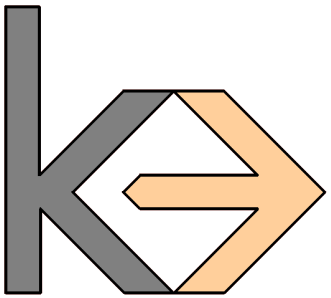


Learning to Compress Images and Videos

von Li Cheng &
S.V. N. Vishwanathan

vorgetragen von Michael Wächter

Motivation

- Bild- & Video-Kompression
  - herkömmliche Verfahren sind frequenzbasiert
  - Aufsatz auf herkömmlichen Verfahren
  - zusätzlicher Platzgewinn ohne Qualitätsverlust
- SW-Bild- & SW-Video-Kolorierung
  - herkömmliche Verfahren zeitaufwändig und manuell
  - jetzt semiautomatisch

Motivation

- Bild-Kompression
  - Auswahl repräsentativer Pixel
  - Lernen eines Farbvorhersage-Modells
  - Speichern des SW-Bilds + Farbpixel
  - Rekonstruktion des Farbbilds
- Video-Kompression analog

Motivation

- SW-Bild- & SW-Video-Kolorierung
  - wie Kompression
  - automatische Pixelauswahl fällt weg
  - Farbinfos werden stattdessen vom Benutzer vorgegeben

Überblick

- Motivation
- Begriffsklärung
- Funktionsweise
  - Kolorierung mit Semi-Supervised Learning
  - Farbpixelauswahl mit Active Learning
- Experimente
- Fazit
- Bemerkungen

Begriffsklärung

- Semi-Supervised Learning

  X *Beobachtungsraum*
  $Y \subset \mathbb{R}$ *Labelraum*
  $\{(x_i, y_i)\}_{i=1}^{m}$ *gelabelte Beispiele*
  $\{x_i\}_{i=m+1}^{n} \subset X$ *ungelabelte Beispiele*
  $f \in H$ *zu minimierende Zielfunktion*
  $l : X \times Y \times H \to \mathbb{R}$ *Loss−Funktion*

- Graph-basierte Methoden:
  - Beispiele als Knoten (gelabelte und ungelabelte)
  - Nachbarschaftsbeziehungen als Kanten
    - Achtung: Glattheitseigenschaften

Begriffsklärung

- Graph

  $G = (V, E \subseteq V \times V)$

  - ungerichtet, gewichtet

- Adjazenz-Matrix

  $W \; mit \; W_{ij} \in (0, \infty) \; falls \; (v_i, v_j) \in E \; und \; W_{ij} = 0 \; sonst$

- Grad-Matrix

  $D \; mit \; D_{ii} = \sum_j W_{ij}$

Begriffsklärung

- Laplace-Matrix

$$L = D - W$$

- normalisierte Laplace-Matrix

$$\Delta = D^{-1/2} L D^{-1/2}$$

Funktionsweise - Kolorierung

$$minimiere \sum_{i=1}^{n} [f(x_i) - \sum_{i \sim j} w_{ij} f(x_j)]^2 + \sum_{i=1}^{m} l(f(x_i, y_i))$$

$$l(f(x_i), y_i) = 0 \; falls \; f(x_i) = y_i \; und \; \infty \; sonst$$

$$oder$$

$$l(f(x_i), y_i) = (f(x_i) - y_i)^2$$

$$\forall i: \; w_{ij} \geqslant 0 \; und \sum_{i \sim j} w_{ij} = 1$$

- Kantengewichte:
  - räumliche Nachbarschaft und Bildtextur
  - rationale Funktion 2. Grades bzgl. Helligkeitsdifferenz
  - ggf. zeitliche Nachbarschaft

