Photo Editing

Seminar Computational Photography

Visual Computing
Department of Computer Science

Photo Editing - Themen

- Digitale Fotomontage
- Structural Image Editing

INTERAKTIVE DIGITALE FOTOMONTAGE

Digitale Photomontage

- Digitale Fotomontage: Kombination von Teilen einer Reihe von Fotos zu einem zusammengesetzten Bild
- Grund:
  - man kann selten in Foto festhalten, was man mit den Augen sieht
  - künstlerischer Bedarf

Voraussetzungen

- Satz von Quellbildern (Image-Stack)

- Bilder sollten aufeinander bezogen sein
  - gleiche Szene, aber unterschiedliche Beleuchtung oder Kamera-Positionen
  - wandelnde Szene, aber feste Kameraposition

Interaktive Digitale Photomontage

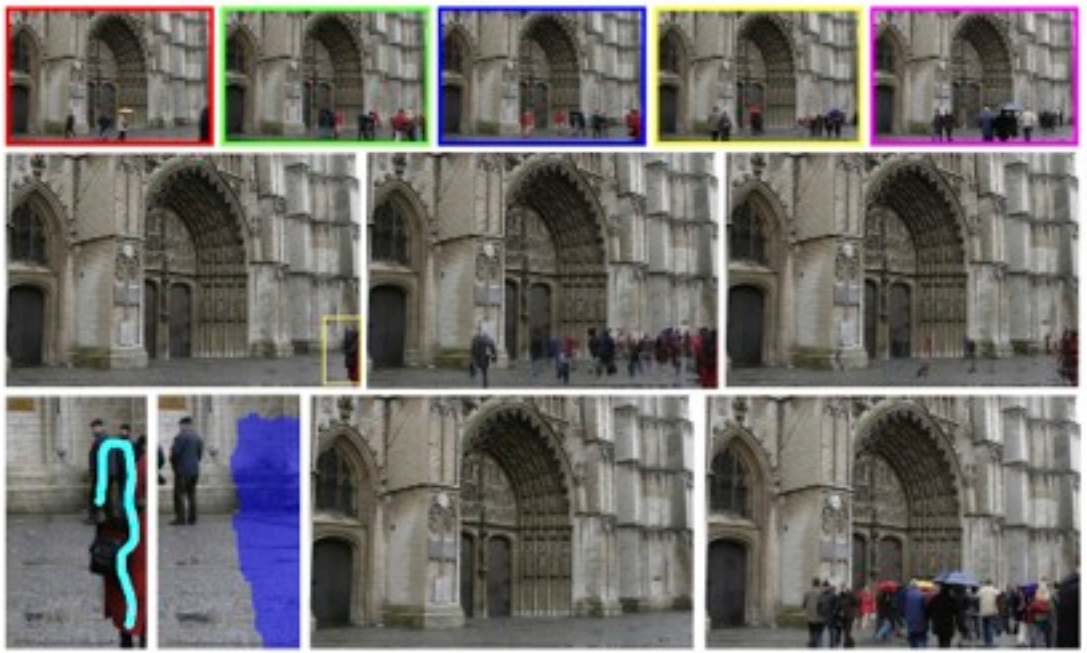
- Aufgabe: Schnitte suchen, an denen man Bildbereiche ausschneiden und wieder neu zusammenfügen kann mit möglichst geringen sichtbaren Nähten
- Zwei Techniken: Graph-Cut-Optimierung für nahtlose Kombinierung, Gradient-Domain-Fusion (Prozess basierend auf Poisson-Gleichungen) zur weiteren Verringerung von Artefakten

Zielfunktionen

- Bild-Zielfunktionen:
  - nach Laden von Stack kann Benutzer eine Bild-Zielfunktion wählen, die global oder lokal angewendet wird
  - wird unabhängig für jede Pixelposition p bestimmt, basierend auf Menge von Pixelwerten aus der gleichen Position p in jedem der Quellbilder (=span)
- Seam-Zielfunktionen
  - erfasst die Eignung eines Seams zwischen zwei Bildbereichen
  - immer global

