

Patch Match

A Randomized Correspondence Algorithm for
Structural Image Editing

Einleitung (1)

- Interaktives Bildbearbeitungstool
- Bisherige Tools entweder flexibel oder schnell
- Diese Arbeit erreicht beides

- Beispiel:



(a) original

(b) hole+constraints

(c) hole filled



(d) constraints

(e) constrained retarget

(f) reshuffle

Einleitung (2)

- Vorteile bei Geschwindigkeit:
 - Intuitive Benutzung
 - Versuch und Irrtum
- Nachbarschaftssuche verbessert durch:
 - Dimension des Verschiebungsraums
 - Bisher durch KD-Trees und Dimensionsreduktion
 - Natürliche Bildstruktur
 - Nutzung durch Betrachtung der Umgebung
 - Gesetz der großen Zahlen
 - Zufallssuche mit vielen Stichproben → große Wahrscheinlichkeit für gutes Ergebnis

Einleitung (3)

- Beschleunigung um 20-100 fache im Vgl. zu KD-Tree mit Dimensionsreduktion
- Implementierung mit GPU 7 fach schneller als CPU

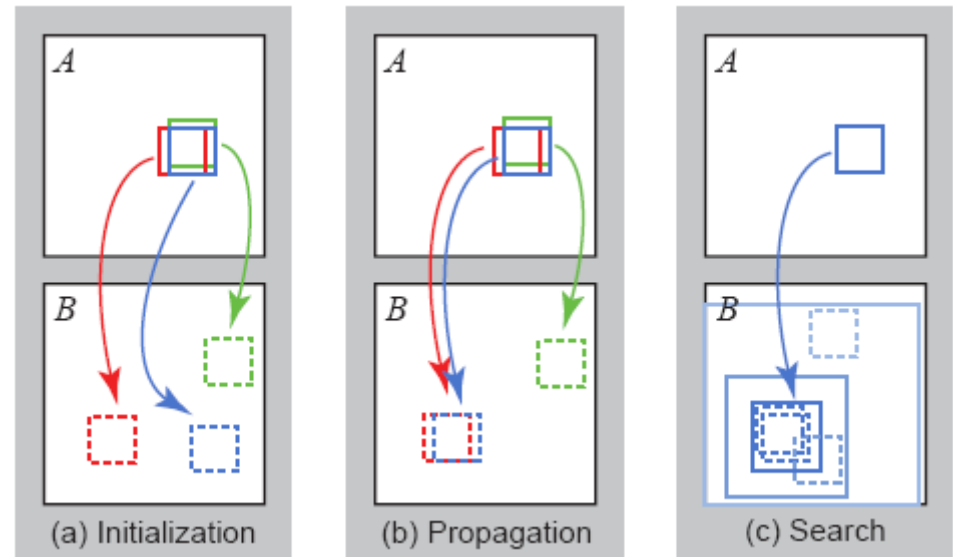
Ähnliche Arbeiten im Umfeld

- Texturen-Darstellung und -Vervollständigung
- Methoden zur Nachbarschaftssuche
- Intuitive und interaktive Bearbeitung
- Inhaltssensible Bildskalierung
- Neuordnung von Bildinhalten

- Einschränkung: kleine Bilder

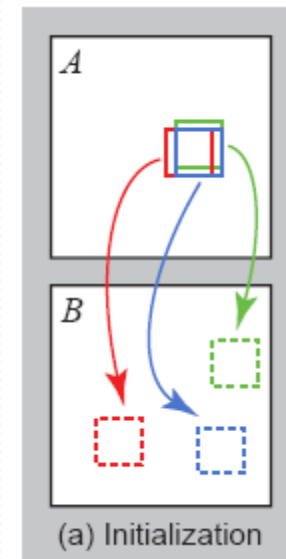
Nachbarschaftssuche

- = Kern der Arbeit
- Prozess besteht aus folgenden Schritten:
 - Initialisierung
 - Iteration
 - Abstände ermitteln
 - Ausbreitung
 - Suche



Initialisierung

- Zuweisung von Zufallsbereichen
- Schrittweise Verfeinerung der Bereiche
- Vermeidung von lokalen Minima

