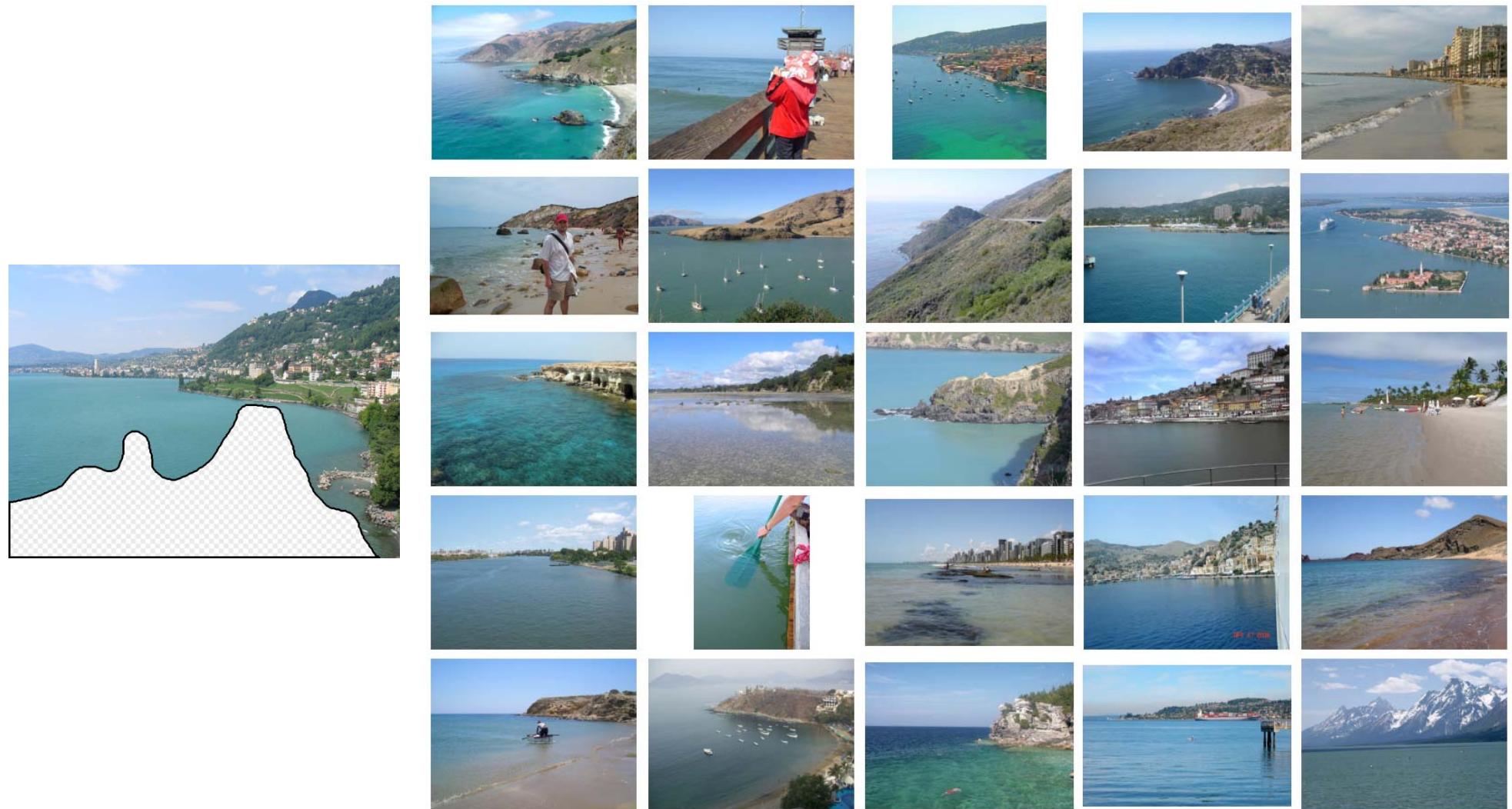


"a horse is standing in the middle of a road."

# Scene Completion



[Hays and Efros. Scene Completion Using Millions of Photographs.  
SIGGRAPH 2007 and CACM October 2008.]

