

論文

SIFT アルゴリズムを応用した環境地図作成方法

大平 倫宏*¹⁾ 富山 真一*²⁾

Environmental map generation applying scale invariant feature transform algorithm

Norihiro Ohira*¹⁾, Shinichi Tomiyama*²⁾

A method of environmental map generation is reported herein. In the case of generating an environmental map, we usually input featureless images like images of the ceiling or a floor and their illumination change. The SIFT (Scale Invariant Feature Transform) algorithm detects local feature points and extracts image features that are reasonably invariant to changes in illumination, image noise, rotation, scaling, and small changes in viewpoint.

In this paper, we focused our attention on the image composition of featureless images applying the SIFT algorithm and proposed one simple method. The proposed method could run in the order of $n \log n$ time to compose two images of order n pixels. Also, the image composition was precise enough to be used, inputting featureless data like images of a ceiling.

キーワード：環境地図, SIFT アルゴリズム, 画像合成

Keywords : Environmental map, SIFT algorithm, images compose

1. はじめに

今日の少子高齢化社会に対応するために、自立的に複数のカメラを連動させて、防犯や福祉目的のために活用するシステムへの関心が高まってきている。このようなシステムの構築のためには、周辺の様子（ドアの開閉や障害物の存在）を記述した地図（環境地図）の作成が必要となる。しかし、環境地図は変化の乏しい天井や床等の画像から作成する機会が多いため、従来の画像合成法では、合成のずれが大きくなる可能性がある。

そこで本研究では、特徴量抽出アルゴリズム SIFT (Scale Invariant Feature Transform) を応用して、変化の乏しい画像群から環境地図を合成する方法を考案した。本提案手法の合成の最悪時間計算量は、画像の画素数を n とし、入力画像が 2 枚の時にオーダー表記で $O(n \log n)$ 時間となり、全探索するような方法でかかる $O(n^2)$ 時間に比べ、高速に合成が可能となる。

2. 画像合成方法

2.1 SIFT アルゴリズム SIFT アルゴリズムは、画像における局所的な特徴点を抽出した後各特徴点に対応する特徴量を抽出するアルゴリズムである⁽¹⁾。SIFT アルゴリズムは、画像のスケール・回転・圧縮率の変化等に対して不変な特徴量を抽出が可能であることが報告されている⁽²⁾。また、照明変化に対しても、影響の少ない特徴量となること

が知られており、照明変化の有り得る環境において環境地図を作成するのに適していると思われる。

図 1 を入力画像として、SIFT アルゴリズムによる特徴量抽出を行った画像を図 2 に示す。各矢印が特徴量を示しており、矢印の根元が特徴点を表している。



図 1. 原画像

*1) 情報技術グループ

*2) 日本大学大学院理工学研究科

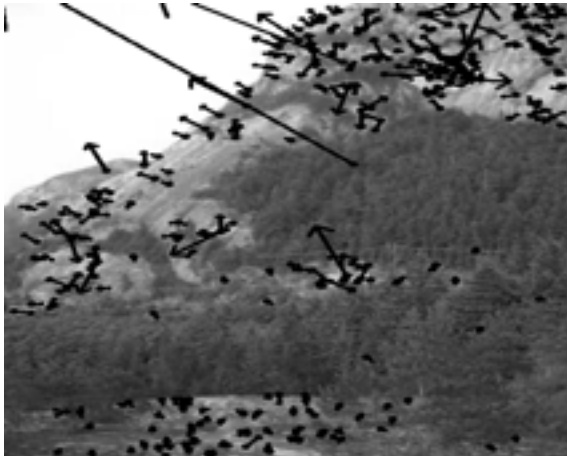


図2. 2枚の画像に対し特徴点の対応付けを行った結果

2.2 提案する画像合成手順 本研究では、SIFT アルゴリズムを用いて、異なる画像間における共通する特徴点、特徴量を算出し、画像合成を行う。以下に本研究で提案する画像合成アルゴリズムの処理手順についてまとめる。