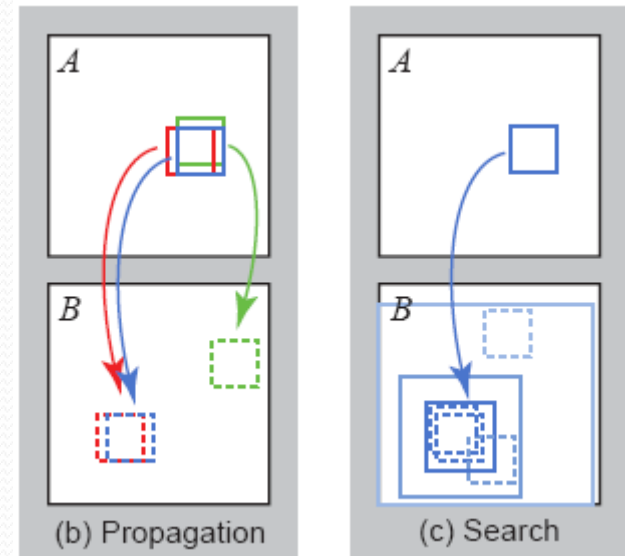


Ausbreitung

- Stelle soll korrigiert werden
- Betrachtung der Umgebung von Übereinstimmungen

Zufallssuche

- Verbesserung der Zufallssuche durch Verkleinerung des Radius



Abbruchkriterium

- Normalerweise verschiedene Abbruchkriterien für verschiedene Probleme
- Konvergenz in ca. fünf Iterationen

Effizienz

- Verbessern durch Einschränken des Radius der zu untersuchenden Umgebung
- Speichern redundanter Werten

Vergleich der Nachbarschaftssuche (1)

- Vergleich bei verschiedenen Algorithmen, Bilder

Megapixels	Time [s]		Memory [MB]	
	Ours	<i>kd-tree</i>	Ours	<i>kd-tree</i>
0.1	0.68	15.2	1.7	33.9
0.2	1.54	37.2	3.4	68.9
0.35	2.65	87.7	5.6	118.3

Table 1: *Running time and memory comparison for the input shown in Figure 3. We compare our algorithm against a method commonly used for patch-based search: kd-tree with approximate nearest neighbor matching. Our algorithm uses $n = 5$ iterations. The parameters for kd-tree have been adjusted to provide equal mean error to our algorithm.*

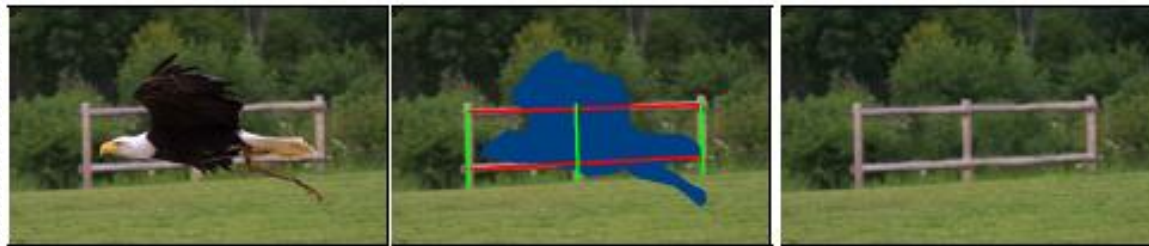
Vergleich der Nachbarschaftssuche (2)

- Im Schnitt 20x bis 100x schneller
- 20x weniger Speicherbedarf
- Größere Bilder möglich (2MP)

Bearbeitungsmöglichkeiten

- Bildrekonstruktion in fehlenden Bereichen
- Neuordnung einzelner Bildobjekte
- Inhaltssensible Bildskalierung und Deformation
- Erhaltung von Strukturen durch Einschränkungen
- Kopieren von Objekten

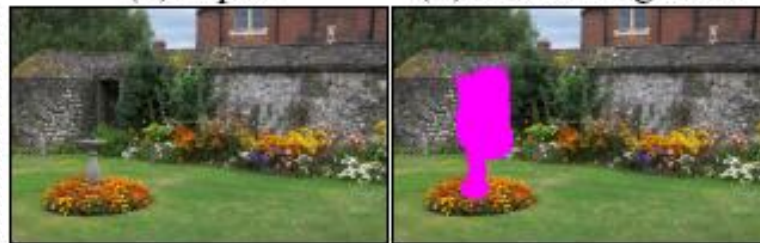
Bildrekonstruktion in fehlenden Bereichen



(a) input

(b) hole and guides

(c) completion result

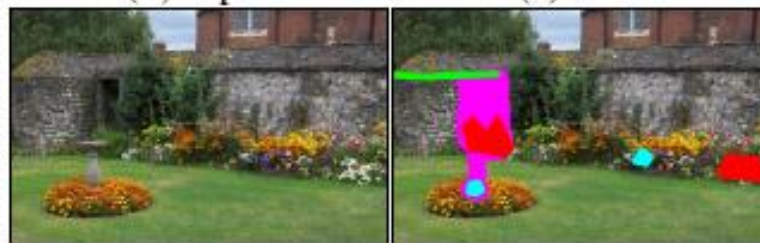


(d) input

(e) hole



(f) completion (close up)