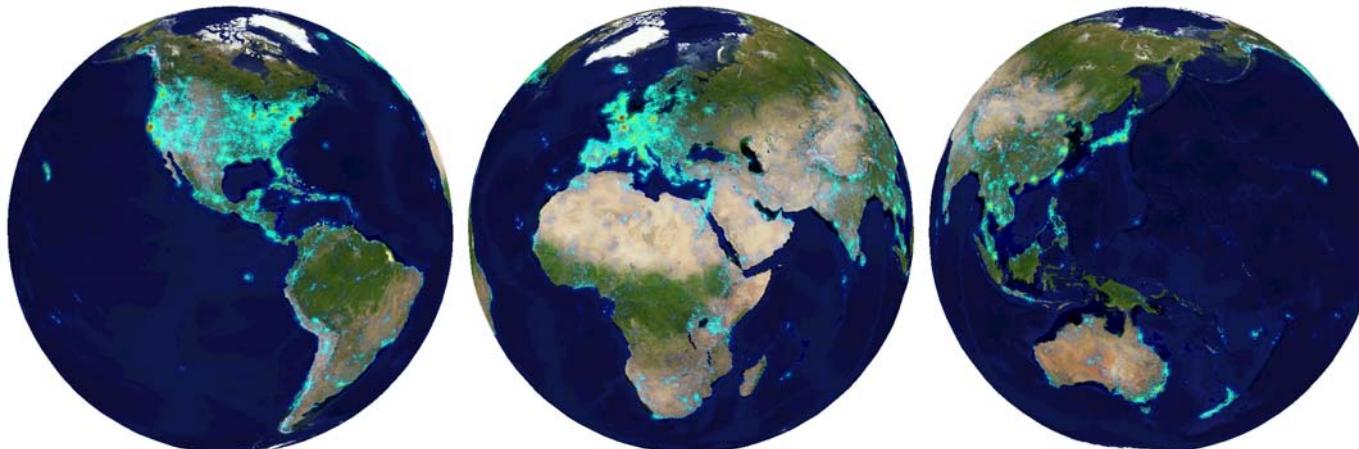


Nearest neighbor scenes from  
database of 2.3 million photos



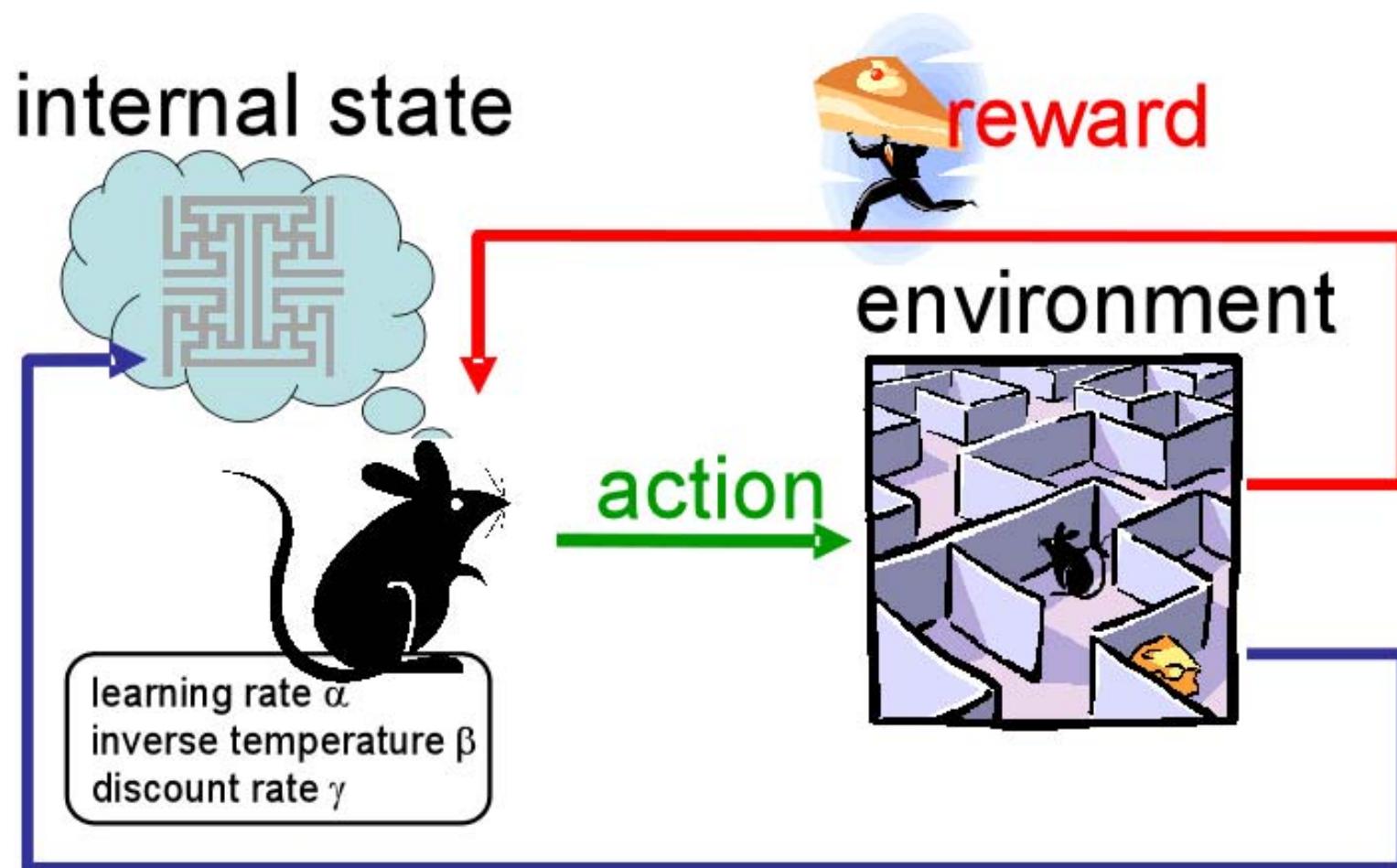
Graph cut + Poisson blending

# Image Localization



Hayes, Efros, CVPR'08

# Learning



observation

# Eine kurze Geschichte der Informatik



# Was ist Informatik?

## Kunstwort

- Informatik: Information + Mathe~~matik~~
- (alternativ: "Information" + "Automatique")

## Engl.: Computer Science

Die Informatik ist gekennzeichnet durch „algorithmisches Denken“, d.h. die Forderung, jedes Problem konstruktiv, durch Angabe eines konkreten Lösungswegs, zu lösen.  
„Algorithmus“ Al-Chwarizmi, der Namensgeber des Algorithmus

# Informatik

**Grundlagen waren**

- **Theorie der Berechenbarkeit (Turing, Church, 1937)**
- **Entwicklung elektromechanischer/elektronischer Rechenmaschinen (Z3, 1941, ENIAC, 1946)**
- **entsprechende Anwendungen (Kryptographie, ballistische Berechnungen, Differentialgleichungen lösen, automatisierte Steuerung, ...)**



# Was ist Informatik?

Wissenschaft von der systematischen Verarbeitung von Information, besonders der automatischen Verarbeitung mit Hilfe von Digitalrechnern

Wikipedia, Duden Informatik

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes.

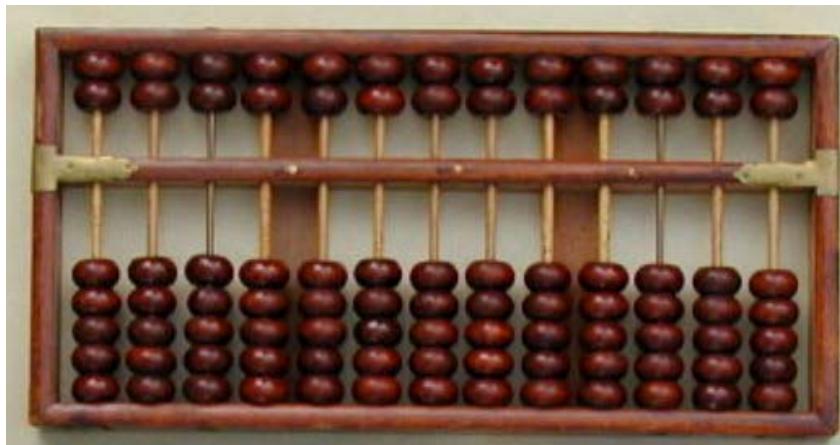
Michael R. Fellows and Ian Parberry, Computing Research News, January 1993



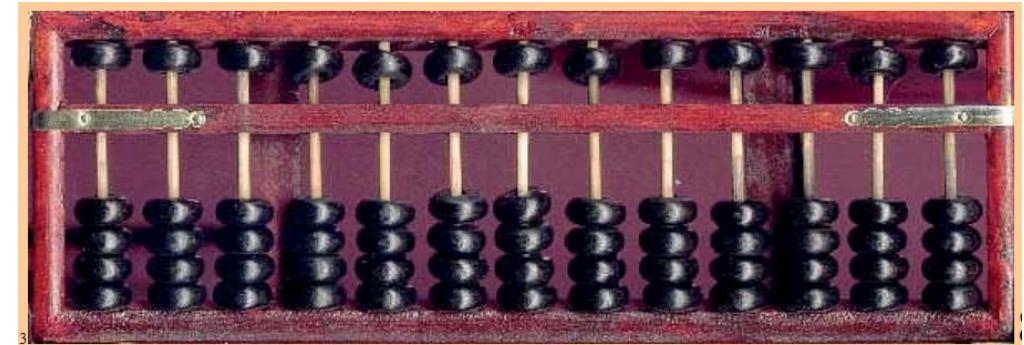
# Der lange Weg zum Computer...

# Entwicklung des Abakus

- vor mehr als 3000 Jahren erste mechanische Geräte zur Unterstützung von Berechnungen



Chinesischer Abakus (Suanpan) 5 + 2



Japanischer Abakus (Soroban) 4 + 1

# Erste mechanische Rechenmaschinen

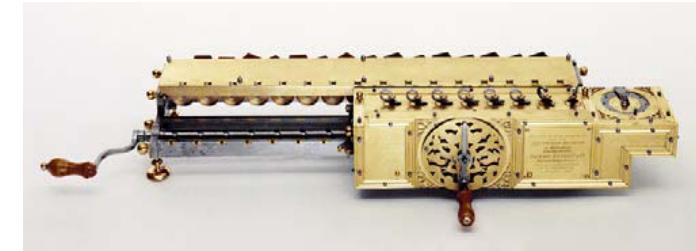


- 1623 konstruierte Wilhelm Schickard für Kepler eine Maschine, die die vier Grundrechenarten ausführen konnte, blieb jedoch unbeachtet

- 1641 baute Blaise Pascal eine Maschine zur Addition sechsstelliger Zahlen (Pascaline)



- 1674 konstruierte Gottfried W. Leibnitz eine Maschine zur Ausführung der vier Grundrechenarten; benutzte erstmalig binäre Darstellung von Zahlen



- 1774 baute Philipp Hahn die erste zuverlässig arbeitende Rechenmaschine

# Erste "programmgesteuerte" Rechenmaschinen

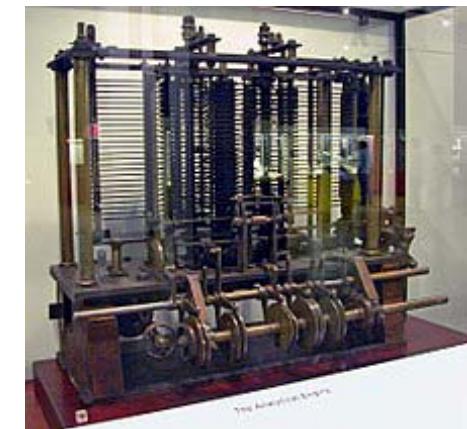
Charles Babbage (1792 - 1871):

- "Differential Engine" zur Berechnung von Polynomen
- Konzeptioneller Entwurf der "Analytical Engine" (1837) für allgemeine Berechnungen; sollte durch Lochplättchen aus Holz gesteuert werden (wie Webstühle)  
nur einzelne Komponenten wurden damals gebaut;  
Entwurf wird aber als korrekt angesehen



Herman Hollerith (1860 - 1929)

- Erfinder der Lochkarte - für Volkszählung 1890 (Vorläufer späterer Lochkarten)
- gründete die Tabulating Machines Corp., die nach Fusion etc. 1924 zu IBM wurde



Problem der bisherigen Maschinen:  
Komplexe Feinmechanik!

