

レンティキュラポートレート撮影システムの作製

Production of a lenticular portrait photography system

○中庭健太郎¹, 吉川浩², 山口健²*Kentaro Nakaniwa¹, Hiroshi Yoshikawa², Takeshi Yamaguchi²

Abstract: The lenticular, by viewing the parallax images in the right and left eyes, it is an image that can obtain the 3D effects. In our previous study, we had proposed the lenticular portrait system with four web cameras. In this paper, to increase the 3D effects, we have developed the lenticular portrait system with five web cameras.

1. まえがき

レンティキュラ^[1]とは、左右の目で視差画像を見ることによって、立体感を得ることができる画像である。本研究では、USB カメラから画像を取得し、レンティキュラ画像を作製するシステムを開発する。従来のシステム^[1]では、4 台の USB カメラから画像の取得を行っている。これに対し本研究で作製するシステムでは、5 台の USB カメラを使用して撮影を行う。カメラの台数を増やすことで立体感の向上や、画像の変化を滑らかにすることができる。さらに、画像を重ね合わせる際には SIFT^[2]と SURF^[3]という特徴点抽出用のアルゴリズムを使用する。従来のシステムでは SURF を用いて特徴点抽出を行っていた。しかし、従来の SURF のプログラムは煩雑で処理速度が遅くなる原因となっているため、これを改善する。

2. レンティキュラ画像の仕組み

レンティキュラの仕組みを Fig. 1 に示す。レンティキュラとは、数枚の画像を縦に細かく分割し、交互に並べたレンティキュラ画像に半円筒状のレンズを張り付けたものである。半円筒状のレンズによって、右目と左目で視差画像を見ることができ、立体感を得ることができる。また、角度を変える事で画像が切り替わる特性を用いて、アニメーションすることも可能である。

3. システム概要

Fig. 2 にシステムのフォームを示す。始めにカメラから画像を取得する。取得した画像を用いてレンティキュラ画像を作製するための視差画像を作製する。作製した視差画像の位置調整を行い重ね合わせる際、最初、SIFT か SURF を用いて特徴点の抽出を行う。これらは、特徴点の検出と特徴量の算出を行うアルゴリズムである。次に、各特徴点の同一点を取得し、その同一点を基準に画像を重ね合わせる。重ね合わせることでレンティキュラ画像を作製する。フォームの

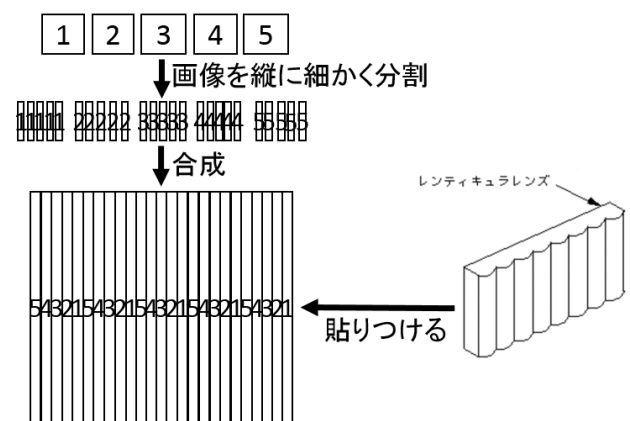


Figure 1. Mechanism of lenticular



Figure 2. Form

①に取得した画像、②にレンティキュラ画像を表示する。作製したレンティキュラ画像は、指定したフォルダに保存する。

4. スレッド処理

本システムでは、5 台のカメラから画像を取得しているためシングルスレッドで処理を行うと、1 つのカメラの動作が完了してから次のカメラの動作に移るため動作が重くなってしまう。そこで、マルチスレッドを用いて 5 台のカメラの処理を並行して行う。このことにより単位時間あたりにカメラから取得できる画像の数を増やす。

5. 特徴点の抽出

Table 1 に SIFT と SURF の比較を示す。本システムでは、SIFT と SURF を使って各画像に対して特徴点の抽出を行う。Table 1 から、SIFT の方が抽出した特徴点数は多く、処理速度においては、SURF の方が速いことが分かる。この結果は 2 つのアルゴリズムの性質通りとなっている。

6. FlannBasedMatcher と BFMatcher

Table 2 に FlannBasedMatcher^[4]と BFMatcher^[4]の比較を示す。本システムでは、BFMatcher と FlannBasedMatcher という方法を用いて同一点取得を行う。BFMatcher は総当たりで、FlannBasedMatcher は最近傍探索でマッチする特徴量を探している。Table 2 から、処理精度において違いは見られないが、処理速度では FlannBasedMatcher の方が速いことが分かる。これは、最近傍探索により処理効率を上げることで処理速度が向上しているためであると考えられる。