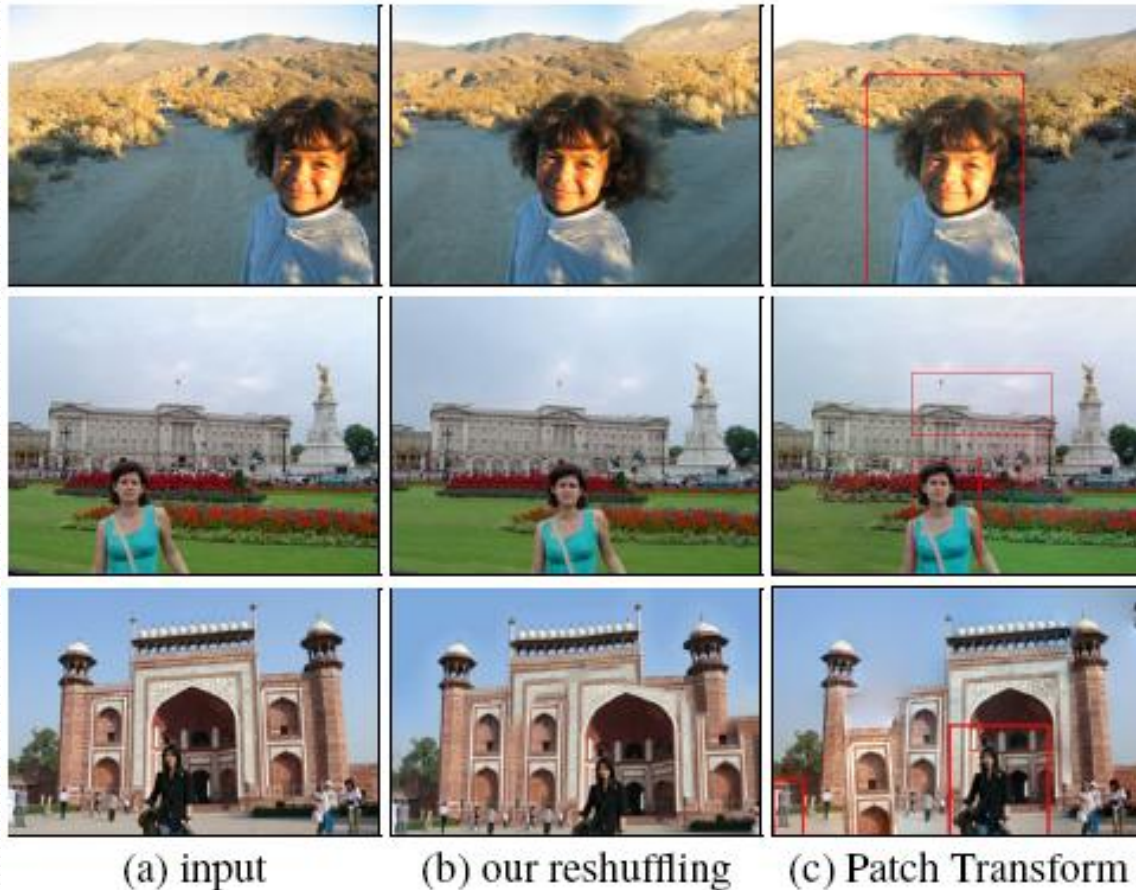
(g) same input

(h) hole and guides



(i) guided (close up)

