

SIFT、BRIEF と ORB の比較

SIFT

Scale-Invariant Feature Transform (SIFT)は4つの処理に大別されます。

1. スケール空間における極値検出：Difference of Gaussian (DoG)を使います。
2. キーポイントの位置同定：2つの固有値の差が閾値より大きければ、そのキーポイントは候補から除外されます。
3. 回転角の計算：回転不変性を実現するため。
4. 特徴量の記述：画像勾配の大きさと向きに基づいて各キーポイントの特徴量記述子を計算する。

この方法の検出器はスケール不変です。

参考：https://docs.opencv.org/master/da/df5/tutorial_py_sift_intro.html

BRIEF

Binary Robust Independent Elementary Features (BRIEF)は特徴量記述子を使うことなく直接2値ベクトルを計算します。平滑化した画像パッチに対して n_d 個の画素(x,y)のペアを構築します。次に、各ペアに対して画素値を比較します。

SIFT は 128 次元の実ベクトル(浮動小数)を計算します。このような特徴点が数千個もあると想像してください。マッチングの際にメモリ使用量が増大し計算時間がかかってしまいます。高速化のために SIFT 特徴量を圧縮できます。それでも、まず初めに SIFT 特徴量を計算しなければいけません。ここでは BRIEF という省メモリかつ高速なマッチングが可能な二値ベクトルを計算する特徴量記述子を使います。

参考：https://docs.opencv.org/master/dc/d7d/tutorial_py_brief.html

ORB

Oriented FAST and Rotated BRIEF (ORB) は基本的に FAST による特徴点検出と BRIEF による特徴量記述子を組合わせたものです。まず始めに FAST によって特徴点を検出し、Harris のコーナー評価により上位 N 点を選びます。また、マルチスケールの特徴を得るため、ピラミッドを使います。

計算コスト，マッチング精度，特許を考慮すると SIFT と SURF の良い代替と言えます。

参考：https://docs.opencv.org/master/d1/d89/tutorial_py_orb.html

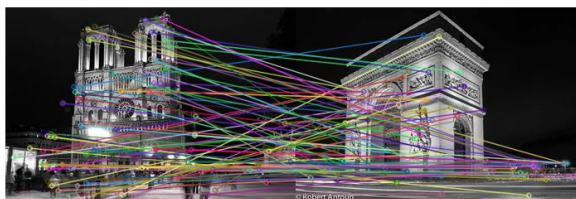
出力

Kp: 検出したキーポイント

- 建物



1



2

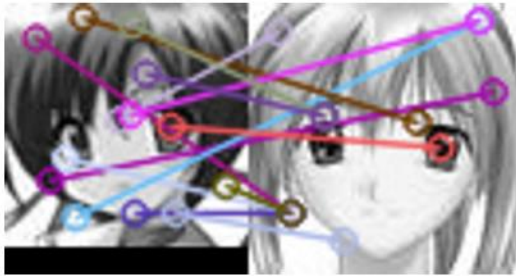


3

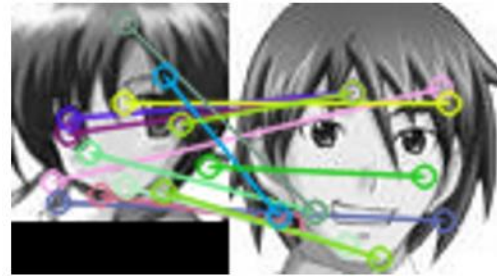
	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.072	739	898	238	29	0.05	739	547	100	15.5	0.06	898	547	147	20.3
BRIEF	0.008	114	226	27	15.8	0.006	114	113	31	27.3	0.006	226	113	62	36.5
ORB	0.017	500	500	98	19.6	0.017	500	495	87	17.4	0.019	500	495	73	14.6

