

Programmierung 2 - SS16

Projekt 3 - C - Seamcarving

Autoren: Rebecca Eifler, Merlin Köhler, Kallistos Weis

25 Mai 2016

Universität des Saarlandes

1. Motivation
2. Algorithmus
3. Bildformat
4. Einführung in das Programm

Git Projekt-Repository

Wir können das Projekt mit `git clone` unter folgender URL beziehen:

```
https://prog2scm.cdl.uni-saarland.de/git/project3/<NAME>
```

<NAME> = Euer Benutzername auf der Prog2-Website

Achtung!

Die Repositories sind nur innerhalb des Uninetzes erreichbar. Von außerhalb kann man eine VPN-Verbindung zum Uninetz einrichten.

Eine Anleitung steht auf der Website unter [Software](#).

Motivation



Der Broadway Tower ist ein weitgehend zweckfreies Gebäude (Folly) nahe dem englischen Dorf Broadway in Worcestershire. Der über 16 Meter hohe Turm steht am Cotswold Way und auf einem der höchsten Punkte der Cotswolds (312 m über dem Meeresspiegel). Bei guter Sicht können vom Turm aus 13 Grafschaften überblickt werden.

(Quelle: https://de.wikipedia.org/wiki/Broadway_Tower)



Der Broadway Tower ist ein weitgehend zweckfreies Gebäude (Folly) nahe dem englischen Dorf Broadway in Worcestershire. Der über 16 Meter hohe Turm steht am Cotswold Way und auf einem der höchsten Punkte der Cotswolds (312 m über dem Meeresspiegel). Bei guter Sicht können vom Turm aus 13 Grafschaften überblickt werden.

(Quelle: https://de.wikipedia.org/wiki/Broadway_Tower)



Der Broadway Tower ist ein weitgehend zweckfreies Gebäude (Folly) nahe dem englischen Dorf Broadway in Worcestershire. Der über 16 Meter hohe Turm steht am Cotswold Way und auf einem der höchsten Punkte der Cotswolds (312 m über dem Meeresspiegel). Bei guter Sicht können vom Turm aus 13 Grafschaften überblickt werden.

(Quelle: https://de.wikipedia.org/wiki/Broadway_Tower)

Motivation



(a) Originalbild.



(b) Verkleinert durch seamcarving.

- Zuschneiden von Bildern
 - Erhaltung von „wichtigen“ Merkmalen
→ Kanten
 - Erhaltung von Proportionen
- ⇒ Geringe wahrnehmbare Verzerrungen

Algorithmus

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Lokale Energie (Beispiel)

- Pixelwert $p_{x,y}$ an der Stelle (x,y)
- Graustufenbild: Werte $p_{x,y} \in [0; 5]$ (üblicherweise $p_{x,y} \in [0; 255]$)

5	5	3	3	3
3	3	3	3	0
3	3	3	0	1
3	3	0	1	1
3	3	0	1	1

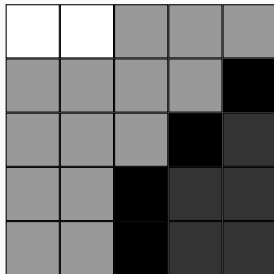
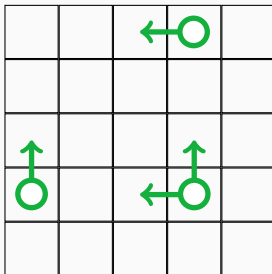


Abbildung 2: Originalbild.

Lokale Energie (Beispiel)

Berechnung lokale Energie:

- Summe der quadrierten Differenz zum oberen und linken Nachbarn (falls vorhanden)



Lokale Energie (Beispiel)

Berechnung lokale Energie:

- Summe der quadrierten Differenz zum oberen und linken Nachbarn (falls vorhanden)

5	5	3	3	3
3	3	3	3	0
3	3	3	0	1
3	3	0	1	1
3	3	0	1	1

(a) Originalbild.

lokale
Energie →

(b) Lokale Energie.

