SITT - A Simple Robust Scaleinvariant Text Feature Detector for Document Mosaicing

Marco Block, Marte Ramírez Ortegón, Alexander Seibert, Jan Kretzschmar, Raúl Rojas
Dept. of Mathematics and Computer Science
Free University of Berlin, Germany
[block|marte|seibert|jkretsch|rojas]@inf.fu-berlin.de

January 2007

***Abstract*— In this paper we present SITT, a simple robust scaleinvariant text feature detector for document mosaicing. Digital image stitching has been studied for several decades. SIFT-Features in combination with RANSAC algorithm are established to produce good panoramas. The main problem of realtime text document stitching is the size of the feature set created by SIFT-Features. We introduce SITT-Features to solve this problem. Our experiments denote that for document images SITT-Features produce faster good results than SIFT-Features.**

I. Introduction And Related Work

Our goal is to preprocess several photos (i.e. from a newspaper), taken by a normal digital camera, in this way, that after the images are stitched together and using an OCR, the identification rate is maximized. Digital image stitching has been studied for several decades and hundreds of paper have been written. There exist two different classes of algorithms. Picture-based methods (direct methods) using the hole picture information for minimizing the pixel-to-pixel dissimilarities ([6] gives an overview). Feature-based methods extract at first distinctive image features and afterwards match the point sets using geometric relationship, i.e. RANSAC [4]. Feature-based approaches have the advantage in comparison with picture-based methods of beeing more robust against scene movement and are potentially faster [6].

To recognise panoramas from several pictures, SIFT-Features [2] are established and in combination with the RANSAC algorithm it is a good and fast way to produce panoramas [1]. The modified PCA-SIFT-Features improves the original features [3] but applied to special case document images there are flaccidities (fig. 1).

As we see in fig. 1 document images produces a lot keypoints. Therefor, RANSAC works slowly.

Fig. 1. Document image (1073*x*776 pixel) produces 16741 PCA-SIFT-Features.

In this paper, we present SITT-Features which are more dedicated and natural than SIFT-Features (fig. 2) on document images.

After the text detection and warping, we try to identificate points. For noise handling, we use a heuristic to reduce the effort, afterwards we classify special features. Then, we use RANSAC for match the point sets.

The feature extraction process is easy to implement and fast. First results are promising.

This paper is organized as follow: In part one we present the related work and introduce this topic. In the second part we describe the extraction of the SITT-Features and the hole stitching process in part three. Results and perspective are discussed in part four.

II. Text Feature Extraction

The first preprocess step is to remove perspective distortion and rotation [7] in the original images

Fig. 2. Document image (1073x776 pixel) produces 324 SITT-Features.

using local binarization [5]. In result, both pictures differ by translation and scale.

Now, we try to describe natural text elements in various size found in all textpages, instead of compute gradients on the gaussian pyramid [2]. The simple idea is to find text points like **'i'**, **'.'** , **':'**, **'ü'**, **'!'**, **'?'**, **...** The reason why it works is the assumption that the given document image is in such good quality that it is possible to recognize text with an OCR. To find the points we uses weighted templates in various size $t_3, t_4, \ldots, t_{13}$ (fig. 3, 4) and match them against the images (fig. 5). So the scaleinvariant property of the feature moves to the template.

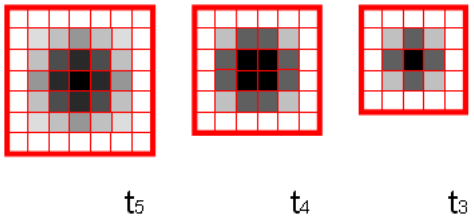

Fig. 3. Point templates with various size.

Pseudocode to create the weighted templates (r the template index, i and j the matrix index):

```
for r=3 to 13 do
  for i=1 to (r-1)/2 do
    for j=1 to (r-1)/2 do
      t_r[i,j]=t_r[i,r-j-1]=
      t_r[r-i-1,j]=t_r[r-i-1,r-j-1]=
      2r^2-((r-1)/2-i)^2-((r-1)/2-j)^2
```

A. Point Feature Extraction

To advance the performance we calculate every template size in one step.

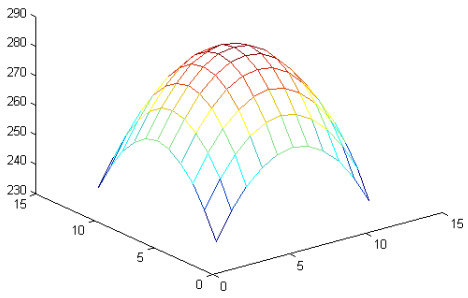

Fig. 4. The point templates are weighted.