Bild-Zielfunktionen

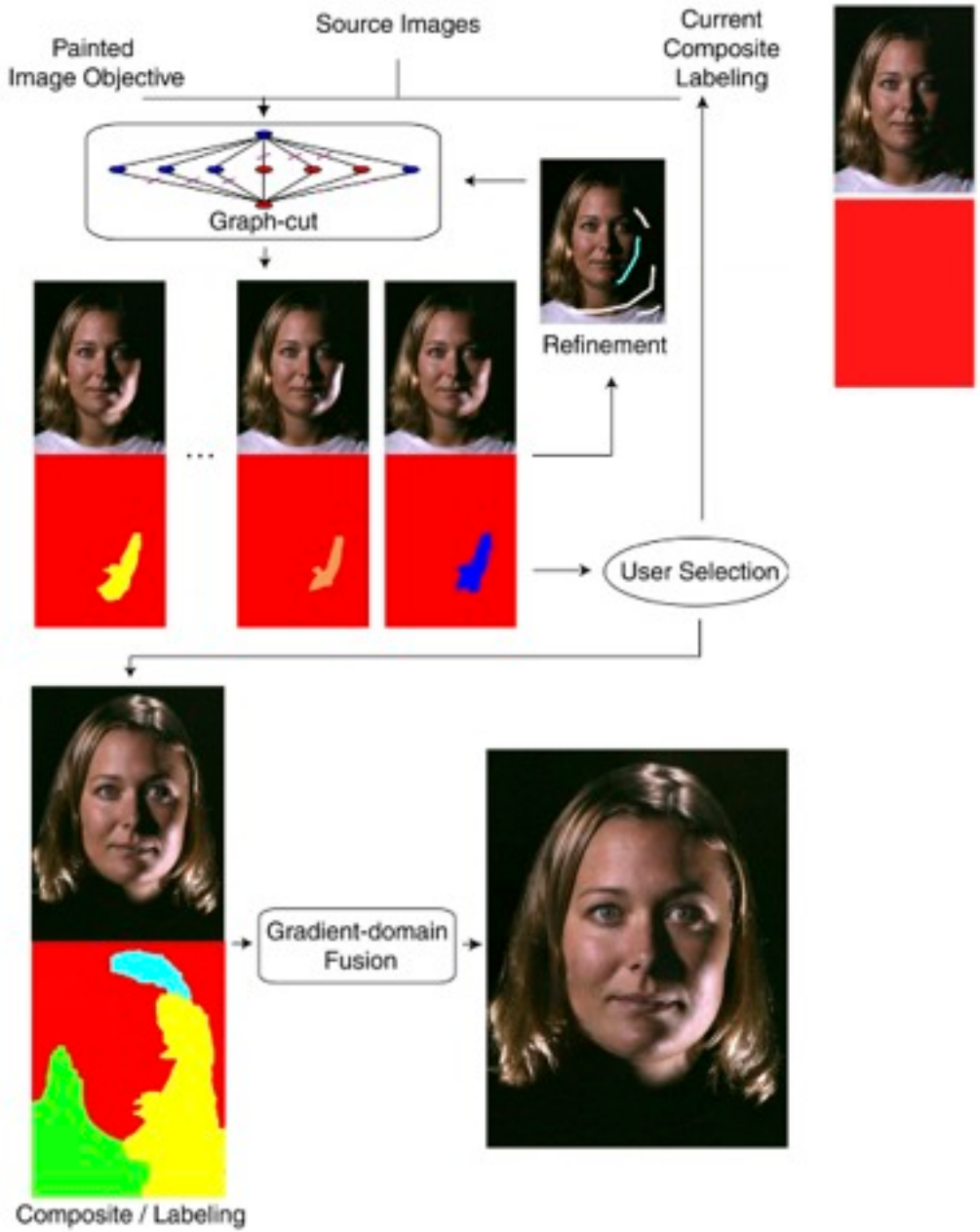
- Designated Color (Gewünschte Farbe)
- Min/Max Luminance (Helligkeit)
- Min/Max Contrast (Kontrast)
- Min/Max Likelihood (Ähnlichkeit)
- Eraser
- Min/Max Difference (Farbunterschied)
- Designated Image (Gewünschtes Bild)



Seam-Zielfunktionen

- Colors: Angleichung von Farben an Seams
- Colors & Gradients: Angleichung von Farben und Gradienten an Seams
- Colors & Edges: Angleichung von Farben an Seams, bevorzugt Seams an Kanten

Ablauf


Source Images
Painted
Image Objective
Current
Composite
Labeling
Graph-cut
Refinement
User Selection
Gradient-domain
Fusion
Composite / Labeling

Graph-Cut Optimierung (1)

- n Quellbilder $S_1,...,S_n$
- um eine Zusammensetzung zu formen müssen wir ein Quellbild S_i für jeden Pixel p wählen
- mapping zwischen Pixeln und Quellbild: „labeling“
- Label für jeden Pixel: L(p)
- ein Seam zwischen zwei benachbarten Pixeln in der Zusammensetzung existiert, wenn $L(p) \neq L(q)$

Graph-Cut Optimierung (2)

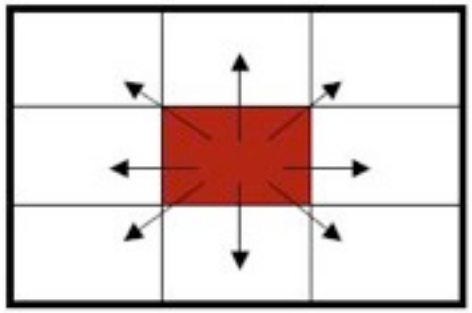
- Definition einer Kostenfunktion C eines Labels L als Summe zweier Terme: Data Penalty C_d über alle Punkte p und Interaction Penalty C_i über alle Paare von benachbarten Pixeln p, q:

