

応用情報工学演習

課題3 シームカービング

谷口研究室

課題3

- シームカービングという画像処理技術を実装する
 - SeamCarving.pyの穴埋め
 - 詳細はseamcarving_ex.pdfを参照
- 提出締切：11/28（木） 16:00
- 提出物：
 - ソースコード（処理概要がわかるようにコメントを入れること）
- 提出先：LETUSの課題提出BOX

シームカービング (Seam Carving)

- 画像のリサイズ技術の一つ



元画像



シームカービング



スケーリング
(画像全体で一定の縮小)



クロッピング
(切り出し)

シームカービング (Seam Carving)

- 画像のリサイズ技術の一つ



元画像



シームカービング



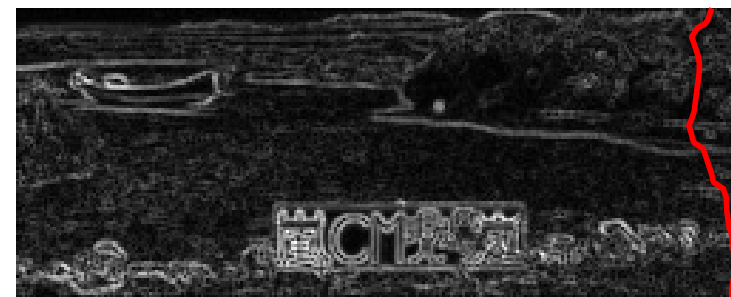
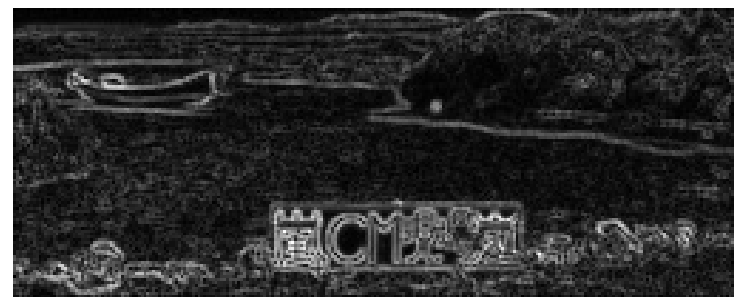
スケーリング
(画像全体で一定の縮小)



クロッピング
(切り出し)

シームカービング

- 取り除いても目立たないシーム（画像の上から下まで走る幅1画素の経路）を繰り返し取り除く



(i) エネルギーマップの計算
（目立つ場所ほど白い）

(ii) 最適なシームの計算

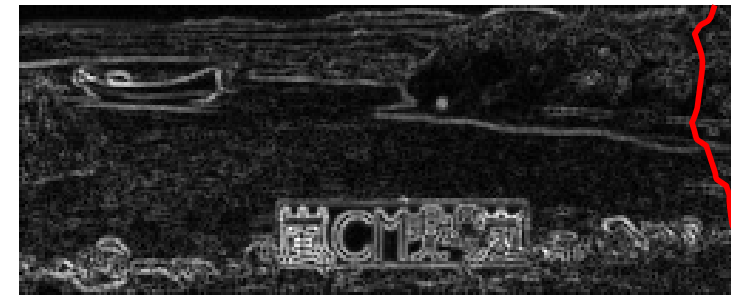
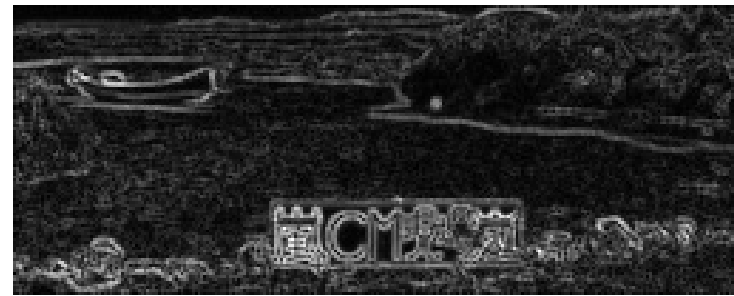
(iii) シームを取り除き、
1画素分だけ横幅を小さくする