Funktionsweise - Kolorierung

- Laplacian Regularized Least Square algorithm:

$$minimiere\ J(f) = c\|f\|_H^2 + \frac{\lambda}{n^2}\|f\|_G^2 + \frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, f)$$

$$mit\ f = [f(x_1), \ldots, f(x_m), \ldots, f(x_n)],$$

$$\|f\|_G^2 = f^T \nabla_G f = f^T L^2 f\ oder\ f^T \Delta f$$

Funktionsweise - Kolorierung

- Lösung von LapRLS:

$$es\ existieren\ \alpha_i\ so\ ,dass\ f\ (x\ )=\sum_{i=1}^{n}\alpha_i\ k\ (x_{\ i}\ ,x\ )$$

$$\alpha=(I_{\ m}K+cmI+\frac{\lambda m}{n^2}\nabla_g K\ )^{-1}\ y$$

$$mit\ \alpha=(\alpha_1,\ldots,\alpha_m,\ldots,\alpha_n)^T\ ,$$

$$I_{\ m}\in\mathbb{R}^{n,n}\ \ mit\ mxm-Einheitsmatrix\ links\ oben\ und\ 0\ sonst\ ,$$

$$K\ \ \ mit\ K_{\ ij}=k\ (x_{\ i}\ ,x_{\ j}\ ),$$

$$\nabla_G=L^2\ oder\ \Delta$$

$$und\ \ y=(y_{\ 1},\ldots,y_{\ m},0,\ldots,0)^T$$

Funktionsweise - Kolorierung

- Implementationsdetails:
  - YUV-Farbraum, Vorhersage von U und V getrennt
  - Kernel: standard Gaussian kernel (mit Parameter σ)
  - Mean Square Loss statt ∂-Loss
  - Δ statt L^2
  - keine zeitliche Nachbarschaft!
  - Problem: Matrixinvertierung
    - Matrix zur Berechnung von α groß und dicht
    - Berechnung einer Super-Pixel-Repräsentation des Ausgangsbilds ==> 1000-5000 Segmente

Funktionsweise - Pixelauswahl

- automatische Pixelauswahl wird für Handkoloration abgeschaltet
- ansonsten per Active Learning:
  - Lerner wählt Beispiele aus und fragt nach Labels
  - muss dafür Kosten bezahlen (hier: Speicherplatz)
- Programmablauf:
  - Start mit ein paar zufälligen gelabelten Pixeln
  - Lernen des Modells

Funktionsweise - Pixelauswahl

- Bild wird mit Modell vorhergesagt und mit Zielbild verglichen
- Qualitätsmaß: $PSNR = 20 \log_{10} \frac{255}{\sqrt{MSE}}$

$$MSE = \frac{1}{n^2} \sum_{i,j=1}^{n} (I_{ij} - I'_{ij})^2$$

- Fehlerbereiche werden geclustert
- aus jedem Fehlercluster wird ein Pixel gewählt, seine Farbinfo abgefragt und der Labelmenge hinzugefügt
- Abbruchkriterium:
  - PSNR=38 oder 5000 abgefragte Pixel
  - außerdem möglich: PSNR in einem Plateau

