

画像処理 レポート

氏名: 木下直樹

学籍番号: 09425521

平成 28 年 1 月 22 日

1 概要

パノラマ画像の自動生成を行うには、両方の画像の同じ物体上に出現している特徴点を自動で探し出す必要がある。この際、第 1 画像の第 i 特徴点と第 2 画像の第 j 特徴点が似ているかどうかを判定する。このための最も簡単な類似度尺度は Sum of Squared Differences である。即ち、特徴点周辺の小領域の画像を高次元ベクトルで表し、高次元空間のユークリッドノルムで画像の類似度を測る。類似度の高い特徴点对を 4 組以上選ぶことで、正しい変換行列の算出が可能となる。

2 greedy.c の作成

特徴点の対応付けプログラム greedy.c を作成する。このプログラムには、TKfilter.c で得た特徴点座標を使用するため、これを greedy.c で使用するためにファイルへ出力するコードを追加しておく。

準備として、出力された二つの画像とそれらの画像の特徴点座標を用い、一方の画像の特徴点一点が他方の画像の特徴点全ての点それぞれに対して特徴点が似ているかどうかの計算を一方の画像の特徴点全てに実行し、それを行列の構造体へ格納する。

行列式は以下ようになる。(ただし、double 型の値を 100000 で割、int 型にキャストした値を表示するため、実際に使用した値とは異なる)

56	84	73	79	88	78	68	64	59	67	45	72	77	72	71	77	69	80	61	88	75	90	75	47	57	38	75	60	71	58
53	84	70	71	60	62	58	62	70	54	61	73	63	65	69	68	65	43	47	78	58	69	88	51	44	47	70	60	64	38
78	83	11	51	85	41	35	54	64	52	59	50	83	52	66	55	38	60	52	90	49	52	63	68	43	51	93	47	40	48
66	80	49	46	45	51	40	60	49	37	60	39	93	48	57	56	53	63	44	75	44	45	44	61	42	45	80	40	54	43
89	114	91	74	7	80	71	80	71	33	100	69	114	60	74	71	98	69	59	92	56	60	81	85	64	75	99	73	86	55
78	99	35	10	70	49	22	71	71	46	66	57	110	58	76	66	64	70	65	111	51	49	73	84	53	64	106	53	57	40
69	75	40	44	68	41	43	65	41	56	58	44	94	50	71	48	53	63	53	73	36	44	46	56	45	51	101	30	42	45
72	102	40	28	53	47	13	62	61	41	69	57	102	59	73	67	67	67	61	97	45	54	68	71	52	56	89	60	64	33
78	78	42	51	72	13	36	50	54	65	62	69	73	48	53	61	59	52	52	50	35	57	73	66	45	56	77	46	38	36
74	74	53	60	43	63	44	64	59	16	56	21	95	47	68	52	47	56	39	84	41	46	56	49	37	44	90	47	51	44
85	64	54	65	60	67	52	76	65	32	59	2	110	43	73	49	48	54	46	85	39	39	58	52	31	54	102	38	48	59
63	42	76	102	107	67	88	82	66	81	30	70	54	49	68	63	50	50	42	47	56	82	73	37	58	39	63	43	56	66
64	69	65	61	72	51	64	57	9	62	57	63	109	45	64	42	82	78	70	66	43	48	37	52	46	67	130	32	47	59
51	64	65	75	90	45	58	46	46	76	47	80	48	68	48	67	63	66	44	46	49	81	64	40	61	41	58	49	50	51
86	71	62	60	71	66	72	99	52	60	71	49	144	33	89	33	82	67	74	88	41	29	44	86	46	87	157	30	53	75
51	56	59	77	85	58	58	43	62	56	38	62	46	62	47	63	38	51	35	66	49	66	66	33	40	5	48	47	47	38
84	67	97	98	74	67	83	63	78	73	80	84	63	57	53	74	72	58	43	50	48	79	91	60	65	58	61	60	62	59
63	58	65	61	82	68	74	78	48	52	54	51	114	46	74	21	80	65	59	86	43	32	39	56	45	78	141	27	51	68
84	60	101	121	101	55	102	74	67	102	74	100	66	56	63	85	79	72	52	8	59	95	85	59	79	64	60	61	62	74
87	75	98	131	140	77	108	68	105	114	68	128	6	97	62	109	62	69	45	69	87	125	132	63	86	53	45	82	73	73
63	51	49	50	53	54	47	52	41	37	42	33	90	38	46	33	41	40	52	77	30	33	54	39	8	40	87	34	37	45