1	1	3	0	2	4	10	On	S	A	C	E	a	e	g	EB	SB	Ch	Sy	U	Sh	Hx	Br	Rm
2	2	4	1	3	E	15	Off	15	B	D	F	b	d	f	SY	X	Fp	Cn	R	X	Al	Gg	Kg
3	0	0	0	0	W	20		0	0	0	J	0	0	0	0	0	0	0	0	0	0	0	0
A	1	1	1	1	0	25	A:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	2	2	2	2	5	30	B:	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
C	3	3	3	3	0	3	C:	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	4	4	4	4	1	4	D:	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
E	5	5	5	5	2	G	E:	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
F	6	6	6	6	A	D	F:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
G	7	7	7	7	B	E	G:	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
H	8	8	8	8	A	F	H:	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
I	9	9	9	9	0		I:	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

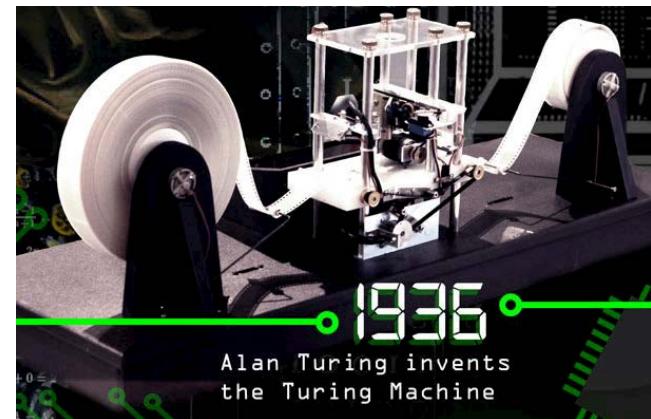
# Ada Lovelace (1815 - 1852)

- veröffentlichte als Erste ein komplexes Programm, geschrieben für die Analytical Engine von Babbage
- wesentliche Aspekte späterer Programmiersprachen, z.B. Unterprogramme und Verzweigungen
- wird heute als erste(r) Programmierer(in) der Welt bezeichnet
- nach ihr benannt:
  - Programmiersprache Ada
  - Lovelace Medal & Ada Lovelace Award

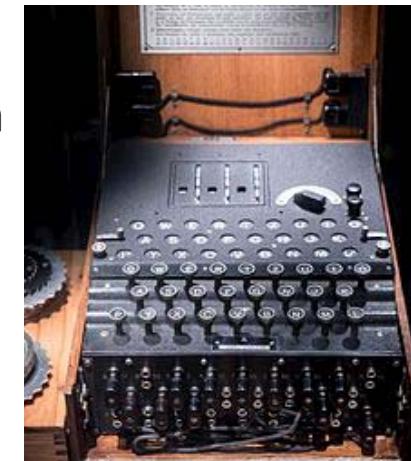


# Alan Turing (1912 - 1954)

- britischer Mathematiker, Informatiker & Kryptoanalytiker
- Erfindung der Turingmaschine  
(wichtiges Rechenmodell der theoretischen Informatik)



- im zweiten Weltkrieg maßgeblich an der erfolgreichen Dechiffrierung der mit Hilfe der Enigma verschlüsselten deutschen Funksprüche beteiligt



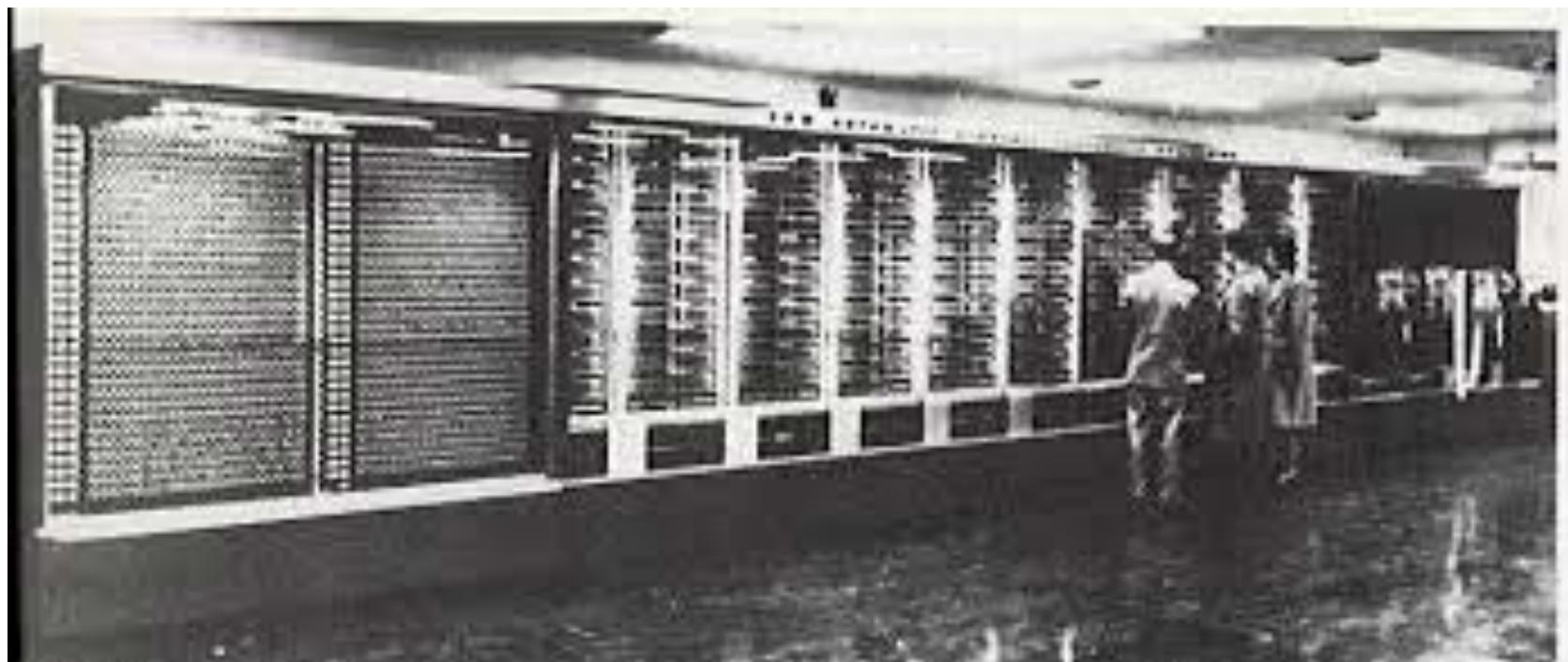
# Konrad Zuse (1910 - 1995) und der erste Computer

- Z1 und Z2 (30er Jahre) : funktionierten nicht zuverlässig
- Z3 (1941): erste funktionsfähige programmgesteuerte Rechenautomat der Welt
  - Zeit für eine Multiplikation: 3 Sekunden
  - Z3 und Unternehmen von Zuse wurden im Krieg zerstört
- 1941 - 45: Nachfolger Z4 und Versuch einer universellen Programmiersprache: Plankalkül



# Howard Aiken und die Mark I

- 1944: Howard Aiken erstellt zusammen mit der Harvard University & IBM die Rechenanlage Mark I
  - 100.000 Teile, 15 Meter lang!
  - Additionszeit: 1/3 sec, Multiplikation 5 sec



# John von Neumann (1903-57)

- von Neumann Architektur (1945):  
Daten & Programme im Speicher
- Spieltheorie
- Manhattan Projekt
- ...