Lokale Energie (Beispiel)

Berechnung lokale Energie:

- Summe der quadrierten Differenz zum oberen und linken Nachbarn (falls vorhanden)

5	5	3	3	3
3	3	3	3	0
3	3	3	0	1
3	3	0	1	1
3	3	0	1	1

(a) Originalbild.

lokale
Energie →

			18	

(b) Lokale Energie.

Lokale Energie (Beispiel)

Berechnung lokale Energie:

- Summe der quadrierten Differenz zum oberen und linken Nachbarn (falls vorhanden)

5	5	3	3	3
3	3	3	3	0
3	3	3	0	1
3	3	0	1	1
3	3	0	1	1

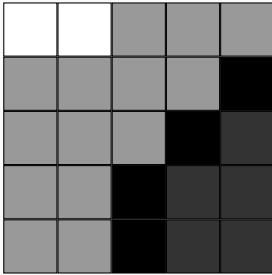
(a) Originalbild.

lokale
Energie →

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

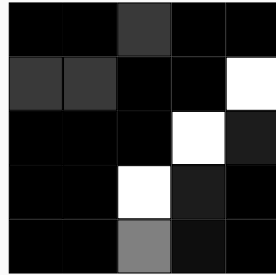
(b) Lokale Energie.

Lokale Energie (Beispiel)



(a) Originalbild.

lokale
Energie →



(b) Lokale Energie.

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

×	a	b	c	×
×	×	y		

(b) Akkumulierte
Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9		

(b) Akkumulierte
Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9		

(b) Akkumulierte
Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	

(b) Akkumulierte
Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	2

(b) Akkumulierte
Energie.

Akkumulierte Energie (Beispiel)

Akkumulierte Energie:

0	0	4	0	0
4	4	0	0	18
0	0	0	18	2
0	0	18	2	0
0	0	9	1	0

(a) Lokale Energie.

akkumulierte
Energie →

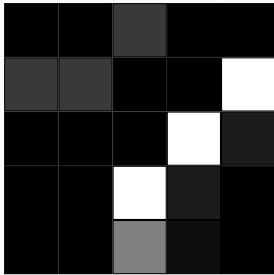
0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	2

(b) Akkumulierte
Energie.

→ Akkumulierte Energie:

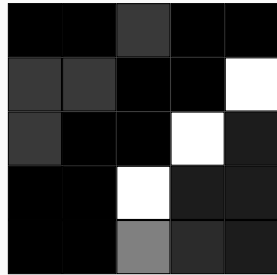
„Minimale Kosten von diesem Pixel nach oben.“

Akkumulierte Energie (Beispiel)



(a) Lokale Energie.

akkumulierte
Energie →



(b) Akkumulierte
Energie.

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Pfad der minimalen Energie (Beispiel)

- Finde den Pixel mit minimaler Energie in der untersten Reihe
 - Akkumulierte Energie:
„Minimale Kosten von diesem Pixel nach oben.“
- Suche Pfad mit minimalen Kosten von diesem Pixel nach oben
 - Zu entfernender Pfad

Pfad der minimalen Energie (Beispiel)

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	2

Abbildung 8: Akkumulierte Energie.

Pfad der minimalen Energie (Beispiel)

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	2

Abbildung 8: Akkumulierte Energie.

Pfad der minimalen Energie (Beispiel)

0	0	4	0	0
4	4	0	0	18
4	0	0	18	2
0	0	18	2	2
0	0	9	3	2

Abbildung 8: Akkumulierte Energie.

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades

Entfernung Pixelpfad (Beispiel)

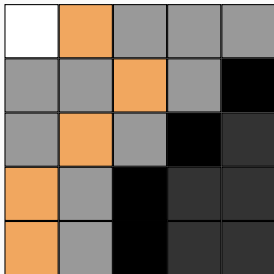


Abbildung 9: Originalbild.