Experimente

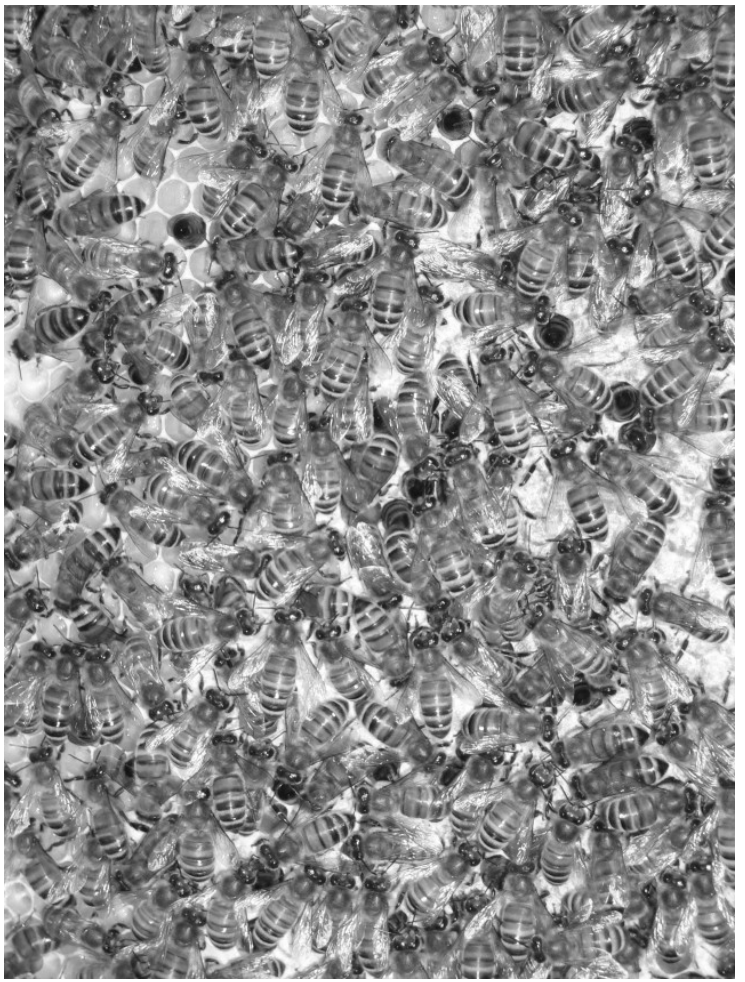
- SW-Bild-Kolorierung

Input with partial color labels

Colorized Output

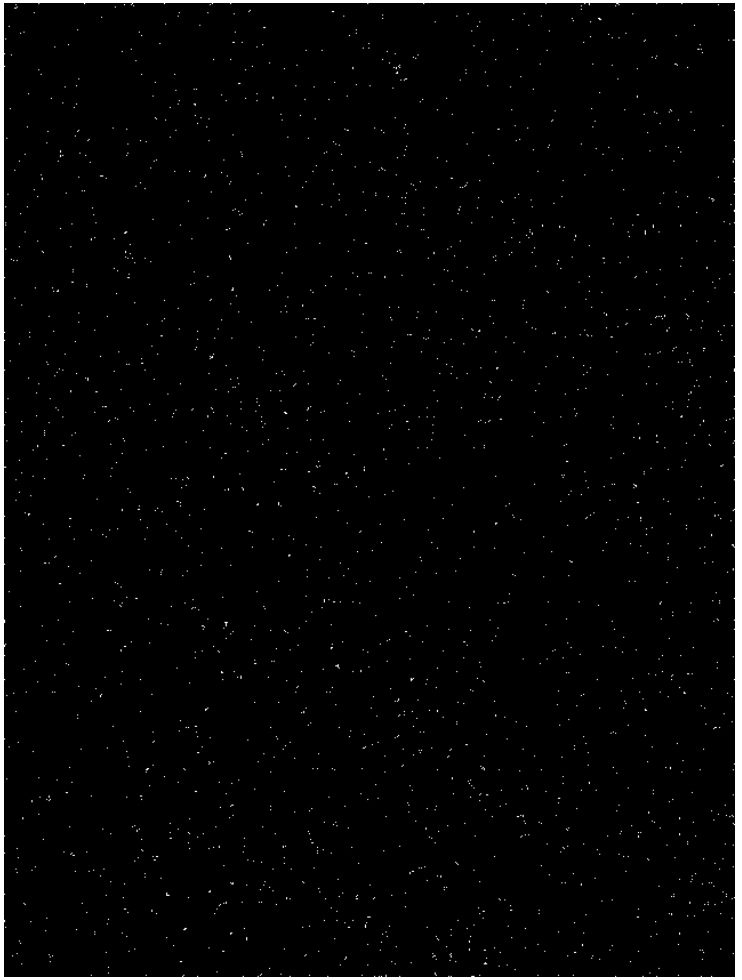
Experimente

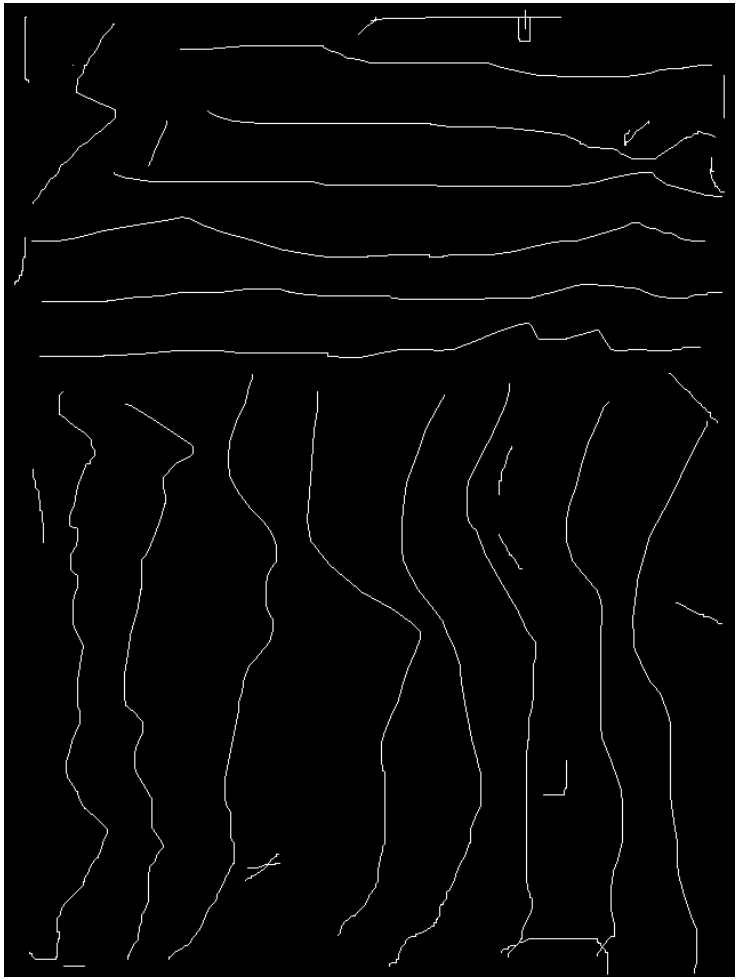
- Vergleich: Active Learning vs. manuelle Pixelauswahl



Experimente

