Grouping of Semantically Similar Image Positions

Lutz Priese, Frank Schmitt, and Nils Hering

Institute for Computational Visualistics,
University Koblenz-Landau, Koblenz
{priese,fschmitt,nilshering}@uni-koblenz.de

Abstract. Features from the Scale Invariant Feature Transformation (SIFT) are widely used for matching between spatially or temporally displaced images. Recently a topology on the SIFT features of a single image has been introduced where features of a similar semantics are close in this topology. We continue this work and present a technique to automatically detect groups of SIFT positions in a single image where all points of one group possess a similar semantics. The proposed method borrows ideas and techniques from the Color-Structure-Code segmentation method and does not require any user intervention.

Keywords: Image analysis, segmentation, semantics, SIFT.

1 Introduction

Let I be a 2-dimensional image. We regard I as a mapping $I : Loc \to Val$ that maps coordinates (x, y) from Loc (usually $Loc = [0, N-1] \times [0, M-1]$) to values $I(x, y)$ in Val (usually $Val = [0, 2^n[$ or $Val = [0, 2^n[^3$). We present a new technique to automatically detect groups $G_1, ..., G_l$ of coordinates, i.e., $G_i \subseteq Loc$, where all coordinates in a single group represent positions of a similar semantics in I. Take, e.g., an image of a building with trees. We are searching for sets $G_1, ..., G_l$ of coordinates with different semantics. E.g., there shall be coordinates for crossbars in windows in some set G_i, for window panes in another set G_j, inside the trees in a third set G_k, etc.. G_i, G_j, G_k form three different semantic classes (for crossbars, panes, trees in this example) for some i, j, k with $1 \leq i, j, k \leq l$. Obviously, such an automatic grouping of semantics can be an important step in many image analysis applications and is a rather ambitious programme. In this paper we propose a solution for SIFT features. Our technique is based on ideas from the CSC segmentation method.

2 SIFT

SIFT (Scale Invariant Feature Transformation) is an algorithm for an extraction of "interesting" image points, the so called SIFT features. SIFT was developed by David Lowe, see [2] and [3]. The SIFT algorithm follows the scale space approach

A.-B. Salberg, J.Y. Hardeberg, and R. Jenssen (Eds.): SCIA 2009, LNCS 5575, pp. 726–734, 2009.

and computes scale- and orientation-invariant points of interest in images. SIFT features consist of a coordinate in the image, a scale, a main orientation, and a 128-dimensional description vector. SIFT is commonly used for matching objects between spatially (e.g. in stereo vision) or temporally displaced images. It may also be used for object recognition where in a data base characteristic classes of features of known objects are stored and features from an image are matched with this data base to detect objects.

Slot and Kim use class keynotes of SIFT features in [5] for object class detection. Those class keynotes have been found by a clustering of similar features. They use spatial locations, orientations and scales as similarity criteria to cluster the features. The regions in which the clustering takes place (the spatial locations) are selected manually. In those regions clusters are built by a grouping via a low variance criteria in scale orientation space.

Mathematically speaking, a SIFT feature f is a tuple $f = (l_f, s_f, o_f, v_f)$ of four attributes: l_f for the location of the feature in x,y-coordinates in the image, s_f for the scale, o_f for the main orientation, v_f for the 128-dimensional vector. The range of o_f is $[0, 2\pi[$. The range of s_f depends on the size of the image and is about $0 \leq i \leq 100$ in our examples. The Euclidean distance $d_E(f, f')$ of two SIFT features f, f' is simply the Euclidean distance between the two 128-dimensional vectors v_f and $v_{f'}$.

3 CSC

Let $I : Loc \rightarrow Val$ be some image. A *region* R in I is a connected set of pixels of I. Connected means that any two pixels in R may be connected by a path of neighbored pixels that will not leave R. A region R is called a *segment* if in addition all pixels in R possess similar values in Val. A *segmentation* $\mathcal{S}$ is a partition $\mathcal{S} = \{S_1, ..., S_k\}$ with

1. $I = S_1 \cup ... \cup S_k$,
2. $S_i \cap S_j = \emptyset$ for $1 \leq i \neq j \leq k$,
3. each $S_i \in \mathcal{S}$ is a segment of I.

$\mathcal{S}$ is a *semi segmentation* if only 1 and 3 hold.