$$C(L) = \sum_{p} C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q))$$

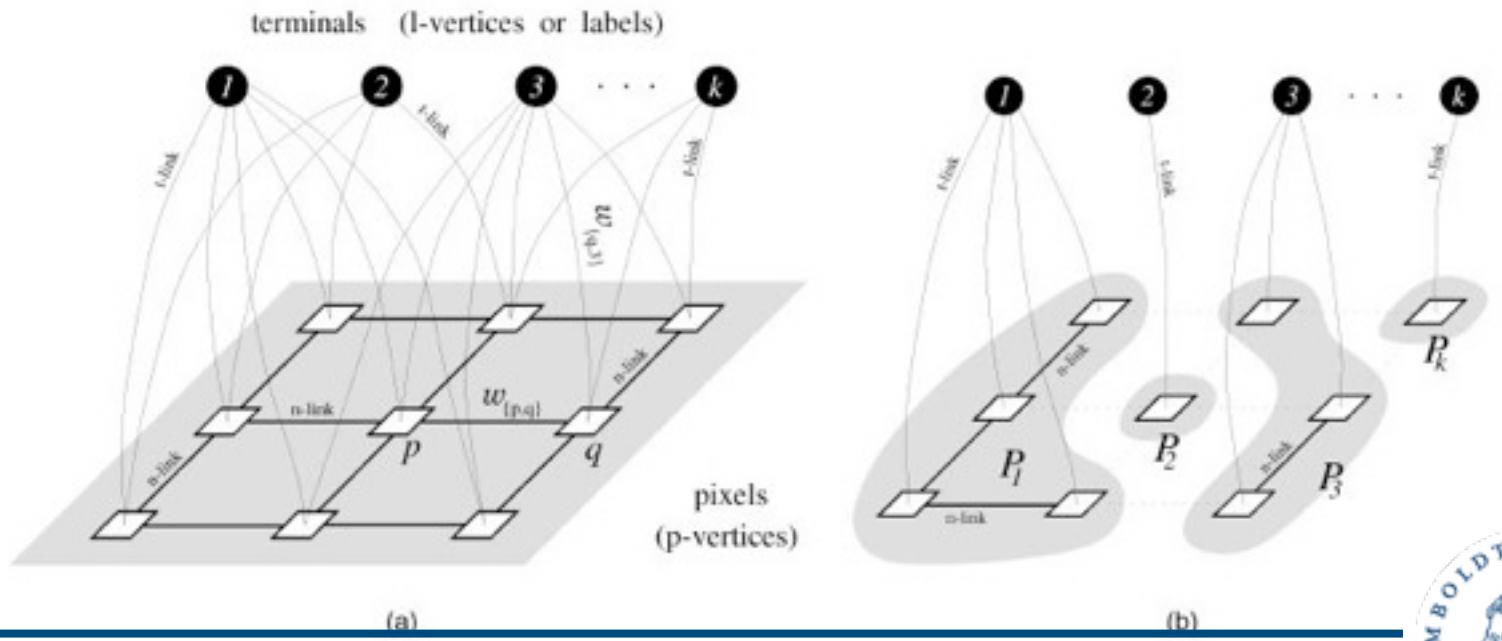
- Data Penalty wird definiert durch den Abstand zur Bild-Zielfunktion (vom Benutzer ausgewählt)
- Interaction Penalty definiert durch Abstand zur Seam-Zielfunktion

Graph-Cut Optimierung (3)

- Bsp. Data Penalty für max. Kontrast: berechnet durch die Summe der Gradienten in alle 8 Richtungen des Pixels

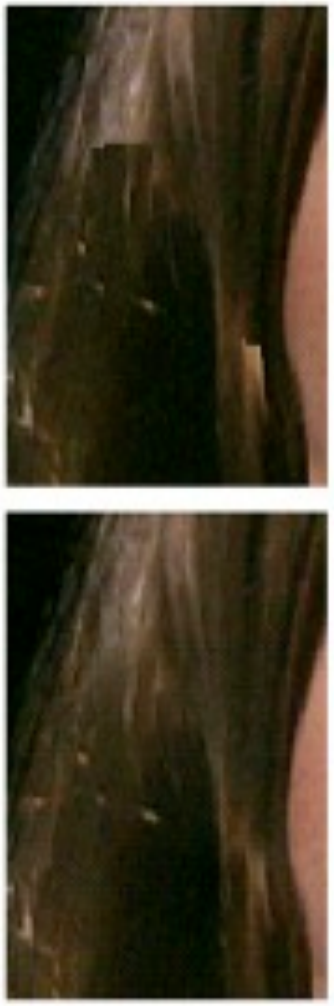


BOYKOV ET AL.: FAST APPROXIMATE ENERGY MINIMIZATION VIA GRAPH CUTS 1231



Gradient-Domain Fusion (1)

- für viele Anwendungen sind Quellbilder zu unterschiedlich, um durch Graph-Cut alleine eine nahtlose Zusammensetzung zu erhalten
- wenn Graph-Cut Algorithmus keinen idealen Seam findet, könnten noch Artefakte existieren
- sinnvoll, die Input-Bilder als Quellen von Farbgradienten statt als Quellen der Farbe zu sehen



Gradient-Domain Fusion (2)

- gleiches Labeling wie bei Graph-Cut, aber Farbgradienten, statt Farben -> zusammengesetztes Vektorfeld entsteht
- Berechnung einer Farb-Zusammensetzung, deren Gradienten am besten dem Vektorfeld entsprechen
- dies erlaubt uns Farbunterschiede zwischen nebeneinander gestellten Bildregionen zu beseitigen

Gradient-Domain Fusion (3)