- Vergleich: Active Learning vs. manuelle Pixelauswahl





Experimente

- Vergleich: Active Learning vs. manuelle Pixelauswahl

Experimente

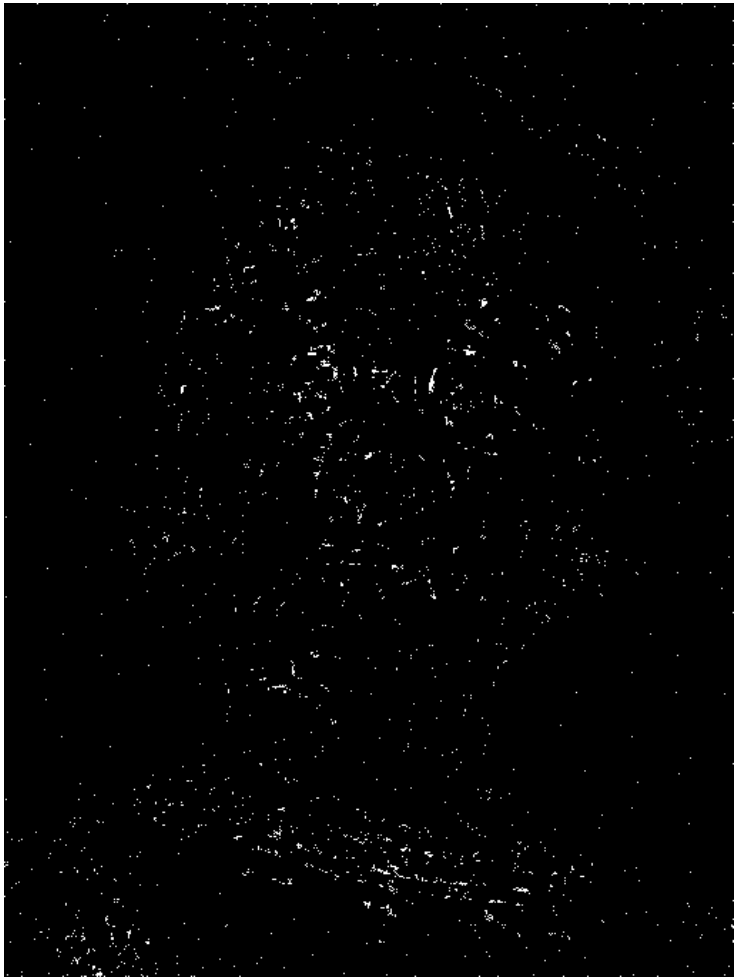
- Ergebnis Bienen:
  - Active Learning
    - PSNR = 31.49
    - 2534 Pixel
    - 7 Iterationen
  - manuell
    - PSNR = 27.00
    - 8558 Pixel