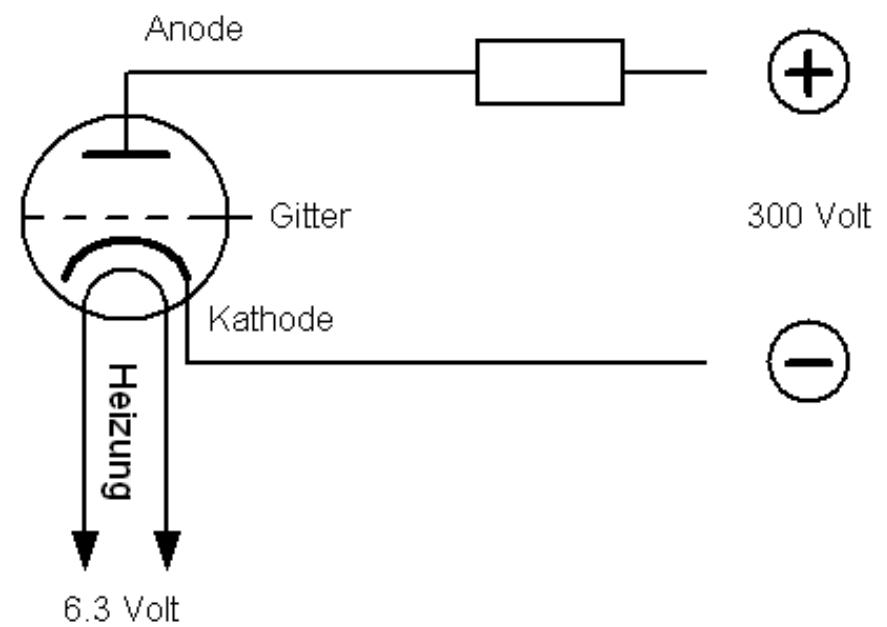
# Grace Hopper

- hatte in den 40er Jahren die Idee, Computerprogramme in verständlicherer (Programmier-)Sprache (und nicht nur Einsen und Nullen) zu verfassen
- entwickelte 1952 den ersten Compiler (A-0) für die UNIVAC
- „First actual case of bug being found.“ (Tote Motte sorgte für Ausfall des Computers)



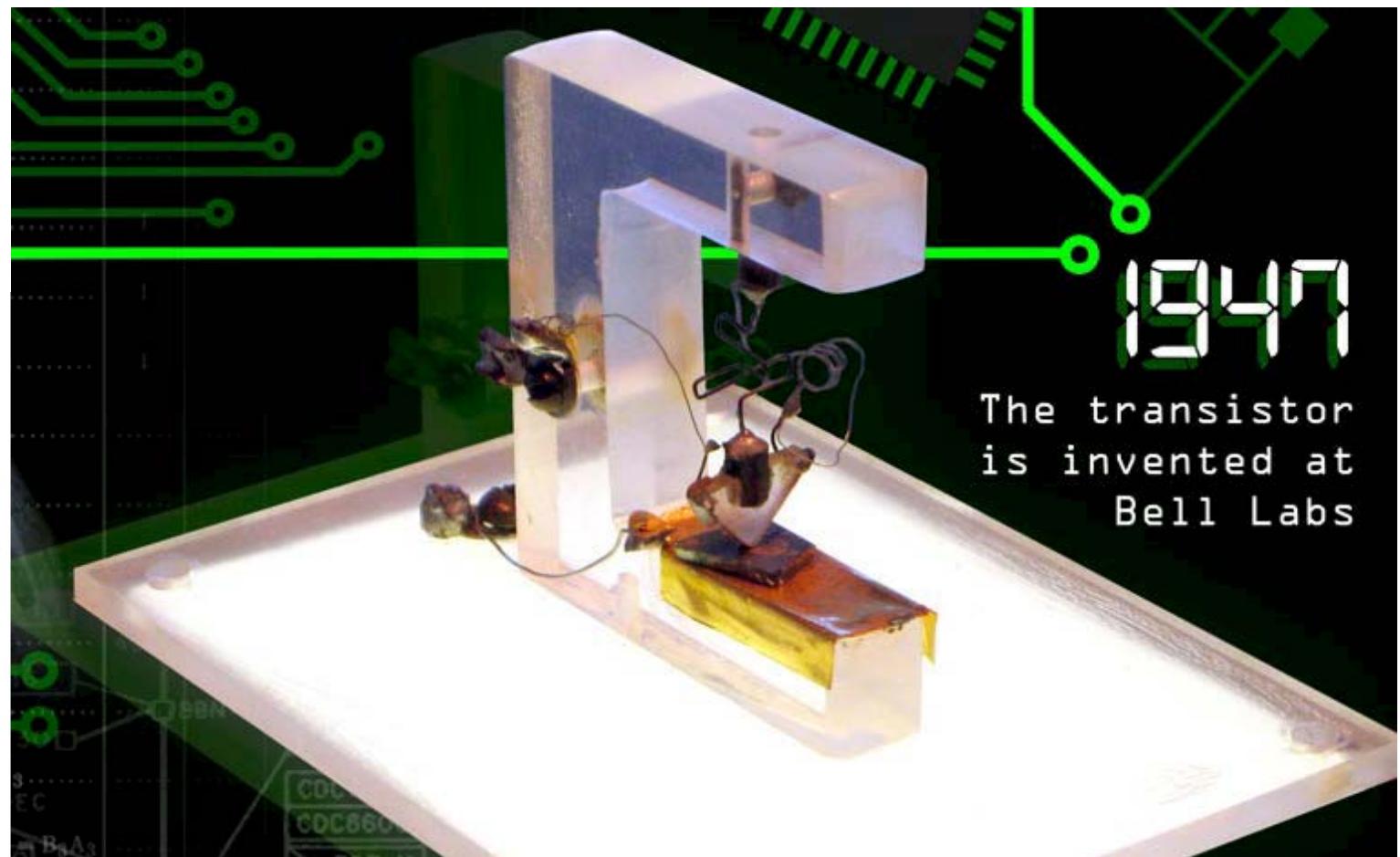
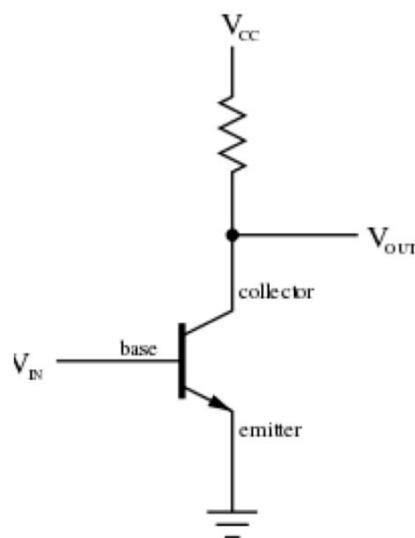
9/9  
0.800 Antran started  
1.000 stopped - antran ✓  
13"uc (03) MP-MC  
(03) PRO 2  
conck  
Prelays 6-2 in 033 fault special speed test  
in Prelay  
Prelays changed  
11.00 Started Cosine Tape (Sine check)  
15.25 Started Multi Adder Test.  
15.45 Relay #70 Panel F  
(moth) in relay.  
First actual case of bug being found.  
16.00 Antran started.  
17.00 closed down.