7. 同一点取得と画像合成

従来のシステムでは、同一点取得の際に動作が重くなる処理がいくつか存在した。主な要因として、基準となる画像の特徴点抽出を 3 回他の画像と比較する度に行っていること。また、同一点取得のループ処理の設定で、無駄な条件検索を行っていることが挙げられる。そのため、従来のシステムでは処理に時間のかかる SIFT は使用されなかった。

Fig. 3 に SIFT と SURF による同一点の取得結果を示す。本システムでは、同一点の取得に際し、Fig. 3 の左上の画像を基準とし、比較対象の画像それぞれとのマッチング処理を行う。さらに、特徴点同士の距離が長いものを排除することで同一点取得の精度を高める。この結果、従来のシステムでは SURF を用いて 3.7[s]かかっていたのに対し、本システムでは、SURF の場合は 1.6[s]で従来のシステムを上回った。SIFT の場合も 3.1[s]で処理が行われ、実用できる処理速度での動作が可能となった。このことから、本システムでは 2 つの機能を実装する。基本的には SURF を使い、特徴点の抽出が難しい画像に対しては SIFT を使用する。Fig. 3 は SIFT と SURF により取得した同一点が赤枠で表示されており、SIFT を用いた処理の方がより多くの同一点が得られていることが分かる。Fig. 3 の左上の画像内の赤枠をクリックすることでその点を基準に画像が合成される。Fig. 4 に同一点による画像合成の流れを示す。

8. むすび

本報告では、レンティキュラ画像の撮影システムの作製を行った。カメラ台数を増やしたが、マッチング

Table 1. Comparison of SIFT and SURF

Algorithm	Feature points	Processing speed[s]
SIFT	905	0.15
SURF	574	0.05

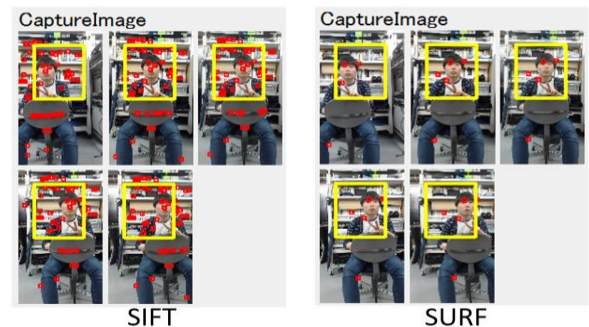


Figure 3. Acquisition result of the identical point

Table 2. Comparison of BFMatcher and FlannBasedMatcher

Identical point acquisition way	Identical points	Processing speed[s]
BFMatcher(SIFT)	826	0.06
FlannBasedMatcher(SIFT)	826	0.03
BFMatcher(SURF)	492	0.04
FlannBasedMatcher(SURF)	492	0.01

この点を基準に合成

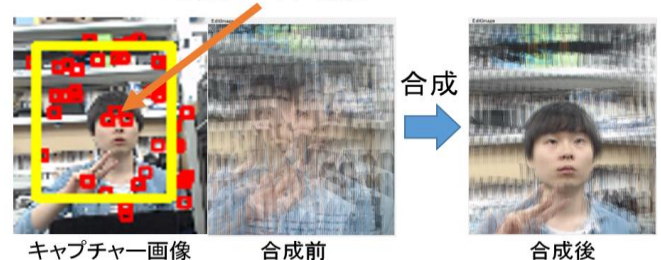


Figure 4. Image synthesis of an identical point

処理のプログラムを改善することで、SIFT と SURF のどちらを使用した場合においても、従来のシステムより処理速度を速くすることが出来た。

参考文献

- [1] 水野将太：「レンティキュラの立体感向上と撮影ソフトの自動化」，日本大学理工学部電子情報工学科卒業論文，2010
- [2] David G. Lowe : "Object Recognition from Local Scale-Invariant Features", Proc. of the International Conference on Computer Vision, Corfu Sept. 1999.
- [3] Herbert Bay, Tinne Tuytelaars, Luc Van Gool : "SURF : Speeded Up Robust Features", computer vision-ECCV Lecture Notes in Computer Science, 2006
- [4] BFMatcher, FlannBasedMatcher
<http://opencv.jp/>