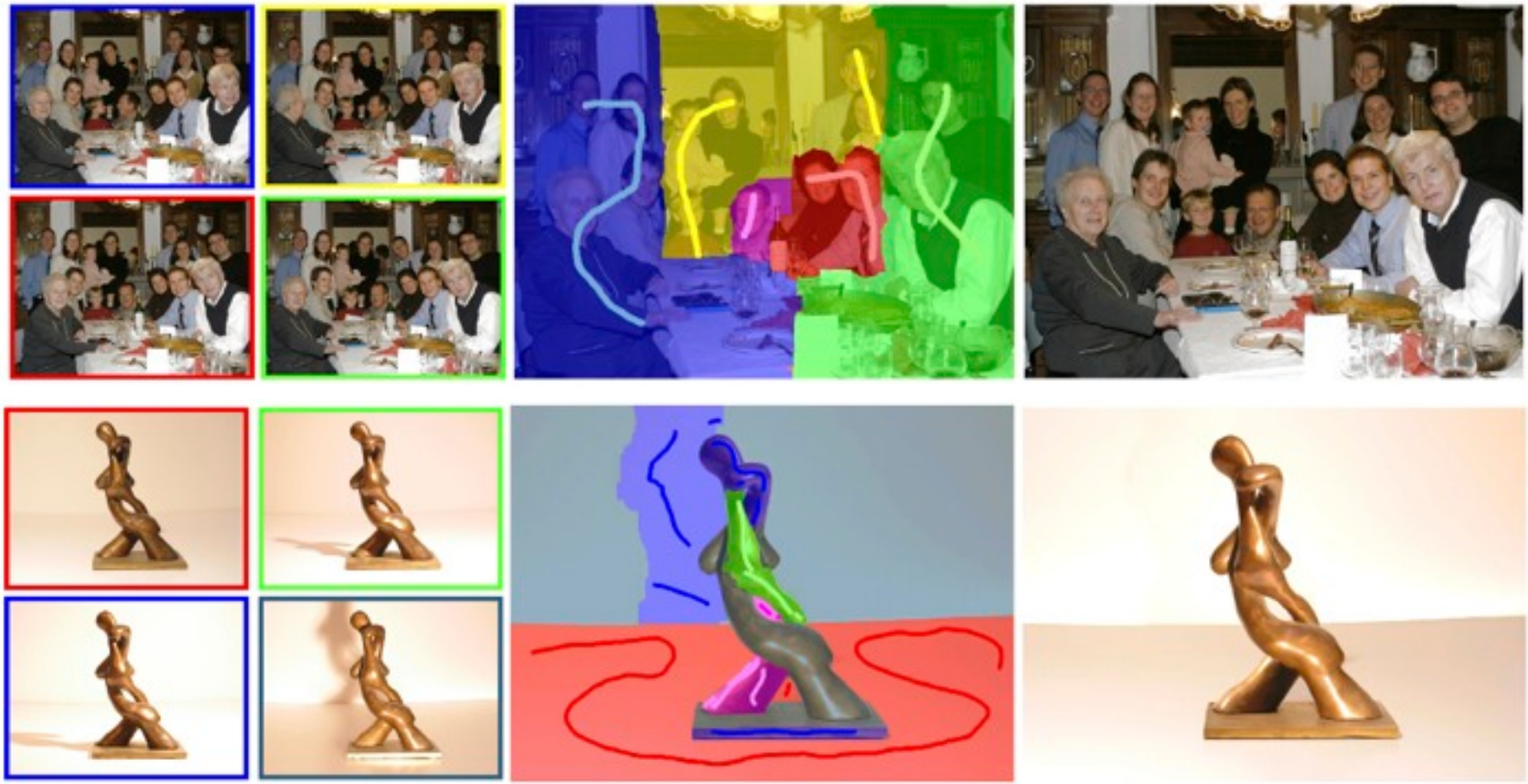
- Farb-Zusammensetzung wird separat in den drei Farbkanälen berechnet
- Pro Farbkanal zwei lineare Gleichungssysteme:

$$v_{x+1,y} - v_{x,y} = \nabla I_x(x,y)$$
$$v_{x,y+1} - v_{x,y} = \nabla I_y(x,y)$$

- es existiert kein Bild, dessen Gradient genau zum Input passt
- Stattdessen kann ein am besten passendes Bild in einem Least-Squares-Sinn durch Lösung einer Diskretisierung der Poisson-Gleichung berechnet

Anwendungsgebiete (1)

- Punktuelle Zusammensetzung
- Bildbasierte Belichtung



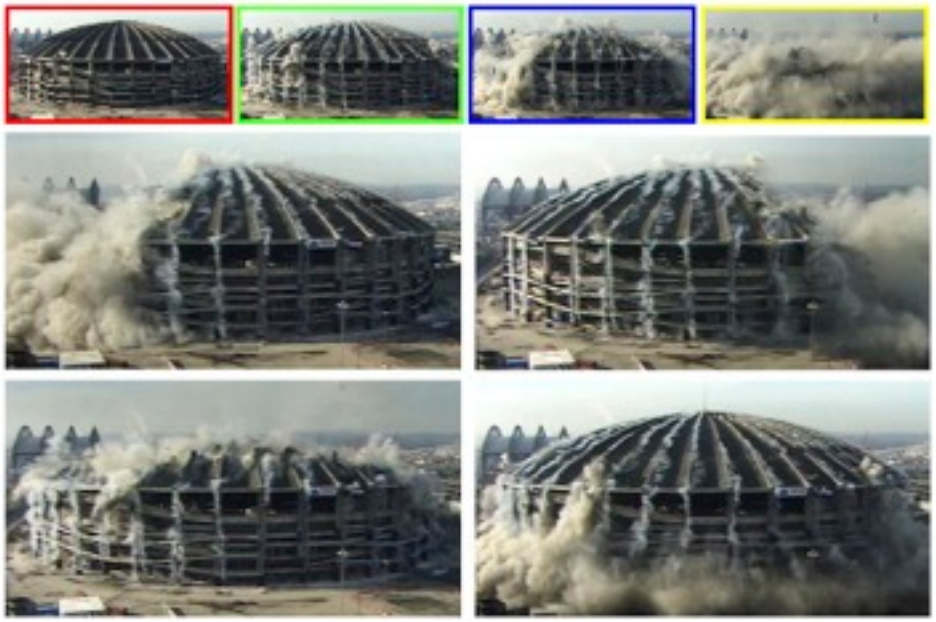
Anwendungsgebiete (2)

- Erweiterte Tiefenschärfe

Anwendungsgebiete (3)