# Elektrische Schalter



# Erfindung des Transistors

- erste Patente bereits 1925 (Lilienfeld) und 1934 (Hell)
- Bau des ersten funktionsfähigen Transistors in Bell Labs



# Weitere wichtige Erfindungen



?

erste Computermaus  
von Douglas Engelbart  
(1963)

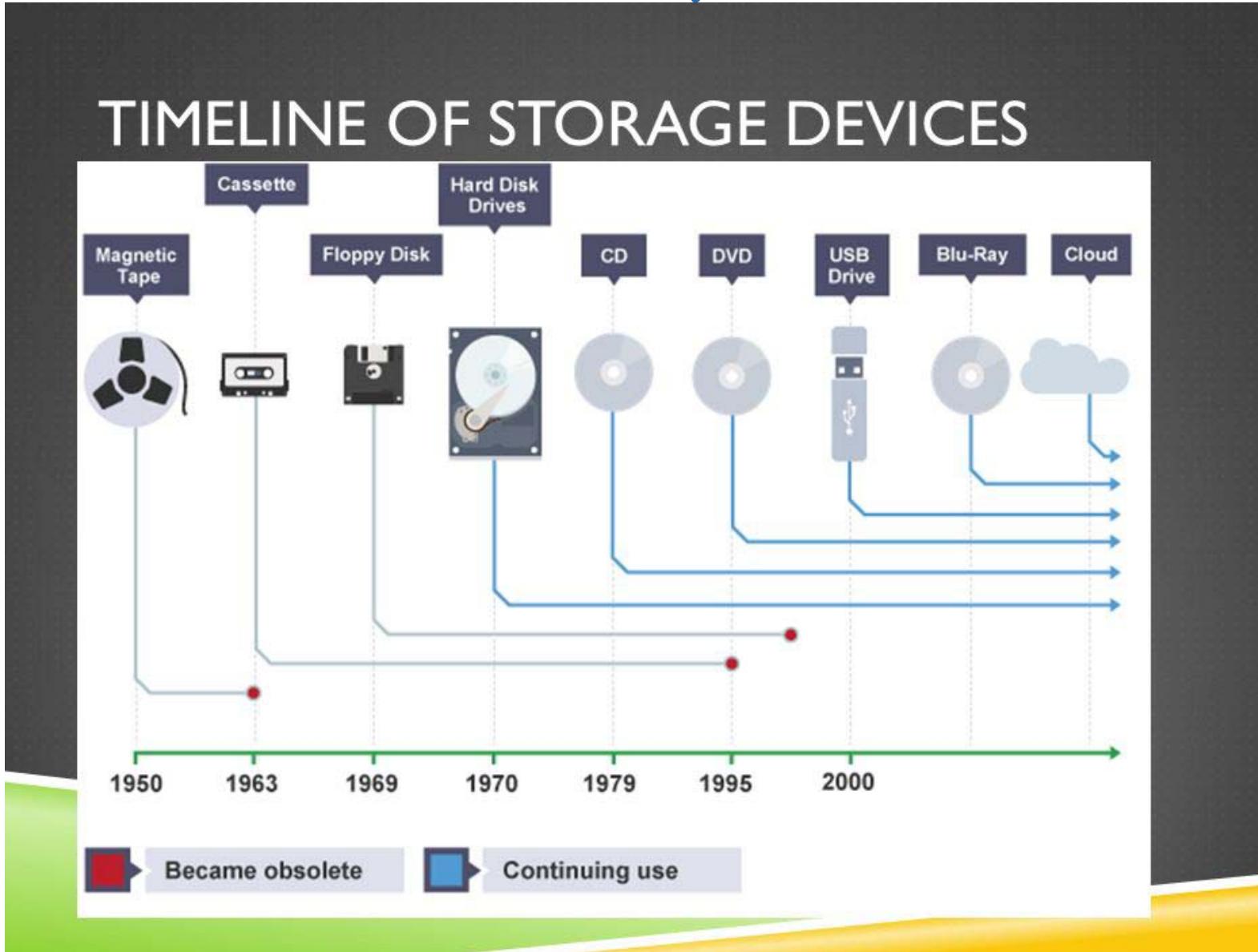


?



Speichermedien  
Floppy disks  
(ab 1969)

# Geschichte der Speichermedien



# Erstes Computerspiel



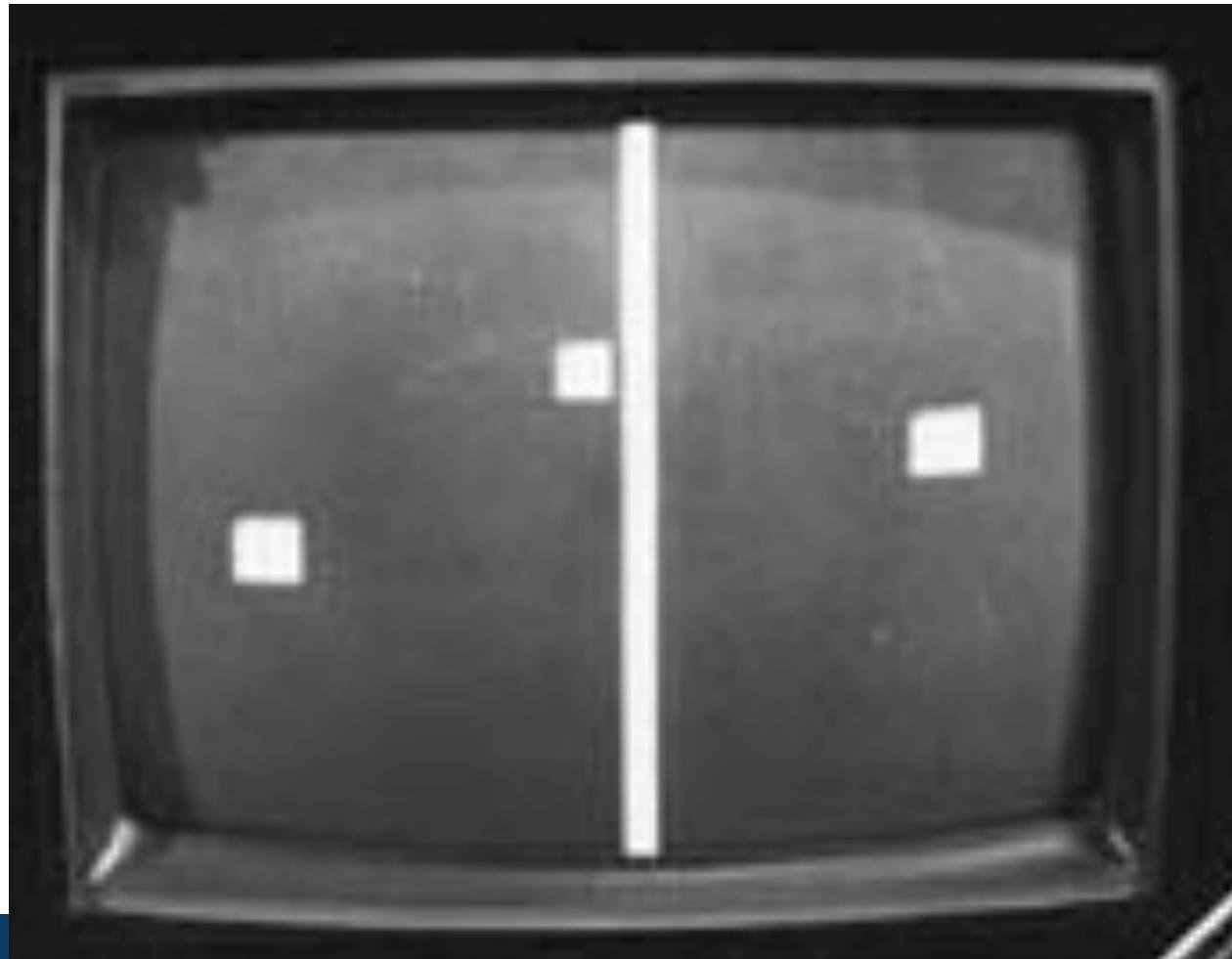
Erstes Videospiel für Computer:  
Spacewar,  
entwickelt am MIT, 1962



lief auf "Minicomputer"  
PDP-1  
(so groß wie zwei Kühlschränke)