- BRIEF 手法が一番は早い。
- 図 1 が一番 Rate が高いはずなので、BRIEF の結果は良くないと考えられます。

- アニメ顔画像



1



2

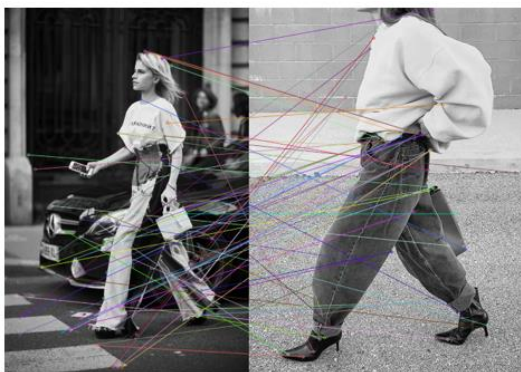


3

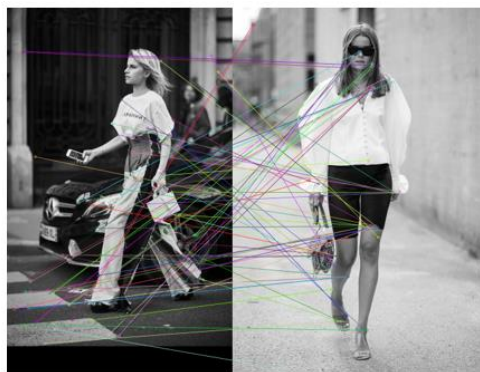
	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.004	59	71	13	20	0.004	59	39	13	26.5	0.003	71	39	15	27.2
BRIEF															
ORB	0.001	0	5	0	0	0.00	0	1	0	0					