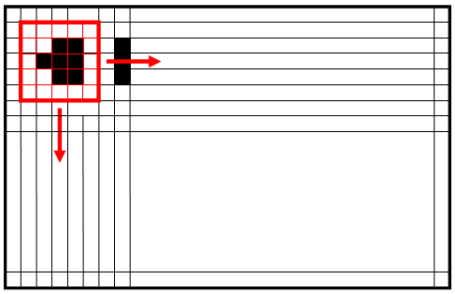

Fig. 5. Template matching in an image (using template t_3).

```
for all valid positions in the image
  for r=3 to 13 do
    if (zero-frame exist) do
      w = score of the inner weighted template
      c = count of non-zero elements in template
      if (w>r^3 && c>=(r-2)^2-3r)
        SITT-Feature found
    end if
```

If one template match, we save the center coordinates and the template size. The current SITT-Features applied to a newspaper (fig. 6).

It is apparent, that document parts with images, produces a lot of features. Compared to text parts we see that the appearance of features has a small density.

B. Noise and Image Handling

Hence, we use this property to identificate the unwanted features (fig. 7) recursivly. This method depends on the neighborhood distance.

The extant features showing interesting properties (fig. 8). In almost every document, there exist feature with special properties that help us to mark similarities. These properties can be special distances among each other (like components: additional german characters, colons or semicolons) and the individual size.

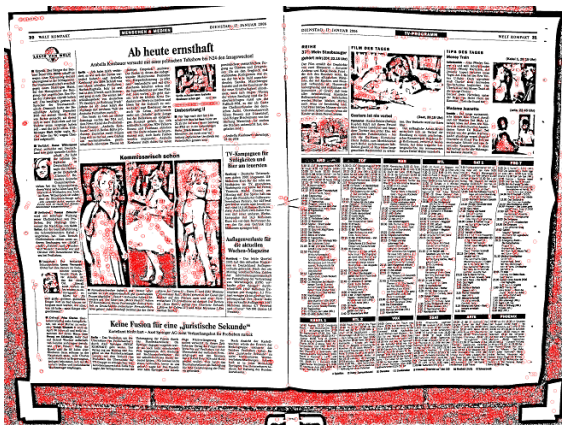

Fig. 6. Potential SITT Features applied to a newspaper, after the extraction step. Pictures and borders produces areas of features.

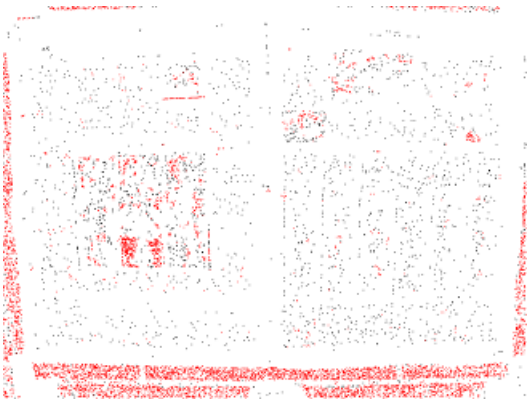

Fig. 7. The unwanted features (red points) in photos or images are identificated against her density. In this example, we reduce the size of the feature set from 8035 to 2469.

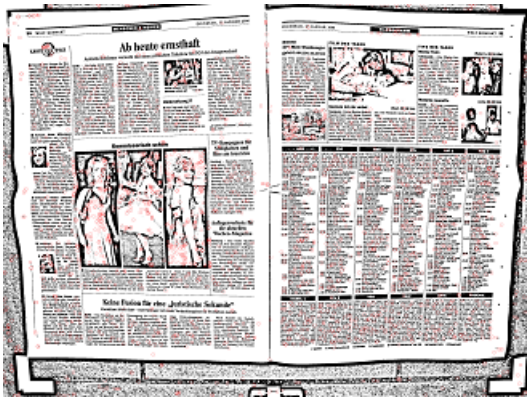

Fig. 8. Part of a image and the corresponding SITT-Features. We can observe, that the distribution is rar in text, but i.e. some special point-pairs are detected.