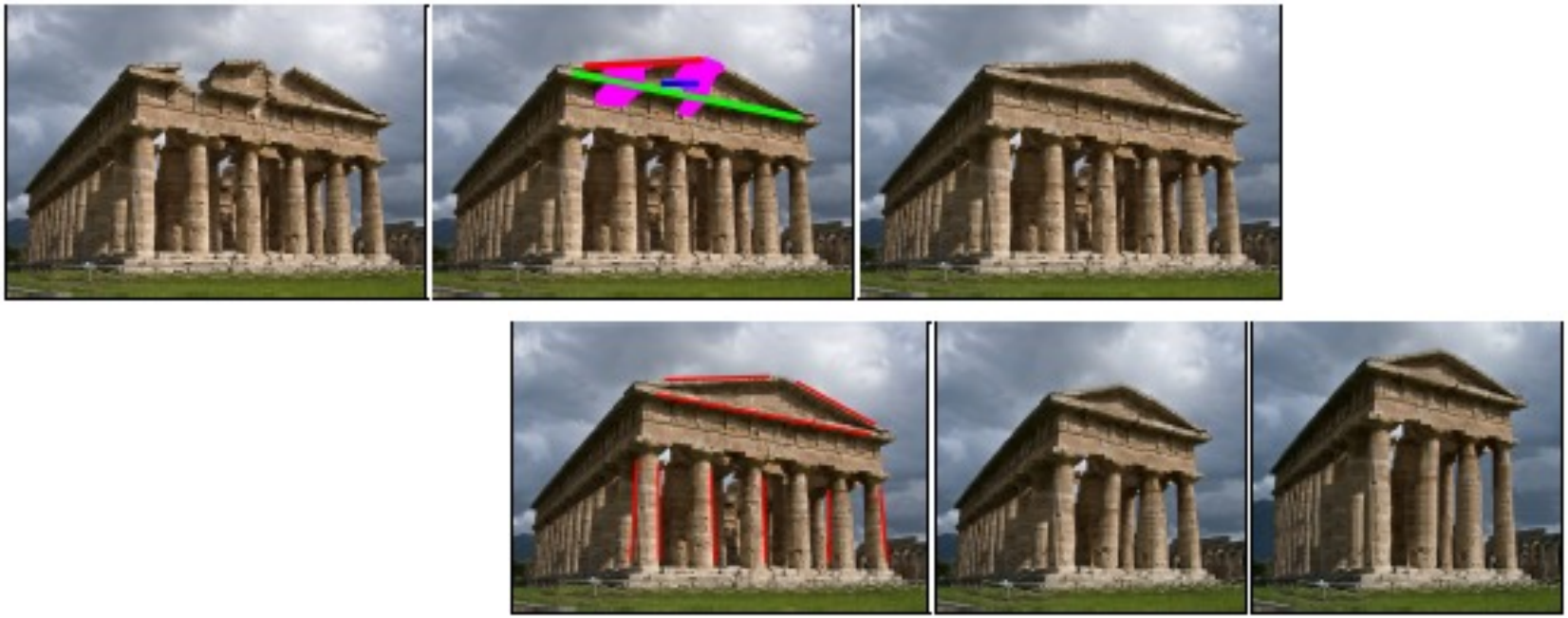
- Bild-Mosaik

- Zeitraffer-Mosaik



Anwendungsgebiete (4)

- Hintergrundrekonstruktion
- Visualisierte Bewegung



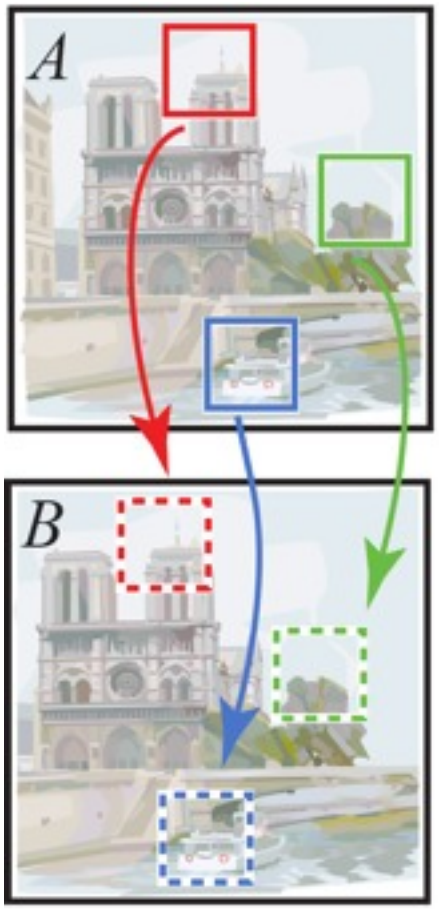
STRUCTURAL IMAGE EDITING

Motivation

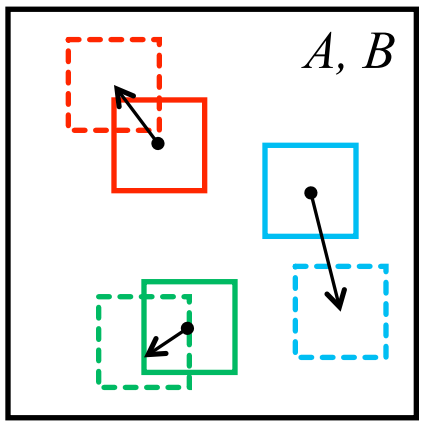
- PatchMatch-Framework
  - Bildmanipulation auf Objektebene
  - Erhaltung der grundsätzlichen Objektstruktur
  - Benutzer markiert Struktur-erhaltende Elemente, oder markiert Löcher, welche ausgefüllt werden sollen
  - Interaktiv/Echtzeitfähig