The CSC (Color Structure Code) is a rather elaborated region growing segmentation technique with a merge phase first and a split phase after that. It was developed by Priese and Rehrmann [4]. The algorithm is logically steered by an overlapping hexagonal topology. In the merge phase two already constructed overlapping segments S_1, S_2 of some level n may be merged into one new segment if S_1 and S_2 are similar enough. Otherwise, the overlap $S_1 \cap S_2$ is split between S_1 and S_2. In region growing algorithms without overlapping structures two similar segments with a common border may be merged. However, possessing a common substructure $S_1 \cap S_2$ leads to much robuster results than merging in case of a common border. Although the CSC gives a segmentation it operates with semi segmentations on different scales.

We will exploit the idea of merging overlapping sets for a segmentation in the following for a grouping of semantics.

Fig. 1. Euclidean distances between appropriate and external features

4 A Topology on SIFT Features

To group semantically similar SIFT features we are looking for a topology where those semantically similar features become neighbors. Unfortunately, the Euclidean distance gives no such topology. Two SIFT features f_1, f_2 of the same image I with a very similar semantics may possess a rather large Euclidean distance $d_E(f_1, f_2)$ while for a third SIFT feature f_3 with a very different semantics $d_E(f_1, f_3) < d_E(f_1, f_2)$ may hold, compare Fig. 1. Thus, the Euclidean distance is not the optimal measure for the semantic distance of SIFT features. To overcome this problem we have introduced a new topology $\mathcal{T}$ on SIFT features in [1]. A *7-distance* $d_7(f, f')$ between f and f' has been introduced as the sum of the seven largest values of the 128 differences in $|v_f - v_{f'}|$. Let $f = (l, s, o, v)$ be some SIFT feature and let $f_i = (l_i, s_i, o_i, v_i)$ denote the i-th closest SIFT feature to f in the image with respect to d_E. For some set N of SIFT features we denote by μ_N^s (μ_N^o) the mean value of N in the coordinate for scale (orientation). The following algorithm computes a neighborhood $N(f)$ for f with three thresholds t_s, t_o, t_v by:

$N :=$ *empty list*; *insert* f *into* N; $i := 0$; $fault := 0$;
repeat
  $i := i + 1$;
  if $|(s, o, v) - (s_i, o_i, v_i)| \leq (t_s, t_o, t_v)$ and ($\mu_N^s \leq 0.75$ or $|s - s_i| \leq 2 \cdot \mu_N^s$)
  and ($\mu_N^o \leq 0.01$ or $|o - o_i| \leq 5 \cdot \mu_N^o$)
    then *insert* f_i *into* N; *update* μ_N^s *and* μ_N^o
    else $fault := fault + 1$
until $fault = 3$.

Thus, the Euclidean distance gives candidates f_i for $N(f)$ and the 7-distance excludes some of them. This semantic neighborhood defines a topology $\mathcal{T}$ on

SIFT features where the location of the SIFT features in the image plays no role.

5 Grouping of Semantics

5.1 The Problem

We want a grouping of the locations of SIFT features with the "same" semantics. The obvious approach is to group the SIFT features themselves and not their locations. Thus, the first task is:

Let $\mathcal{F}_I$ be the set of all SIFT features in a single image I detected by the SIFT algorithm. Find a partition $\mathcal{G} = \{G_1, ..., G_l\}$ of $\mathcal{F}_I$ s.t.

1. $\mathcal{F}_I = G_1 \cup ... \cup G_l$,
2. l is rather small, and
3. G_i consists of SIFT features of a similar semantics, for $1 \leq i \leq l$.

Each $G \in \mathcal{G}$ represents one semantic class. We do not claim that $G_i \cap G_j = \emptyset$ holds for $G_i \neq G_j$.

$loc(\mathcal{G}) := \{loc(G) | G \in \mathcal{G}\}$ becomes the wanted grouping of locations of a similar semantics in I where $loc(G)$ is the set of all positions of the features in G.

The topology $\mathcal{T}$ was designed to approach this task. All features inside a neighborhood $N(f)$ are usually of the same semantics as f. Let T_C be a known set of all SIFT features with a common semantics C as a ground truth and suppose f, f' are two features in T_C. Unfortunately, in general $N(f) \neq N(f')$ and $N(f) \neq T_C$ holds. $N(f)$ is usually smaller than T_C and may sometimes contain features not in T_C at all. Thus, computing $N(f)$ does not solve our task but will be the initial step towards a solution.