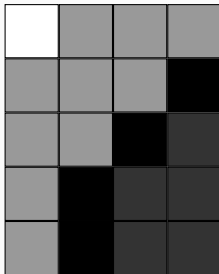
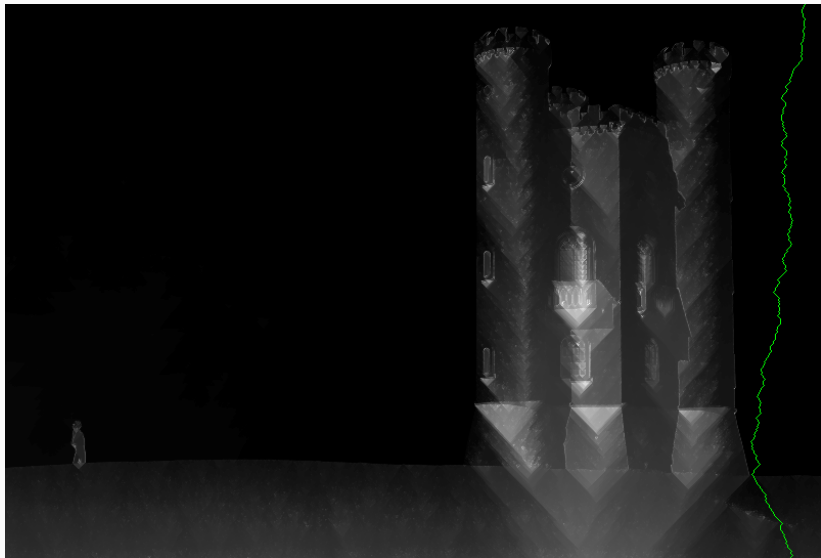


Abbildung 10: Originalbild verkleinert.

Seamcarving Algorithmus Überblick

- Berechnung der lokalen Energie jedes Pixels
→ Kantenerkennung
- Berechnung der Energie vertikaler Pixelpfade
- Berechnung des Pfades mit minimaler Energie
- Entfernung der Pixel dieses Pfades