# Erstes Computerspiel “für alle”

- Pong, Atari, 1972



# Entwicklung des Personal Computers

Apple 1 von Steve Wozniak  
(1976)



IBM PC (1981)



# Entwicklung des Internets

- ab 60er Jahre Entwicklung der ersten Netzwerk-Technologien, Einzelnetze
- 1969: ARPANET - erste Kommunikation zwischen Netzwerken verschiedener Forschungseinrichtungen (Internet)
- 1971 - erste Email (1984 - erste Email in Deutschland)
- 70er/80er Jahre - Entwicklung von einheitlichen Protokollen; Internet wird nur von wenigen Forschungseinrichtungen genutzt
- 1985 erster Domainname vergeben
- 80er Jahre in Frankreich: Minitel
- 1990: Abschaltung des Arpanets, Kommerzielle Nutzung des Internets beginnt / World Wide Web
- 1995 Amazon
- 1997 Google

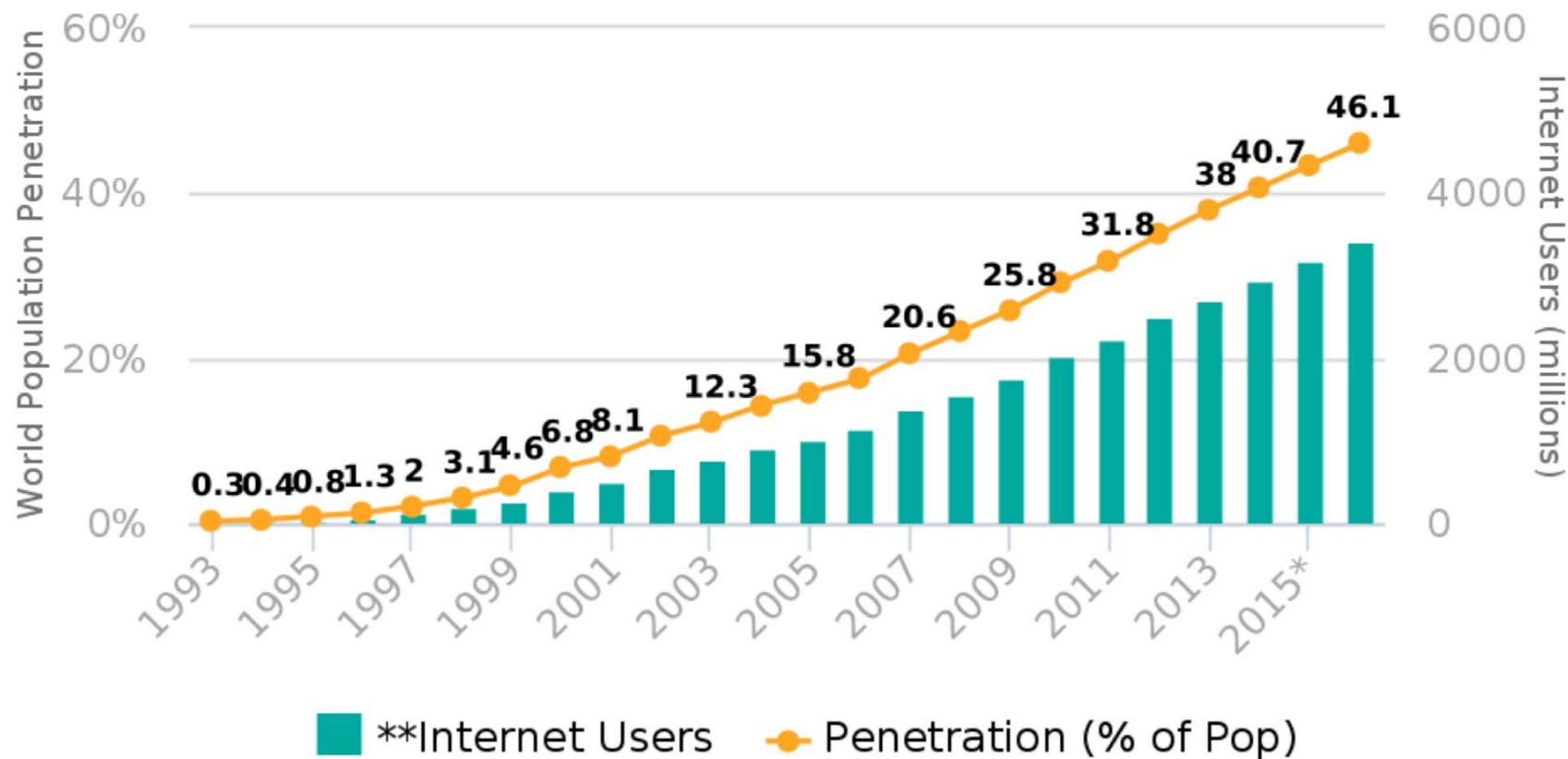


# Entwicklung des Internets

## Global Internet Users and Penetration Rate

\* estimate for July 1, 2016

\*\* **Internet User** = individual who can access the Internet at home, via any device type and connection.



# Künstliche Intelligenz

- Deep Blue vs. Gary Kasparov:

In 1997, the IBM-chess computer Deep Blue was the first computer system to defeat a reigning chess world champion (then Garry Kasparov) in a match under standard chess tournament conditions.

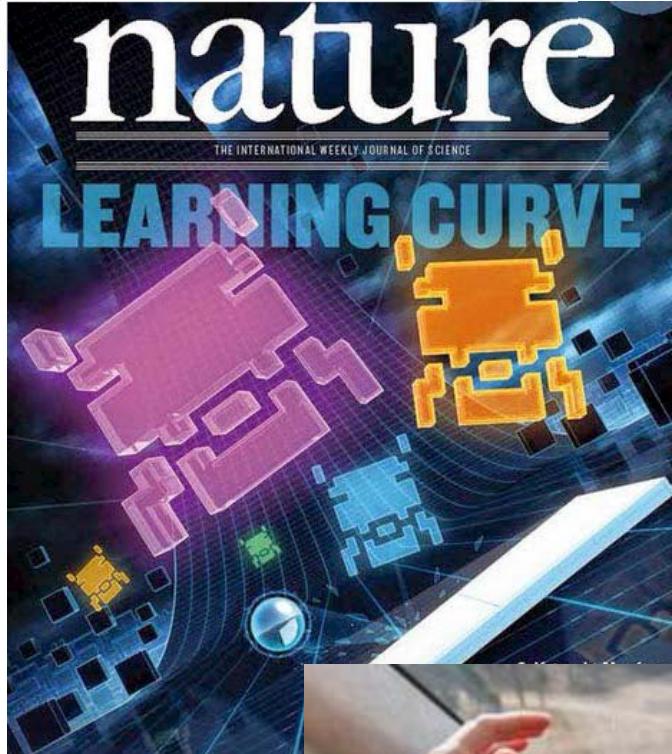


# Künstliche Intelligenz

- 2007: Jeopardy / Natural Language  
Watson by IBM
- 2016: Go  
AlphaGo by DeepMind / Alphabet
- 2017: Poker  
Lengpudashi by CMU / Strategic Machine