- BRIEF と ORB 手法はうまくいかなかった。画像質が低いまたは画像サイズが小さいのが原因かもしれません。キーポイントをうまく検出できなかった。

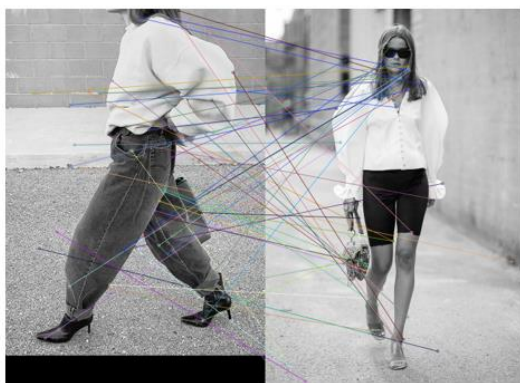
- 服



1



2

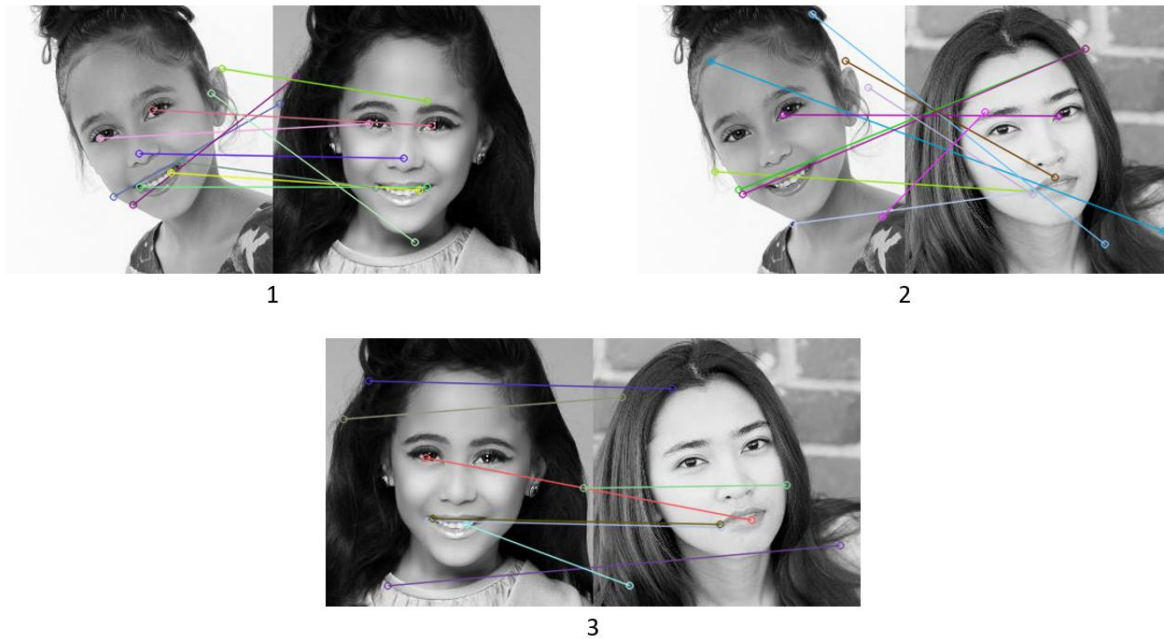


3

	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.323	1810	4145	69	2.31	0.24	1810	922	74	5.41	0.27	4145	922	62	2.44
BRIEF	0.04	497	309	135	33.4	0.04	497	194	150	43.4	0.05	309	194	73	29.0
ORB	0.05	500	500	82	16.4	0.04	500	500	82	16.4	0.05	500	500	89	17.8

○ 図 1 の似ているジーンズを検出できなかった。単色で処理するからだと思います。

- 顔画像



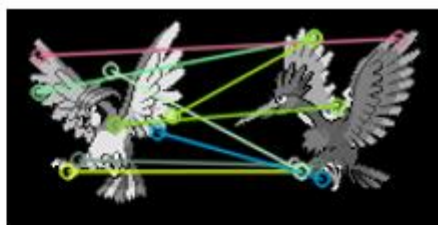
	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.02	167	154	10	6.23	0.023	167	187	10	5.64	0.024	154	187	8	4.69
BRIEF	0.00	16	25	5	24.3	0.00	16	22	1	5.26	0.00	25	22	1	4.25
ORB	0.00	371	390	21	5.51	0.01	371	342	13	3.64	0.00	390	342	25	6.83

- 図 1 が同じ人物の画像なので、Rate が一番高いはずなので、ORB の結果だけが間違った。
- BRIEF 手法は顔の特徴を検出するのが得意と考えられます。

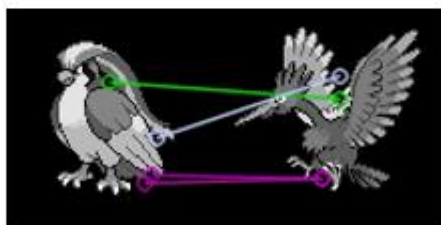
- ポケモン



1



2

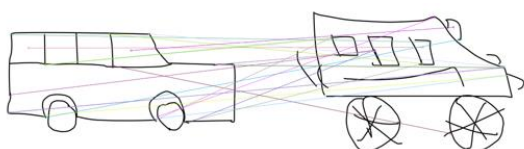


3

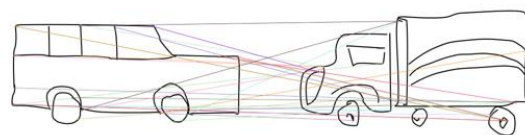
	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.009	68	46	4	7.01	0.006	68	57	8	12.8	0.009	46	57	4	7.76
BRIEF	0.000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORB	0.002	61	31	1	2.17	0.001	61	35	10	20.8	0.001	31	35	1	3.03

○ BRIEF がキーポイントを検出できなかった。

- スケッチ



1



2



3

	1					2					3				
	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)	Time (sec)	Kp1	Kp2	Match	Rate (%)
SIFT	0.42	109	392	27	10.77	0.36	109	217	31	19.01	0.35	392	217	77	25.28
BRIEF	0.14	139	376	27	10.48	0.142	139	251	34	17.43	0.14	376	251	66	21.05
ORB	0.067	500	500	71	14.2	0.06	500	500	65	13.0	0.066	500	500	50	10.0

- 図 1 は同類の車なので、ORB 方法だけ正しい結果得られた。
- 図 3 は違う種類ですが、構造的には似ているので SIFT と BRIEF の Rate が高かった。