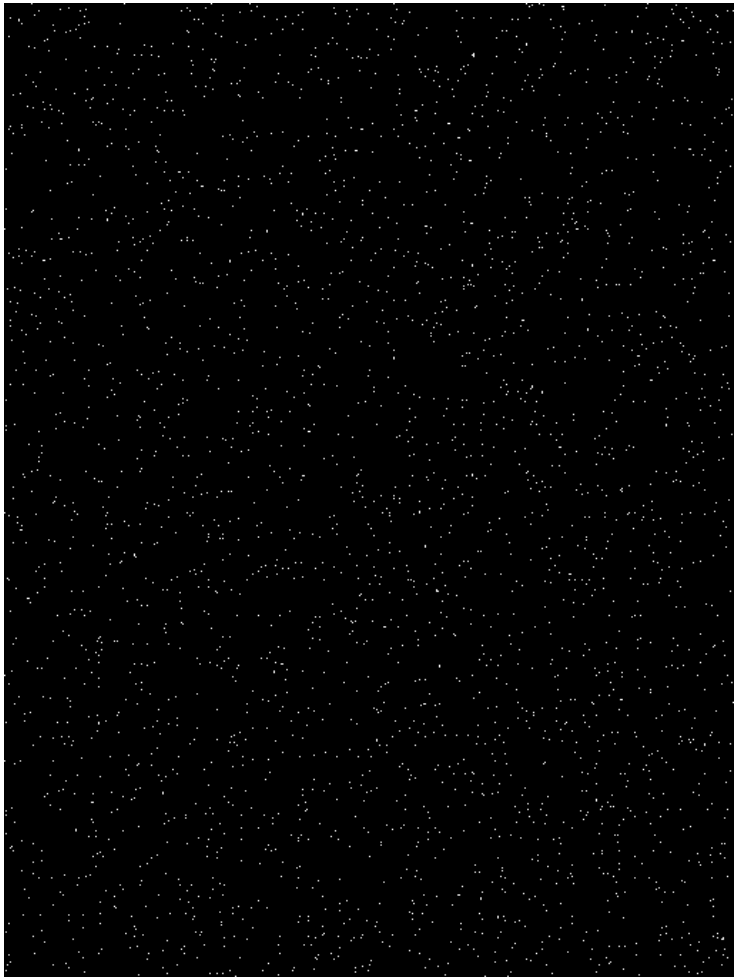
Experimente

- Vergleich: Active Learning vs. zufällige Pixelauswahl

Experimente

- Vergleich: Active Learning vs. zufällige Pixelauswahl





Experimente

- Vergleich: Active Learning vs. zufällige Pixelauswahl

Experimente

- Ergebnis Mädchen:
  - Active Learning
    - PSNR = 40.95
    - 2766 Pixel
    - 17 Iterationen
  - zufällig
    - PSNR = 38.41
    - 2976 Pixel

Experimente

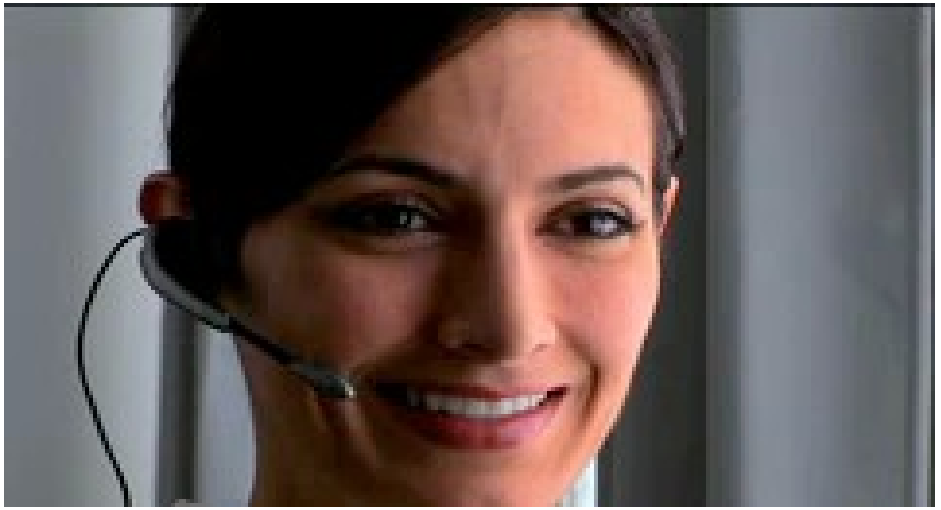
- Kompressionsraten:
  - Bienen: 0.754
  - Mädchen: 0.781