(1) **特徴点・特徴量の算出** 入力画像全てに対して、それぞれの特徴点・特徴量を、SIFTアルゴリズムにより算出する。

(2) **共通する特徴量の検索** 入力画像間で抽出された各特徴量の比較を行い、一致する場合にその特徴点同士を対応付ける。また、一致する特徴量の個数に閾値を設定し、一致する特徴量の個数が閾値以上かつ最大であれば、合成対象候補として(3)以降の処理を行う。閾値以下であった場合は、合成対象の候補から除外する。

図3に特徴点の対応付けを行った結果を表す。右側と左側で別の入力画像となっており、対応する特徴点同士を線で結んでいる。



図3. 2枚の画像に対し特徴点の対応付けを行った結果

(3) **対応付ける特徴点の絞り込み** 単に特徴量と同じ特徴点を対応付けるだけでは、図3の中で傾きが小さく長い線の場合のように、明らかに無関係であると思われる点まで、対応付けられてしまうことがある。このようなノイズを削減するために、対応する特徴点の絞り込みを行う。具体的には、図6に示すように特徴点を対応付けた線の角度

ヒストグラムを作成し、最頻の値となる対応付けだけに絞り込む。このような操作を行うことで、図4のように絞り込むことができる。



図4. 図3で対応する特徴点の絞り込みを行った結果

(4) **合成に最適な座標候補の検索** 合成対象候補の画像の組に対して、対応する特徴点間の距離の平均が最小となるような位置を探索し、その位置で合成を行う。探索方法としては、2分探索法を利用する。

図5は図4の2枚の画像に対して、対応する特徴点間の距離の合計が最小となるような位置を探索し、合成した結果を表している。なお、合成により画像が重なる部分については、平均を取った後に明度を上げて表示している。



図5. 最適な位置の探索結果

(5) **終了条件の判定** (2)の処理において一致する特徴量の個数が閾値以下の場合もしくは、入力画像全てに対して(2)~(4)の処理を行った場合に終了する。それ以外は上記処理を繰り返す。この手法による画像合成にかかる最悪時間計算量は、画像の画素数を n として、入力画像が2枚の時に $O(n \log n)$ となり、入力画像が m 枚の時には $O(m! n \log n)$ なる。

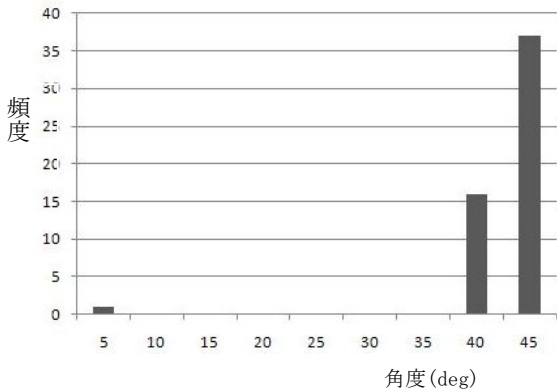


図6. 角度ヒストグラム

3. 結果と考察

3.1 合成の精度 合成対象として、当初の目的であった変化に乏しい画像を用いて提案手法の合成精度を調査した。入力画像は、図7のような天井画像3枚とした。比較対象として、画像合成によく用いられるフリーソフトである AutoStitch⁽³⁾で合成を試みたが、入力画像が変化に乏しく十分な特徴量が抽出できなかつたためか、エラーとなり合成ができなかつた。そのため、商用のソフトである ArcSoft PhotoBase Pro⁽⁴⁾の試用版を用いて、手動でおおまかな合成位置を入力し、比較を行った。ArcSoft PhotoBase Pro では、おおまかな合成位置を手動で入力する必要があり、そこからより精密な合成位置を探索しているようである。