C. Point Classifier

To near points depends on the size, we expect characters like: **':'**, **';'**, **'ä'**, **'ü'** or **'ö'**. We classify this kinds of points and use this property later for effectifly finding the right transformation (fig. 9). The advantage of this features are the exact position.

```
for all points i
  n_i=list of next neighbors to i
  r=size of template (saved when i was found)
  if (n_i[0]<5*r-2)
    point i is a special character
    like: ':', ';', 'ä', 'ü' or 'ö'
  else
    point i is a single point
```

Fig. 9. SITT-Features are green and detected special points red.

Method [7] delivers the orientation and the textline positions. We see good possibilities to extend this approach, f.e. an additional criterion to differ points, could be the relative position of a point in dependence to its identify textline (to solve the „vice versa“-problem[1]).

III. Stitching Process

To stitch the images together, we use selective image blending [8]. Usually at the end of the panorama process stands the blurring of the common part of the images. This method deliver good panoramas of natural scenery (fig. 10).

Applied to text documents, this technic delivers not satisfiing results. In consequence, we have to avoid this theoretical natural approach for this kind of problems (fig. 11).

With selective image blending we are able to maintain the words. According to the contrast beetween the involved images, we take different parts into the resultimage.

IV. Experiments And Perspective

To compare the approaches (SIFT-Features versus SITT-Features), we prepare a database of well known digital documents. After printing the documents, we take pictures from several positions

[1]Means, that the text document is 180° rotated.

Fig. 10. Panorama of natural scenery. The overlapping image parts looking very natural (the example images are from the autostitch homepage [12]).

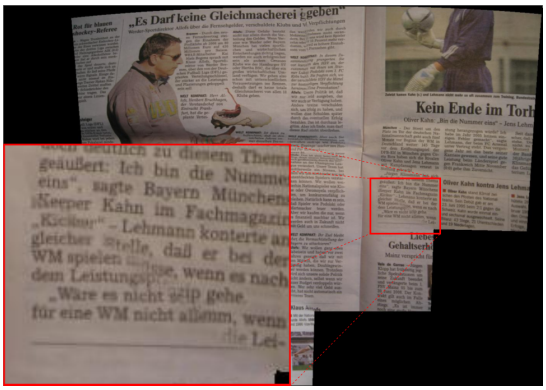

Fig. 11. Panorama of text documents. A zoom into the overlapping part shows the problem.

and distinctive parts. Following, we stitch them together. The resultimage will be recognise by the OCR Abbyy FineReader [11]. Now, we compare the original text with the photo stitched image using the Needleman-Wunsch-Algorithm [10].

The quality are quite similar, but the consumption of time fore the feature extraction with following RANSAC is a little bit more convenient with SITT.

SITT-Features seems to be the more natural approach for using in realtime systems. The perspective could be a realtime system for visual impaired persons for detecting documents and prepare them for the OCR. With image stitching we are able to deal with large documents and in connection with selective image blending [8] we realise quality document mosaicing.

Fig. 12. Document mosaic using SITT and RANSAC.

REFERENCES

[1] Brown M, Lowe D.G.: *„Recognising Panoramas“*, ICCV 2003, pages 1218-1225

[2] Lowe D.G.: *„Distinctive image features from scale-invariant keypoints“,* International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

[3] Ke Y., Sukthankar R.: *„PCA-SIFT: A More Distinctive Representation for Local Image Descriptors“*, CVPR (2) 2004: 506-513

[4] Fischler M.A., Bolles R.C.: *„Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.“,* Commun. ACM 24(6): 381-395 (1981)

[5] Ramírez M., Block M., Rojas R.: *„New Robust Binarization Approach in Letters“*, Guadalajara México, CONCIBE 2006

[6] Szeliski R.: *„Image alignment and stitching: A tutorial“*, MSR-TR-2004-92, Microsoft Research, 2004

[7] Liang J., DeMenthon D., Doermann D.: „*Flattening Curved Documents in Images*“, CVPR 2005

[8] Liang J., DeMenthon D.: „*Camera-Based Document Image Mosaicing*“, ICPR 2006

[9] Zappalá A., Gee A., Taylor M.: „Document mosaicing“, Image and Vision Computing(17):589-595, 1999

[10] Needleman S., Wunsch C.: “*A general method applicable to the search for similarities in the amino acid sequence of two proteins*“, JMolBiol. 48(3):443-453, 1970

[11] Abbyy: *http://www.abbyy.de/*

[12] Autostitch: *http://www.cs.ubc.ca/~mbrown/autostitch/*