# KI und Deep Learning



Google  
Translate

# Zukunft der Künstlichen Intelligenz

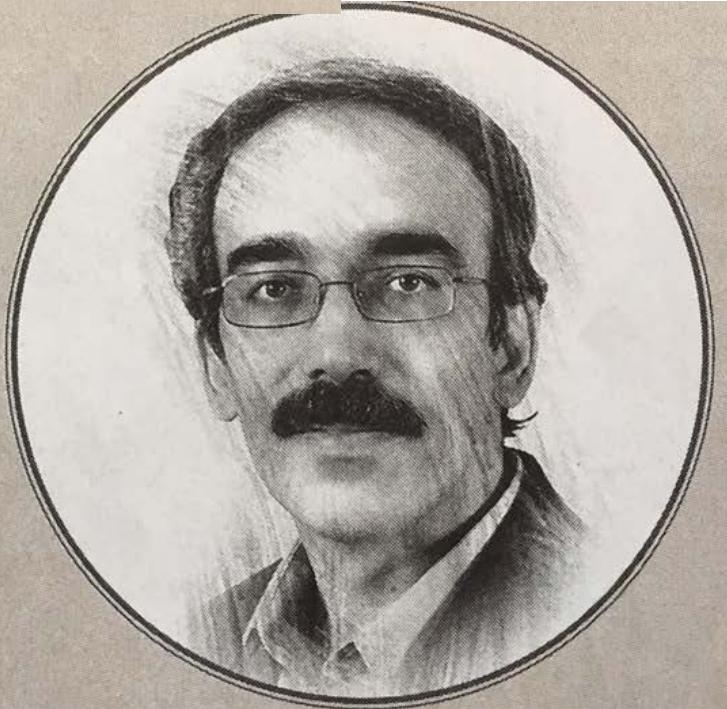
- Wann werden Computer die Fähigkeiten des menschlichen Gehirns haben?

## Ruchir Puri

*Chief architect, IBM Watson*

**When will we have computers as capable as the brain?**

A human brain is fundamentally different than being a champion chess, “Jeopardy!,” or Go player. It is something that entails essential traits like caring, empathy, sharing, ingenuity, and innovation. These human brain traits might prove to be elusive to machines for a long time, not to mention the incredible form factor and energy efficiency with which [the] human brain operates and learns. Although AI’s impact on society will accelerate further...to reach new heights in solving problems which are currently

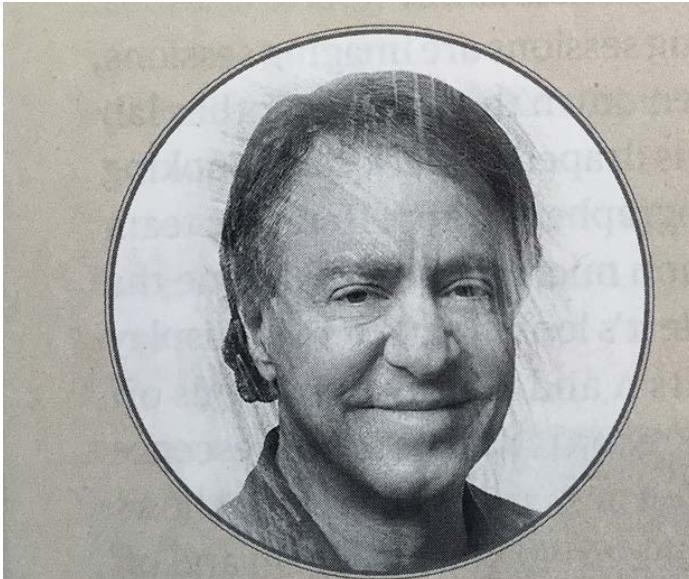


grand challenges for human society, it will be a while before we will be able to holistically answer [that] question.

# Vorhersagen der KI Zukunft

## Ray Kurzweil

Cofounder and chancellor,  
Singularity University



**When will we have computers as capable as the brain?**  
I believe computers will match and then quickly exceed human capabilities in the areas where humans are still superior today by 2029.

## Rodney Brooks

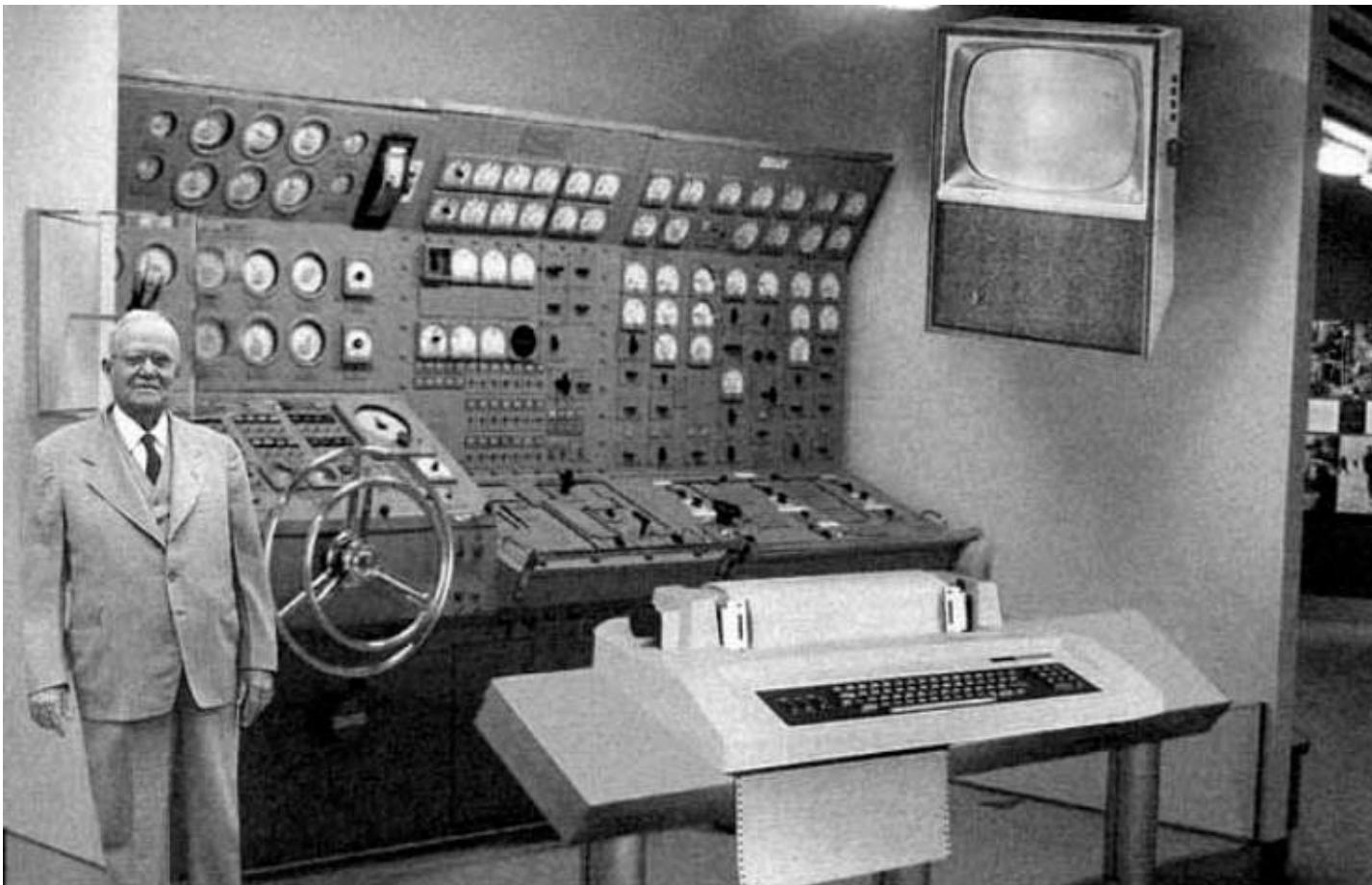
Chairman and CTO, Rethink Robotics

**When will we have computers as capable as the brain?**  
**Rodney Brooks's revised question:**  
**When will we have computers/robots recognizably as intelligent and as conscious as humans?**  
Not in our lifetimes, not even in Ray Kurzweil's lifetime, and despite his fervent wishes, just like the rest of us, he will die within just a few decades. It will be well over 100 years before we see this level in our machines. Maybe many hundred years.