Approximate Nearest-Neighbour Algorithm

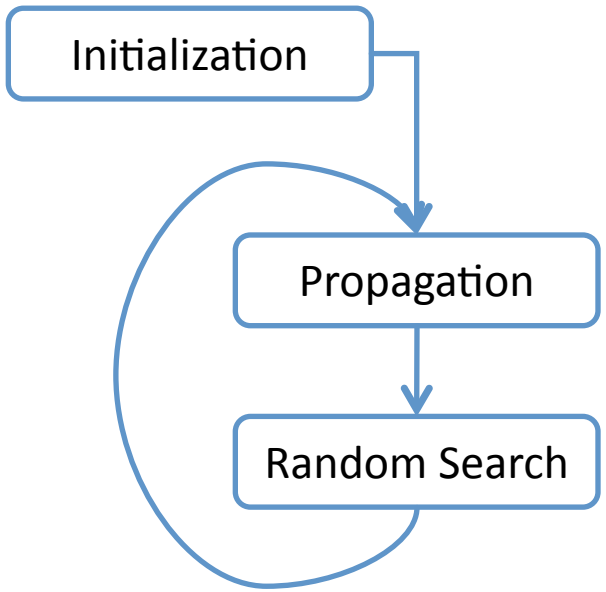
- Berechnung eines NNF (*Nearest-Neighbor Field*)



$$f : A \mapsto \mathbb{R}^2$$

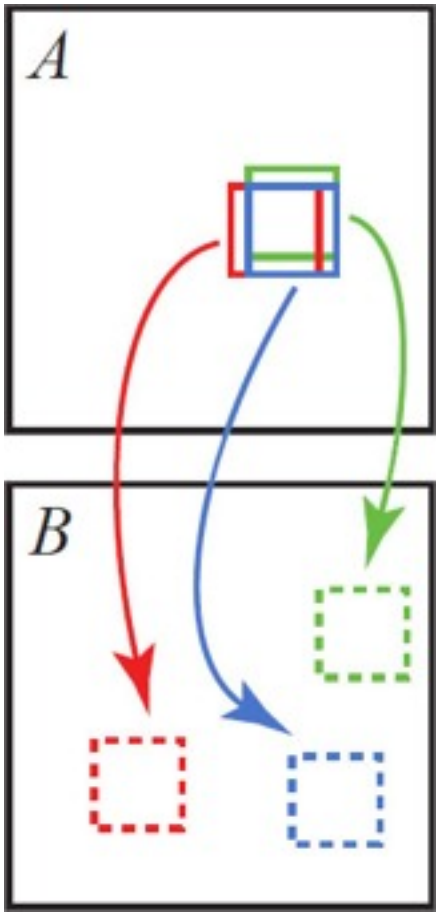


Patch-Abstände



Algorithmus-Phase: Initialisierung

- Matches werden zufällig zugeordnet

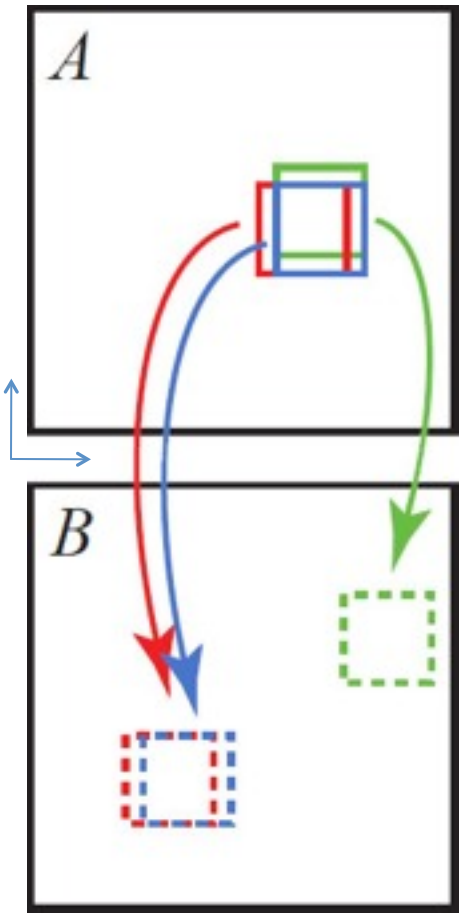


Algorithmus-Phase: Propagation

- Verbesserung der Matches aufgrund vorhandener Matches
- Annahme: Benachbarte Patches haben mit hoher Wahrscheinlichkeit auch „benachbartes“ Matching
  - Für $f(x, y)$ wähle bestes Matching aus

$$\{f(x, y), f(x-1, y), f(x, y-1)\}$$

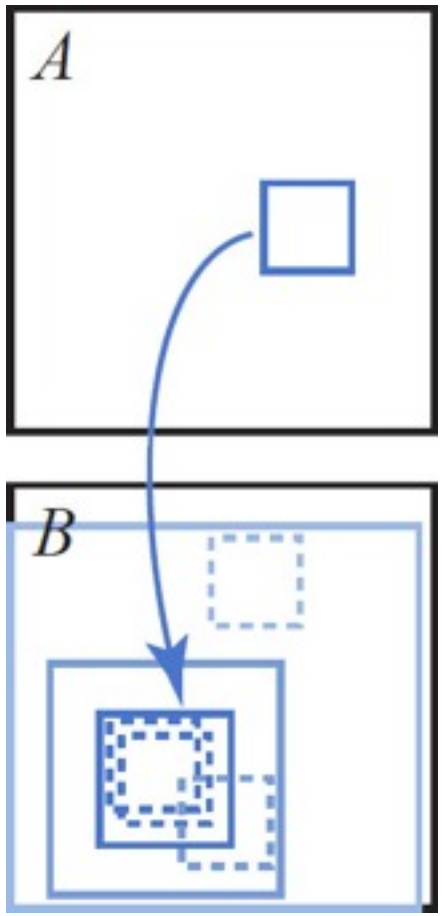
  - Bestes Matching = kleinster Patch-Abstand
  - Jede zweite Iteration: Umkehrung der Reihenfolge