Realbeispiel



Bildformat

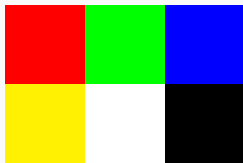
P3

3 2

255

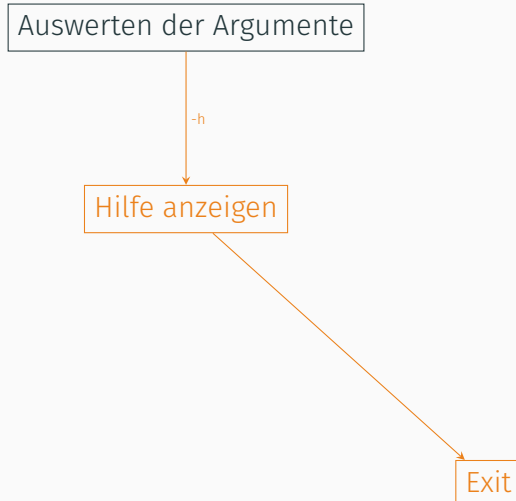
255 0 0 0 255 0 0 0 255

255 255 0 255 255 255 0 0 0

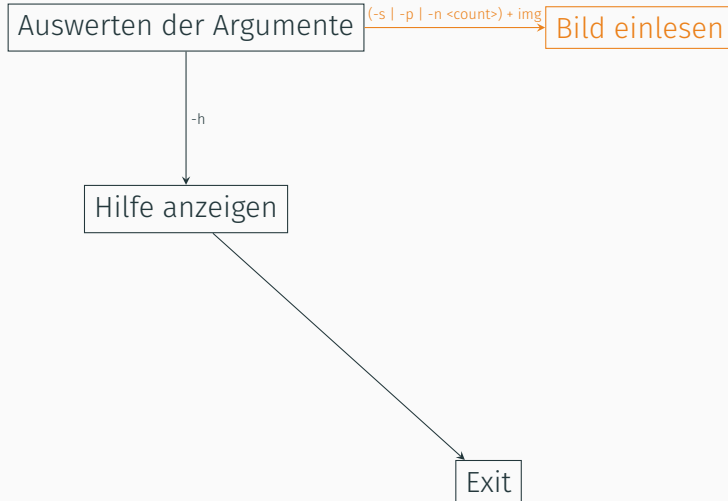


Einführung in das Programm

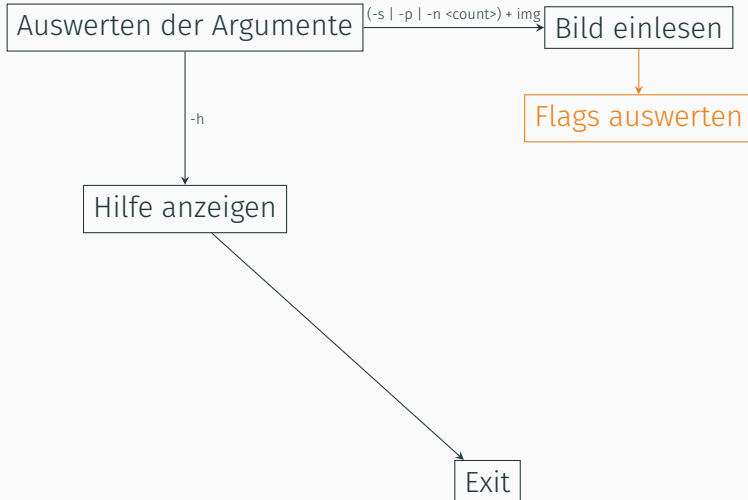
Auswerten der Argumente



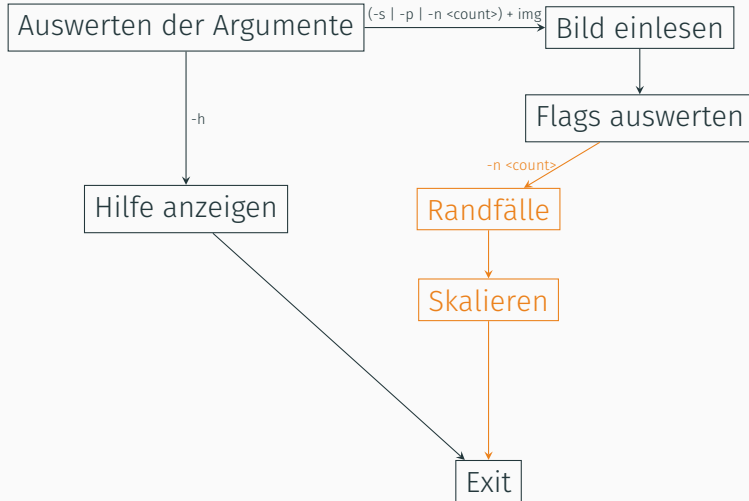
Programmfluss



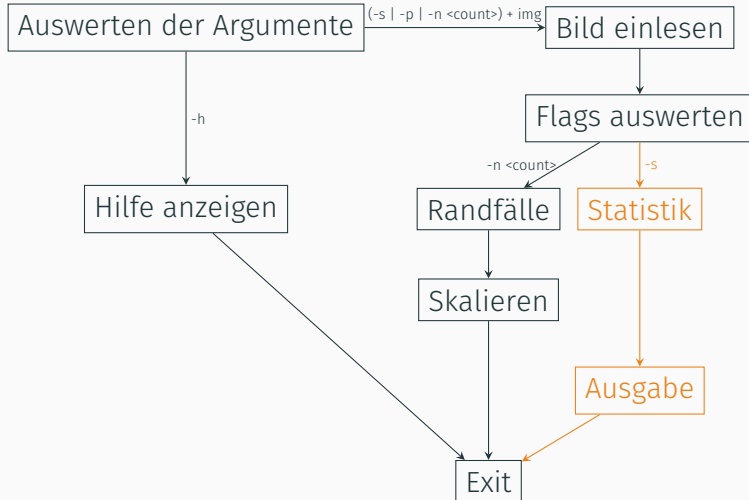
Programmfluss



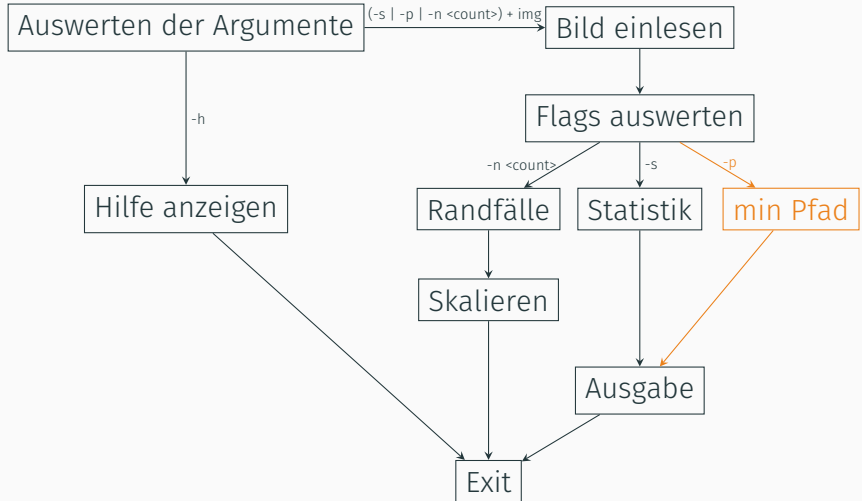
Programmfluss



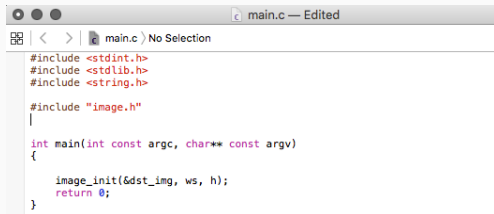
Programmfluss



Programmfluss



Schritt. Verfeinerung

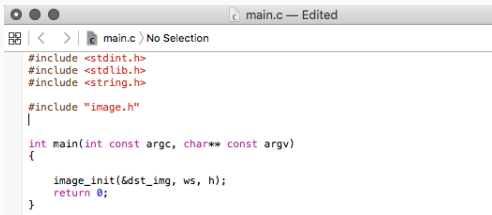


```
#include <stdint.h>
#include <stdlib.h>
#include <string.h>

#include "image.h"
|

int main(int const argc, char** const argv)
{
    image_init(&dst_img, ws, h);
    return 0;
}
```

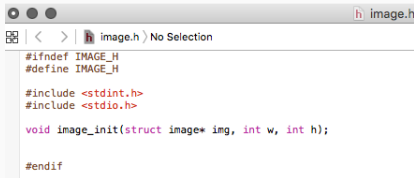
Schritt. Verfeinerung



```
#include <stdint.h>
#include <stdlib.h>
#include <string.h>

#include "image.h"
|

int main(int const argc, char** const argv)
{
    image_init(&dst_img, ws, h);
    return 0;
}
```

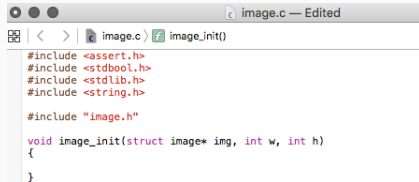


```
#ifndef IMAGE_H
#define IMAGE_H

#include <stdint.h>
#include <stdio.h>

void image_init(struct image* img, int w, int h);

#endif
```



```
#include <assert.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

#include "image.h"

void image_init(struct image* img, int w, int h)
{
}
```

Fragen?