Neuanordnung einzelner Bildobjekte



Inhaltssensible Bildskalierung und Deformation



(a) building marked by user



(b) scaled up, preserving texture

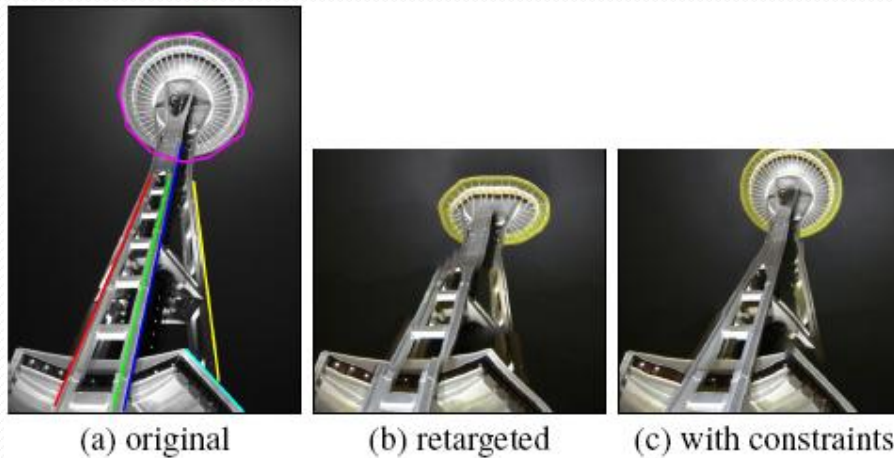
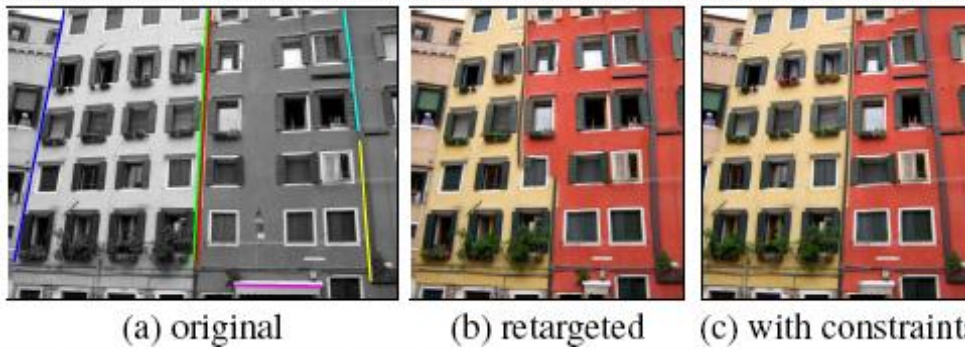


(c) bush marked by user



(d) scaled up, preserving texture.

Erhaltung von Strukturen durch Einschränkungen



Vergleich zu anderen Tools zur Bildmanipulation



Figure 10: Retargeting. From left: (a) Input image, (b) [Rubinstein et al. 2008], (c) [Wang et al. 2008], (d) Our constraints, (e) Our result.

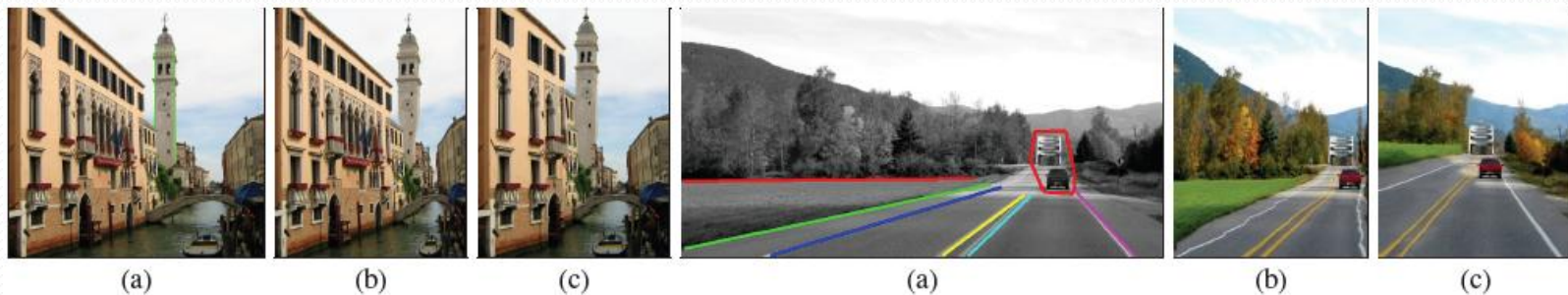


Figure 11: Retargeting: (a) Input image, annotated with constraints, (b) [Rubinstein et al. 2008], (c) Our output.

Nachteile und Vorteile

- Algorithmus liefert in wenigen Fällen falsche Ergebnisse
- Extreme Veränderungen stören Konvergenz Algorithmus
- Ausgleich durch Geschwindigkeit, Versuch mit anderen Randbedingungen oder neuer Anfangswert für die Zufallssuche
- Vergleich mit bisherigen Anwendungen:
 - Schneller oder mehr Bearbeitungsmöglichkeiten
- Durch Lerneffekt intuitive Handhabung

Ergebnis

- Neuer schneller Algorithmus zur Nachbarberechnung für Bildpaare oder Bildregionen
- Interaktive Manipulation in Echtzeit mit intuitiver Steuerung
- Viele verschiedene Bearbeitungsmöglichkeiten

Ausblick

- Liste der Nachbarn für jeden Pixel
 - keine Berechnung
 - schneller aber mehr Speicherbedarf
- Implementierung des Algorithmus auf der GPU
- Umsetzung des Algorithmus für Echtzeit-Video-Bearbeitung