Algorithmus-Phase: Search

- Suche nach besseren, zufällig ausgewählten Matches
- Finden von besseren Matches in einer Menge von Patches:

$$\mathbf{v}_0 = f(x, y)$$
$$\mathbf{u}_i = \mathbf{v}_0 + w\alpha^i \mathbf{R}_i$$
$$i = 0, 1, 2, \ldots \text{ bis } w\alpha^i < 1$$



R_i: Richtung [-1,1]x[-1,1]

α: Verhältnis zwischen Suchfenster Größen

w: Max. such „Radius“

Werkzeuge

- Benutzergesteuerte Ermöglichung von
  - Inpainting
  - Retargeting
  - Reshuffling
  - ...
- Voraussetzung:
  - Bidirektionales Distanzmaß (zwischen Quell- und Zielbild), aufbauend auf dem NNF
  - Darauf dann Anwendung von Constraints

Bidirectional Similarity Synthesis Approach

- Werkzeuge basieren auf „Bidirektionalem Distanzmaß“ zwischen Quellbild S und Zielbild T

$$d_{BDS}(S,T) = \overbrace{\frac{1}{N_S}\sum_{s \subset S}\min_{t \subset T} D(s,t)}^{d_{\text{complete}}(S,T)} + \overbrace{\frac{1}{N_T}\sum_{t \subset T}\min_{s \subset S} D(t,s)}^{d_{\text{coherence}}(S,T)}$$

D → SSD („Summe der quadrierten Abweichungen“)

$d_{complete}$ Möglichst viele Informationen des Quellbilds sollen im Zielbild vorhanden sein

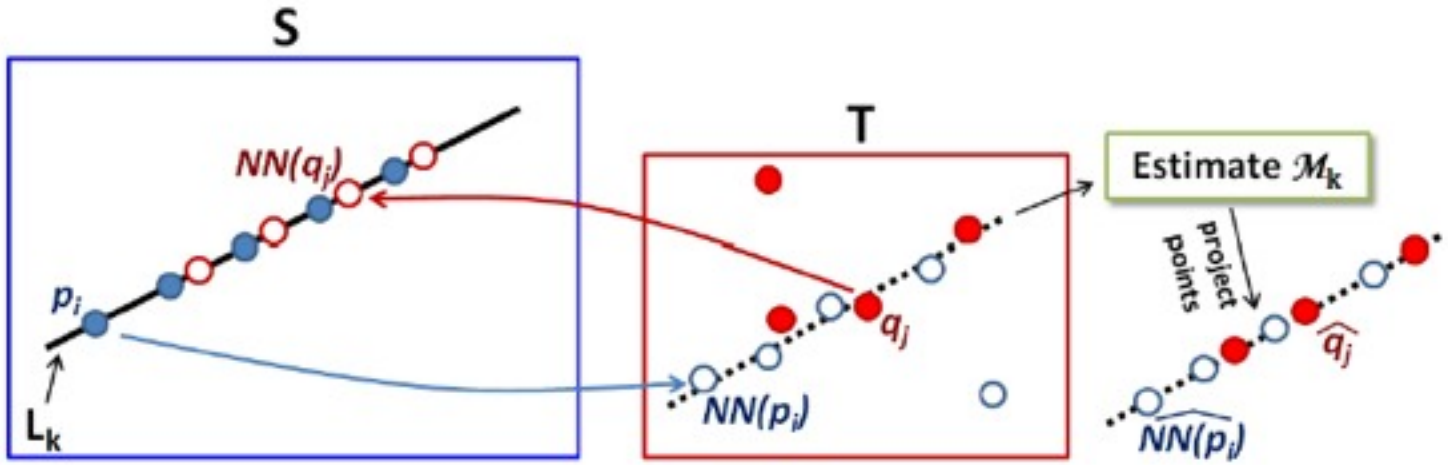
$d_{coherence}$ Vermeidung von Artefakten (möglichst wenig Informationen, die nicht im Quellbild waren)

Constraints

- Beeinflussung des Ergebnisse durch Markierung von Flächen oder Linien
  - Ohne Constraints mangelhafte Erhaltung von Strukturen/Objekten
  - Steuerung der Werkzeuge
- Schwerpunkt liegt auf Erhaltung von geraden Linien
- Verschiedene Farben, für verschiedene Strukturen

Constraints

- Menge möglicher Nearest-Neighbors in T wird für bestimmte Teilmenge von S beschränkt



NN(p_i): Nearest-Neighbor von p_i in T (p_i ist Punkt auf L_k)