(iv) 指定した横幅になるまで繰り返す

シームカービング

- 取り除いても目立たないシーム（画像の上から下まで走る幅1画素の経路）を繰り返し取り除く

STEP 1



(i) エネルギーマップの計算
（目立つ場所ほど白い）

(ii) 最適なシームの計算

(iii) シームを取り除き、
1画素分だけ横幅を小さくする

(iv) 指定した横幅になるまで繰り返す

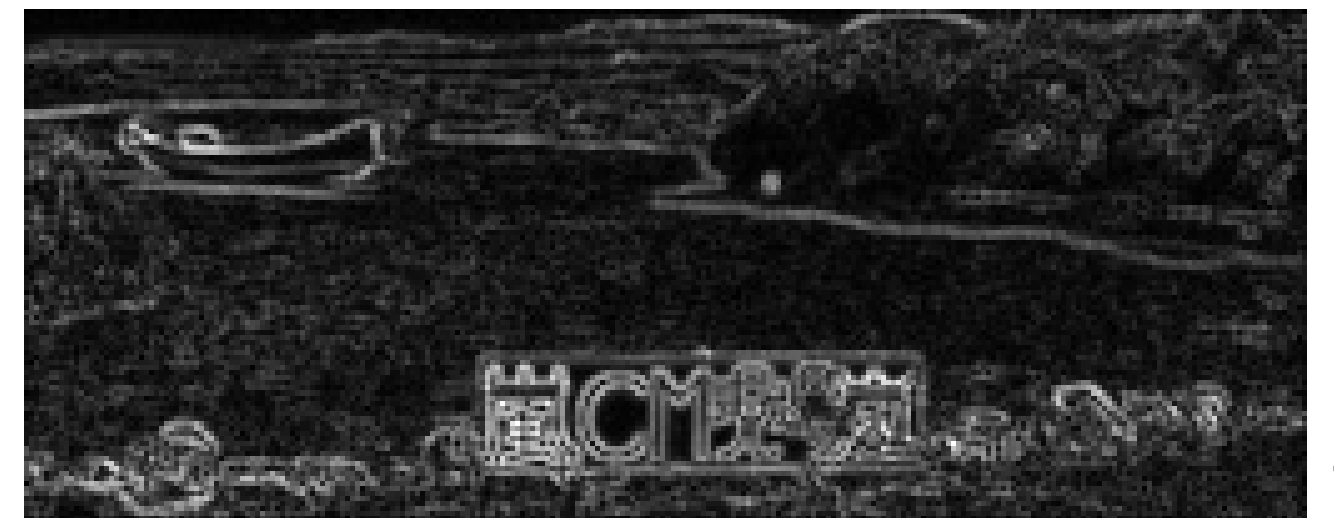
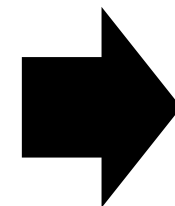
(i) エネルギーマップの計算

- 各画素でエネルギー（取り除くとどれだけ目立つか）を計算
 - エッジ付近（画素値が急激に変化する場所）は目立つ
- 微分フィルタでエネルギーマップを計算する

y行x列目の
エネルギーマップの値

y行x列目の
入力画像の値

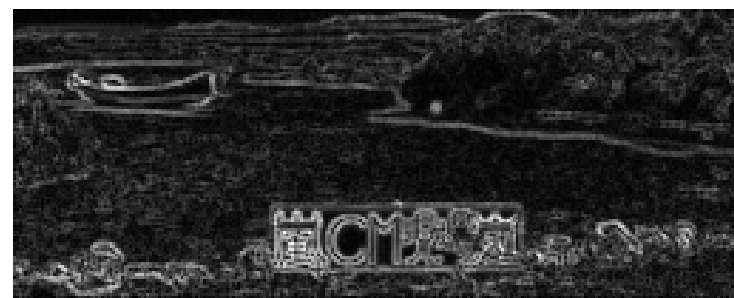
$$E(y, x) = \underbrace{\left| \frac{\partial}{\partial x} I(y, x) \right|}_{\text{x方向のSobelフィルタ}} + \underbrace{\left| \frac{\partial}{\partial y} I(y, x) \right|}_{\text{y方向のSobelフィルタ}} + \underbrace{\left| \frac{\partial^2}{\partial x^2} I(y, x) + \frac{\partial^2}{\partial y^2} I(y, x) \right|}_{\text{Laplacianフィルタ}}$$