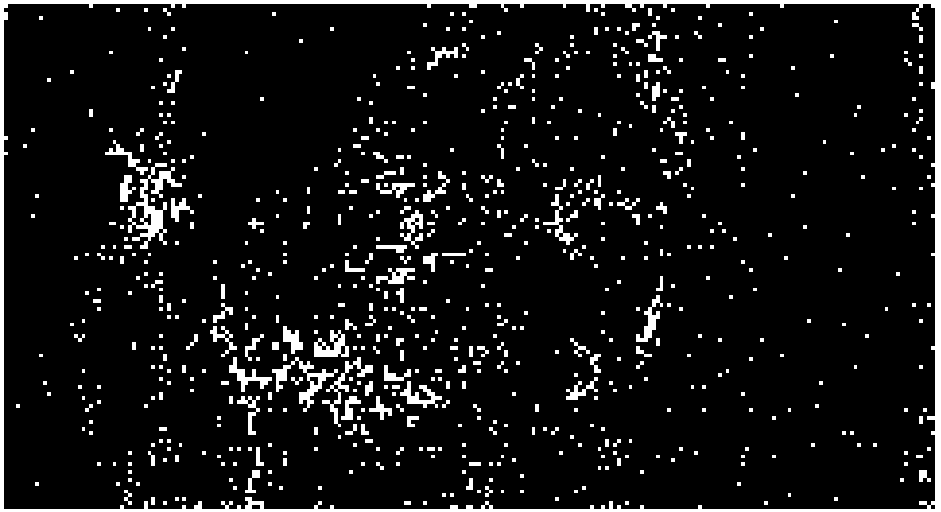
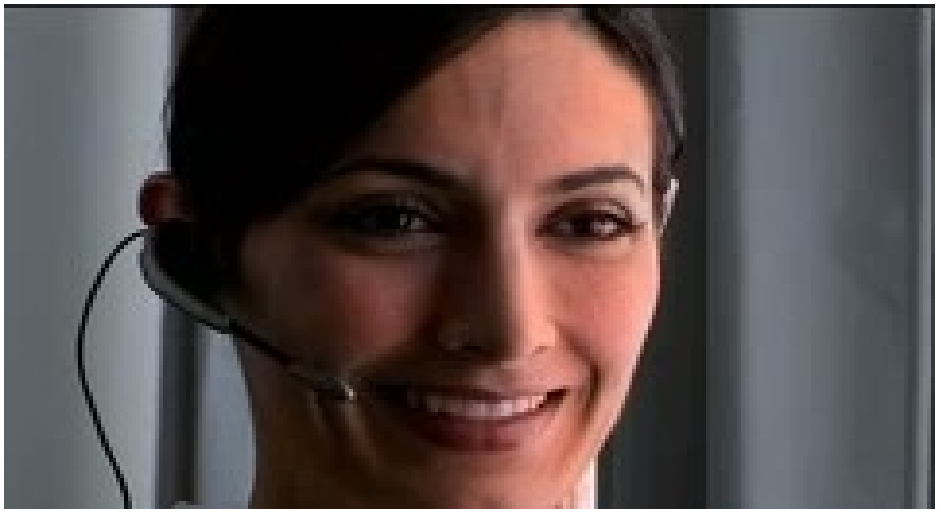
Experimente