75	47	44	62	80	34	61	52	44	54	53	46	76	37	44	30	41	40	39	57	29	30	58	50	29	52	91	22	4	55
54	65	48	51	54	59	48	58	44	34	51	31	98	46	65	27	51	45	49	91	35	29	47	46	24	42	103	34	43	40
53	37	77	97	91	64	79	65	59	69	38	61	48	63	46	56	49	51	40	49	50	75	71	19	44	34	61	46	47	64
52	71	67	69	32	60	60	51	66	29	57	61	56	52	50	56	53	47	25	60	44	57	66	49	46	34	51	51	56	34
69	40	45	54	85	49	54	63	51	55	35	46	77	46	41	46	41	54	55	74	36	39	67	41	18	50	95	37	31	57
71	47	50	59	76	49	67	70	42	53	54	45	99	30	57	24	67	52	46	61	30	26	43	52	39	70	122	17	28	64
65	81	40	43	58	40	34	49	57	45	57	60	64	53	60	67	40	60	38	73	43	62	67	54	44	35	62	53	53	11
60	57	39	37	60	48	36	60	43	40	37	35	93	37	50	34	44	47	55	80	29	29	46	46	18	42	94	32	37	43
66	74	47	63	72	38	41	48	61	55	58	57	60	56	56	69	40	53	37	59	48	64	66	54	42	27	49	47	43	29

第一画像の第 (行数) 番目の特徴点と第二画像の第 (列数) 番目の特徴点の SSD(差の二乗和) が 30 点 × 30 点 = 900 点格納される。この行列の値は小さいものほど対応する特徴点二点が一致している可能性が高いといえる。そのため greedy.c でもっとも重要とされるのはこの行列中の値で小さいものを探索し、第一画像の特徴点と第二画像の特徴点をうまく対応させる処理である。その処理として、以下の様な方法を実装した。

- 方法 1

第一画像で特徴点らしさの高い点から第二画像の特徴点との対応を決定する。処理手順としては、上記の様な行列の一番上の行から一行ずつその行の中で一番値の小さな点を探索し、第一画像の行数番目の特徴点と第二画像の列数番目の特徴点が対応すると決定する。

行列の決定した点の列の値に無限値を格納し、それ以降の探索で同じ点を採用しないようにする。実行結果の特徴点の対応セットは 30 セット中 12 セットが一致していた。特徴点をより正確に一致させるために次の様な方法を採用した。

- 方法 2

行列の中の値全てを探索し、一番値の小さい値の対応点を決定とする。対応点が決まるとその点の行と列の値に無限値を格納し、それ以降の処理で同じ点を取らない様にする。実行結果は 30 セット中 19 セットが実際に一致していた。

3 問題と改善案

今回作成した greedy.c では方法 1 で取り組んだ場合も方法 2 で取り組んだ場合も現状 30 個の特徴点を全て正確に対応させられていない。しかし、この問題は greedy.c だけの問題ではない。greedy.c の入力ファイルである 0.fea と 2.fea に記された座標はそれぞれ 30 点であるが、一方の特徴点他方の画像中にあるとは限らないのである。そのため、いくら greedy.c の性能を向上させても 30 点全てを正確に対応させることは物理的に不可能なのである。

- 改善案 1

TKfilter.c から出力される特徴点を 100 個にするなど、余分に多く特徴点を出力する。これにより、多くの特徴点の中から対応する特徴点のセットを考えることができ、greedy.c で指定する特徴点セットの数が少ないほどその指定した数の特徴点セットの信頼性が高くなり、また今回の 30 点で試行する場合でも方法 2 よりもよい結果になることが期待できる。しかし、現状の greedy.c の処理では方法 1 も方法 2 も処理が重いため、行列の大きさが飛躍的に大きくなってしまふこの改善案を採用するには greedy.c の処理効率を向上させなければならない。

- 改善案 2

TKfilter.c から出力される一方の画像の特徴点座標を他方の画像中にあるもののみとする条件を加える。これにより greedy.c が捌く行列は大きくならないため、処理の軽量化が期待できる。ただしこの方法では、一方の画像中の点他方の画像中にあるかないかを判断できることが前提である。