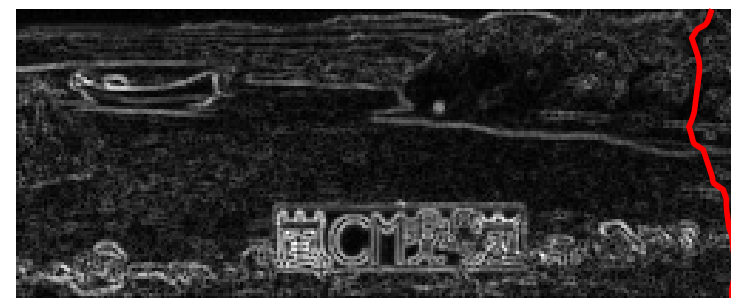
シームカービング

- 取り除いても目立たないシーム（画像の上から下まで走る幅1画素の連続な経路）を繰り返し取り除く

STEP 2



(i) エネルギーマップの計算
（目立つ場所ほど白い）



(ii) 最適なシームの計算



(iii) シームを取り除き、
1画素分だけ横幅を小さくする

(iv) 指定した横幅になるまで繰り返す

(ii) 最適なシームの計算

- 経路上のエネルギーの和が最小になるようなシーム（画像の上から下まで走る連続な経路）を見つける



(ii) 最適なシームの計算

- 経路上のエネルギーの和が最小になるようなシーム（画像の上から下まで走る連続な経路）を見つける

簡単なエネルギーマップの例を使って説明する

6	4	2	0	5	1	7
3	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

(ii) 最適なシームの計算

- 経路上のエネルギーの和が最小になるようなシーム（画像の上から下まで走る連続な経路）を見つける

簡単なエネルギーマップの例を使って説明する

6	4	2	0	5	1	7
3	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

最適なシーム（エネルギーの和は $0+1+2+0=3$ ）

(ii) 最適なシームの計算

- 動的計画法による見つけ方
 - エネルギーの累積計算
 - 最小値を見つける
 - バックトラック

6	4	2	0	5	1	7
3	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
3	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

Diagram illustrating the dynamic programming process for finding the optimal seam. The grid shows energy values. The current cell being processed is (1,0) with value 3, highlighted in green. A blue arrow points down from (0,0) to (1,0). A red arrow points from (1,0) to (1,1), which is circled in red and labeled '+ 4', indicating the addition of the current cell's energy to the cumulative value from the previous cell.

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	7	1	+0	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	7	1	9	8	1	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	7	1	9	8	1	8
9	8	6	3	9	8	9
3	3	5	4	0	6	7

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	7	1	9	8	1	8
9	8	6	3	9	8	9
11	9	8	7	3	14	7

注目画素に到達可能な経路のうち、エネルギーの低いほうを選択して累積する
※この際に選んだ経路を保存しておく

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

- 2. 最小値を見つける**

3. バックトラック

6	4	2	0	5	1	7
7	7	1	9	8	1	8
9	8	6	3	9	8	9
11	9	8	7	3	14	15

一番下の行において最小値を見つける

※この値が考えられる全経路のうち最小の累積エネルギーとなっている

(ii) 最適なシームの計算

- 動的計画法による見つけ方

1. エネルギーの累積計算

2. 最小値を見つける

3. バックトラック

6	4	2	0	5	1	7
7	7	1	9	8	1	8
9	8	6	3	9	8	9
11	9	8	7	3	14	15

保存した経路を戻ることによって最適なシームを求めることができる

(ii) 最適なシームの計算

- 経路上のエネルギーの和が最小になるようなシーム（画像の上から下まで走る連続な経路）を見つける

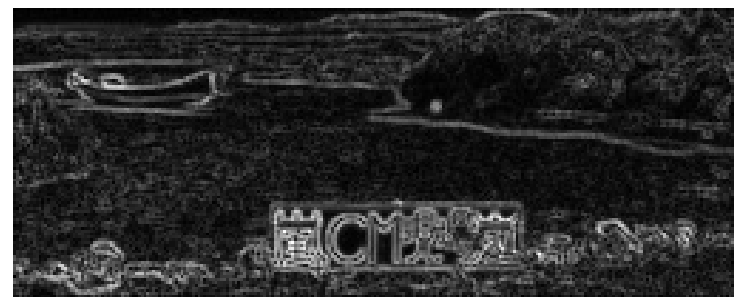
列						
0	1	2	3	4	5	6
6	4	2	0	5	1	7
3	5	1	9	8	0	7
2	7	5	2	8	7	8
3	3	5	4	0	6	7

シームを表すベクトル
(各行で何番目の列を取り除くか)

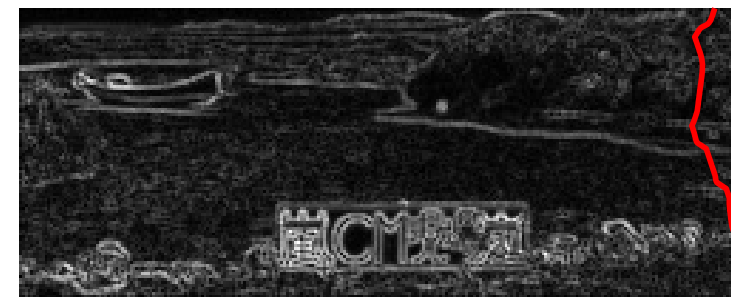
$$\mathbf{s} = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

シームカービング

- 取り除いても目立たないシーム（画像の上から下まで走る幅1画素の連続な経路）を繰り返し取り除く



(i) エネルギーマップの計算
（目立つ場所ほど白い）



(ii) 最適なシームの計算



(iii) シームを取り除き、
1画素分だけ横幅を小さくする

(iv) 指定した横幅になるまで繰り返す

実装済み

実装済み

オプション課題

- 絶対に取り除かれない領域をマウスで指定できる機能をPyQtのGUIで実装しなさい
 - 実装済みの部分を好きに書き換えてよい
 - 取り組んだ場合はSeamCarving.pyとは別ファイルで提出すること