- Videokolorierung

Experimente

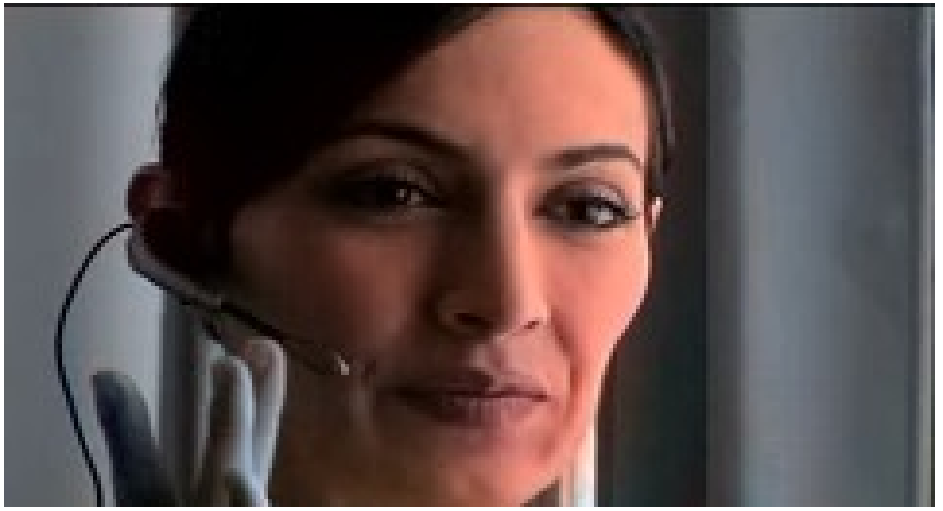
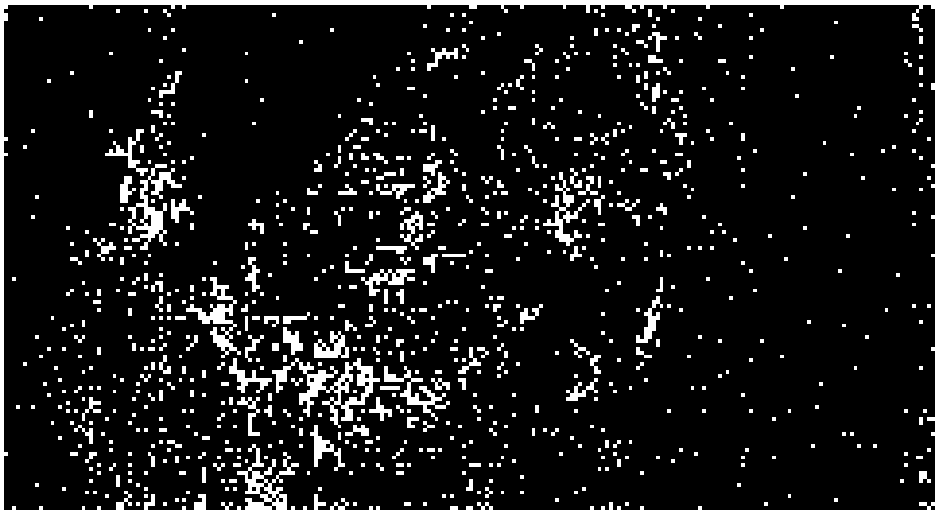
- Videokompression



Experimente

- Videokompression


Experimente

- Kompressionsrate 0.899
  - Berechnung allerdings unrealistisch, realistischer sind eher 0.925

Fazit

- optisch ansprechende kolorierte Bilder und Videos
- Kompression mit guten Kompressionsraten als Aufsatz auf herkömmliche Verfahren
- Videokompression streaming-fähig
- mögliche Verbesserung:
  - „Vergessen“ von Labels ==> selber PSNR bei niedrigeren Kosten
  - Beweis von performance boundaries

Bemerkungen

- weitere Verbesserungsmöglichkeiten:
  - evtl. Verwendung von spezialisierten SW-Kompressionsverfahren
- nach welchen Kriterien wurden die Bilder und Videos der Experimente ausgewählt?
  - „non-stationary video sequences“
- evtl. muss α auch gespeichert werden

Vielen Dank für Ihre Aufmerksamkeit!

Quellen

- sämtliche Bilder entstammen der Seite http://sml.nicta.com.au/~licheng/LearnCompressImgVid/LearnCompressImgVid.html oder dem Artikel „Learning to Compress Images and Videos“, welcher auch auf dieser Seite zu finden ist.