- Optimierung der d_{BDS} je nach gewünschtem Werkzeug (M_k)

$$arg\ min\ d_{BDS}\ \ M_k(p_i, NN(p_i), q_j, NN(q_j)) = 0$$

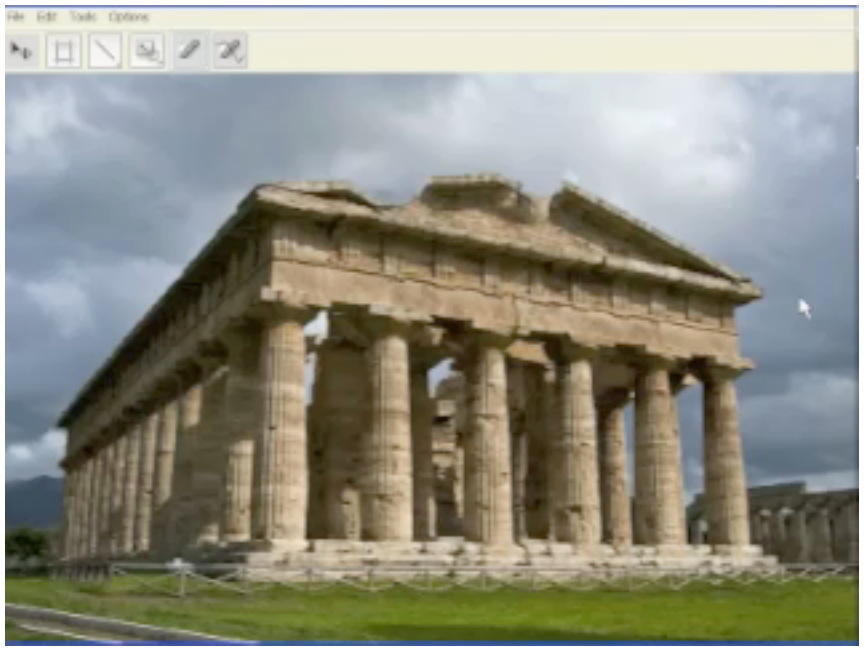
  - Für Linien: $NN(p_i)^T\, l = 0$
  - Für Regionen: $H_k p_i - NN(p_i) = 0$

Werkzeuge: Inpainting

- Entfernung oder Vervollständigung von Objekten
- Mit d_{BDS} ohne *completeness term*
  - **T**: Zu füllender Bereich
  - **S**: Restliches Bild

Werkzeuge: Inpainting

- Entfernung oder Vervollständigung von Objekten
- Mit d_{BDS} ohne *completeness term*
  - **T**: Zu füllender Bereich
  - **S**: Restliches Bild



Werkzeuge: Retargeting

- Unveränderte d_{BDS}
- „Importance masks“ für nicht zu verändernde Regionen
- Model für $M_{k:}$
  - „Free lines“ → Linien fest, Translationen und Neigungen frei

Werkzeuge: Retargeting

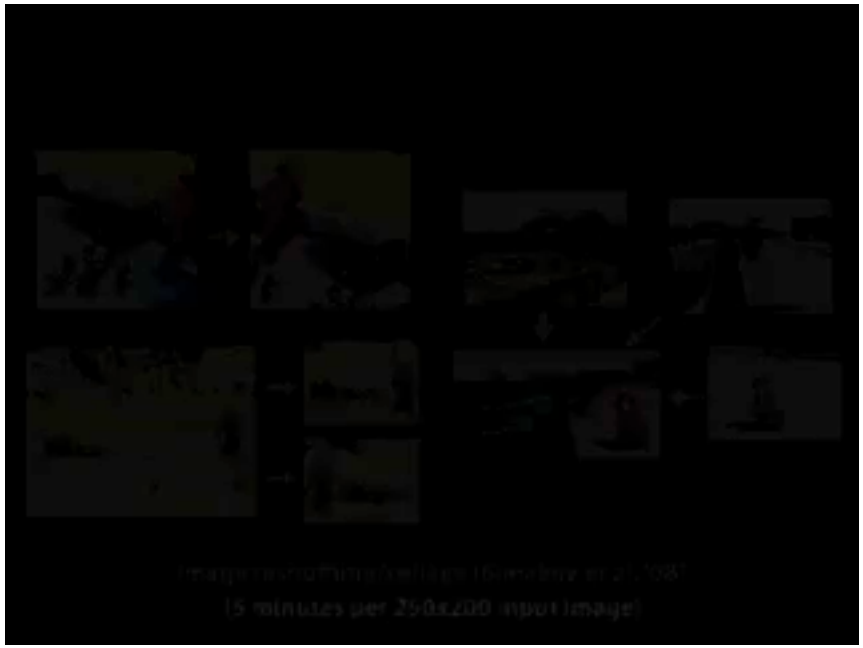
- Model für $M_{k:}$
  - „Free-slope lines“ → Translationen frei, Neigungen entsprechen Input

Werkzeuge: Reshuffling

- d_{BDS} unverändert
- Objekte werden verschoben und Bild rekonstruiert
- NNF wird für bestimmte Regionen festgesetzt
  - > „Hard Constraints“

Werkzeuge: Reshuffling

- d_{BDS} unverändert
- Objekte werden verschoben und Bild rekonstruiert
- NNF wird für bestimmte Regionen festgesetzt
  - > „Hard Constraints“



Structural Image Editing

Fragen?

Quellen

- Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen. Interactive Digital Photomontage. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004), 2004.
- Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. *PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing.* ACM Transactions on Graphics (Proc. SIGGRAPH). 28(3) August 2009.
- Simakov, D., Caspi, Y., Shechtman, E., and Irani, M.. *Summariting visual data using bidirectional similarity*. In CVPR, 2008