図7. 合成ソフトへの入力画像：天井画像3枚

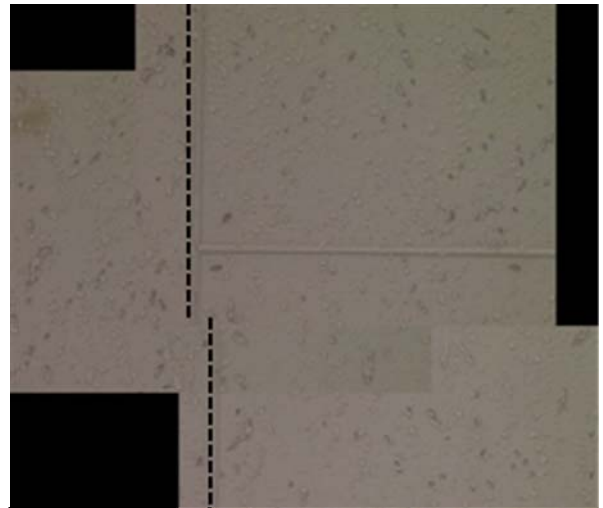


図8. 天井画像に対して従来法を用いた画像合成結果



図9. 天井画像に対して提案法を用いた画像合成結果

実際に合成を行った結果は図8、図9のようになる。ArcSoft PhotoBase Proを用いて合成した結果が図8である。今回提案手法を用いて、2.2節の(1)から(5)の手順に従って合成した結果が図9である。ただし、図9では、合成処理により重なった部分の明度を上げる処理を行わないで、上から重ねて表示してある。また、比較のために、どちらの図にも、合成画像に対して、縦の溝の線に沿って黒の点線を補助的に上から書き加えて表示してある。実際の合成画像自体には、点線は引かれていないため、注意が必要である。

図8と図9において、特に縦に入っている溝の部分を目安として比べると、図9では黒の点線のずれがほぼないが、図8においては黒の点線がずれて合成されているのが見て取れる。図8は実際には、10 pixelほどずれて合成されている。このように、環境地図を作成する際に、入力画像として用いられる変化の乏しい画像においては、AutoStitchでは画像合成そのものが出来ず、他の合成ソフトでは手動でお

おまかな合成位置の入力を行っても 10 pixel ほどずれるといふ結果になった。提案手法は、そのような従来方法に比べ、環境地図を作る際に変化に乏しい画像を入力としても精度良く合成可能であると思われる。また、入力画像さえ指定すれば、後の計算は全て自動で行うので、その面でも優れていると思われる。

3.2 合成の処理時間 今回提案した手法を用いて、画像合成処理にかかった時間をまとめた表が表 1 である。それぞれ同じ入力画像に対して画像サイズの変更を行い、2 枚の画像を合成した際の処理時間を表している。

実行環境としては、OS は Windows XP 上で、CPU は Pentium M 1.6 GHz、メモリ 512 MB の PC で、開発は Visual C++ 2005 Express Edition 上で行っている。表から見てわかるように、入力画像のサイズが大きくなっても、処理時間はそれほど大きく変わらない。これは 2 枚の画像を合成する際の全体の計算量が $O(n \log n)$ となることに起因する。

表 1. 合成処理時間

	入力画像の幅	入力画像の高さ	提案法の処理時間[s]
ケース 1	448	336	21.7391
ケース 2	640	480	23.3579
ケース 3	1024	768	22.2624
ケース 4	1600	1200	26.8014

4. まとめ

合成の方法として、SIFT アルゴリズムを使用した環境地図作成方法を提案した。SIFT アルゴリズムを活用することで、照明変化等にも比較的強い合成アルゴリズムとなっている。また、今回作成したソフトウェアでは、入力画像が他の入力画像に対して回転している場合には対応していないが、合成位置の探索において回転方向の探索を付け加えることで、対応することが可能である。実際の応用としては、天井、床の外観検査や、複数台の自律移動型ロボットによる環境地図の作成などが挙げられる。今後は共同開発研究などを通して、中小企業との連携を図って行きたい。

(平成 21 年 7 月 2 日受付, 平成 21 年 9 月 8 日再受付)

文 献

- (1) D. G. Lowe, "Object recognition from local scale-invariant features", Proc. of IEEE International Conference on Computer Vision (ICCV), pp.1150-1157, 1999.
- (2) 藤吉弘亘, "Gradient ベースの特徴抽出 -SIFT と HOG -", 情報処理学会研究報告 CVIM 160, pp. 211-224, 2007
- (3) Matthew Brown, "AutoStitch." AutoStitch. 2009-06-10.
<http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html>, (参照 2009-11-26)
- (4) 株式会社ジャングル, "ArcSoft Panorama Maker 4 Pro." ARCSOFT-JUNGLE.JP.
<http://www.arcssoft-jungle.jp/products/pm/>, (参照 2009-11-26)