**As intelligent and as conscious as dogs?** Maybe in 50 to 100 years. But they won't have noses anywhere near as good as the real thing. They will be olfactorily challenged dogs.



# Falsche Vorhersagen



RAND Corp. (1954) : ein PC im Jahr 2004!!

# Vorhersagen über Zukunft der Informatik

- "640 K ought to be enough for anybody."  
Author unknown, (wrongly) attributed to Bill Gates
- "Two years from now, spam will be solved." -  
Bill Gates, 2004 World Economic Forum
- "There is no reason anyone would want a computer in their home."  
Ken Olsen, 1977 (President of Computer Company DEC)
- "We will never make a 32-bit operating system."  
Bill Gates, 1989 (after the first 16 bit systems had been introduced)

... und noch eine richtige, aber etwas konservative Vorhersage:

- "Computers in the future may weigh no more than 1.5 tons."  
Popular Mechanics, 1949  
(der ENIAC-Computer zu dieser Zeit wog 30 Tonnen)

# Vorhersagen über Zukunft der Informatik



There's no chance that the iPhone is going to get any significant market share.  
Steve Ballmer, Microsoft CEO, 2007

# Teilgebiete der Informatik

- **Theoretische Informatik**  
Logik und Berechenbarkeit, Automatentheorie und formale Sprachen, Semantik, Komplexitätstheorie
- **Technische Informatik**  
Elektrotechnische Grundlagen, Architektur von Rechenanlagen, Chipentwurf, Netzwerkkomponenten, Fehlertoleranz, . . .
- **Praktische Informatik**  
Betriebssysteme, Softwareengineering, Datenbanken, Programmiersprachen
- **Angewandte Informatik**  
Machine Learning, Computer Vision, Robotik, Visualisierung, Mensch-Maschine-Interaktion, interdisz. Anwendungen, ...



# Informatik in Heidelberg

**Institut für  
Informatik  
IFI**

**Institut der Fakultät  
Mathematik & Informatik**

**Zentrales Institut für  
Technische Informatik  
ZITI**

**Zentrale Einrichtung**

**Interdisziplinäres  
Zentrum für  
Wissenschaftliches  
Rechnen  
IWR**

**Zentrale Einrichtung**

# Informatikstudiengänge in HD

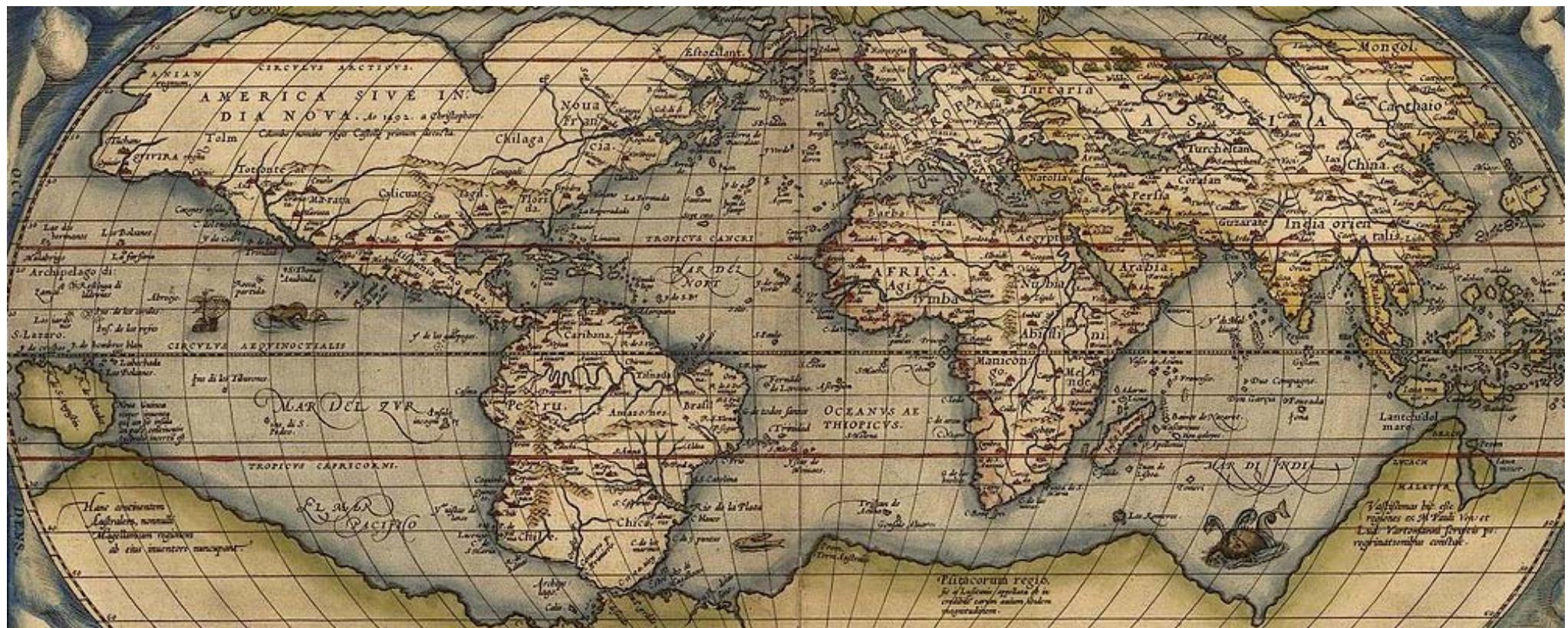
Master  
Angewandte  
Informatik

Master  
Technische  
Informatik

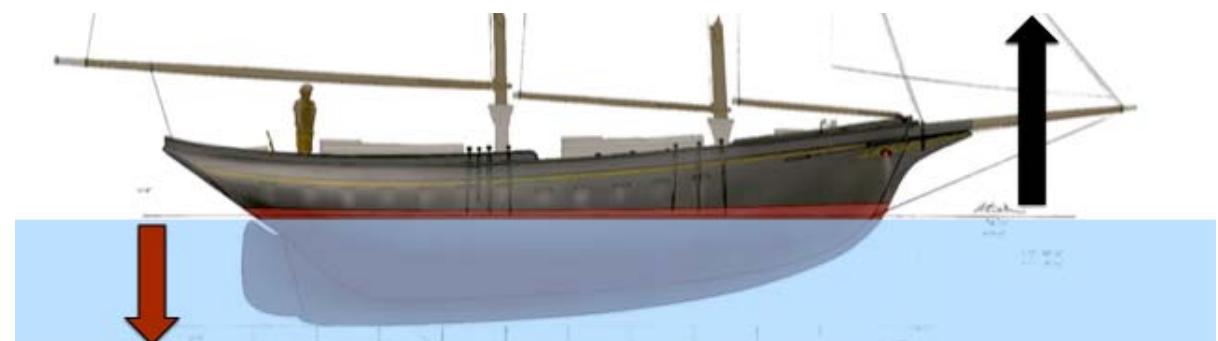
Master  
Scientific  
Computing



Bachelor Angewandte Informatik



# Building Below the Waterline



**Viel Erfolg und Freude für Ihr Studium  
der Informatik in Heidelberg!**