5.2 The Solution

One may imagine $\mathcal{F}_I$ as some sparse image $\mathcal{F}_I : Loc \rightarrow \mathbb{R}^{130}$ into a high dimensional value space with

$$\mathcal{F}_I(p) = \begin{cases} (s_f, o_f, v_f) & : \text{ for some } f \in \mathcal{F}_I \text{ with } l_f = p, \\ \text{undefined} & : \text{ if } \nexists f \in \mathcal{F}_I \text{ with } l_f = p. \end{cases}$$

Thus, the task of grouping semantics is similar to the task of computing a semi segmentation. The main difference is that $\mathcal{F}_I$ is rather sparse and connectivity of a segment plays no role. As a consequence, a region in $\mathcal{F}_I$ is simply any subset of $\mathcal{F}_I$ and a segment in $\mathcal{F}_I$ is a subset of features of $\mathcal{F}_I$ with a pairwise similar semantics. We will devise the segmentation technique CSC into a grouping algorithm for sparse images.

In a first step $N(f)$ is computed for any SIFT feature f in the image. $\mathcal{N} := \{N(f) | f \in \mathcal{F}_I\}$ is a semi segmentation of $\mathcal{F}_I$. However, there are too many overlapping segments in $\mathcal{N}$. $\mathcal{N}$ serves just as an initial grouping.

In the main step overlapping groups G, G' will be merged if they are similar enough. Here similarity is measured by the overlap rate $\frac{|G \cap G'|}{min(|G|,|G'|)}$. In contrast to the CSC we do not apply a split phase where $G \cap G'$ becomes distributed between G and G' in case that G and G' are not similar enough to be merged. The reason is that the rare cases where a SIFT feature is put into several semantic classes may be of interest for the following image analysis. In short, our algorithm AGS (**A**utomatic **G**rouping of **S**emantics) may be described as:

$\mathcal{H} := \mathcal{N}$;
(1) $\mathcal{G}$:= *empty list*;
for $0 \leq i < |\mathcal{H}|$ do $G := \mathcal{H}[i]$;
  for $0 \leq j < |\mathcal{H}|, i \neq j$ do
    if $G = \mathcal{H}[j]$
      then *remove* $\mathcal{H}[j]$ *from* $\mathcal{H}$
      else if G *and* $\mathcal{H}[j]$ *are similar* then $G := G \cup \mathcal{H}[j]$
  end for;
  insert G *into* $\mathcal{G}$
end for;
if $\mathcal{H} \neq \mathcal{G}$ then $\mathcal{H} := \mathcal{G}$; goto line (1) else end.

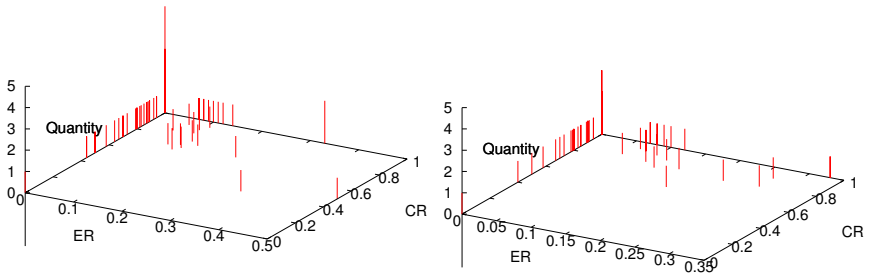
6 Some Examples

We present some pairs of images (Fig. 3 to 7) in the Appendix where the AGS algorithm has been applied. The left images show the coordinates of all features as detected by the SIFT algorithm. In a few cases two features with different scale or main orientation may be present at the same coordinate. The right ones show locations of some groups as computed by AGS. All features of one group are marked by the same symbol. Only groups consisting of at least five features are regarded in those examples. The number of such groups found by the AGS are given in #group and the semantics of the presented groups is named. Obviously, the results of this version of the AGS depend highly on the results of SIFT (as AGS regards solely detected SIFT features). The following qualitative observations are typical: The AGS algorithm works well on images with many symmetric edges (as in images of buildings). However, the quality is not good on very heterogeneous images with only very few symmetric edges (as in Fig. 5 where only one group with more than four elements is detected). In images with a larger crowd of people the AGS failed, e.g., to group features inside human faces.

7 Quantitative Evaluation

7.1 SIFT

Let $\mathcal{G} = \{G_1, ..., G_n\}$ be the set of SIFT features groups as computed by the AGS. Let $L_i := loc(G_i)$. Thus, $loc(\mathcal{G}) = \{L_1, ..., L_n\}$ is the found grouping



(a) SIFT (b) SIFT$_{noi}$

Fig. 2. Distribution of CR and ER

of locations of the same semantics. We now present a quantitative evaluation of $loc(\mathcal{G})$. We have manually annotated the SIFT locations for some semantic classes $(C_1, ..., C_n)$ in a set A of images as a ground truth. Let GT_i be the annotated ground truth for one semantic class C_i. Our evaluation tool computes the semantic grouping $\mathcal{G}$ of the AGS and compares each L in $loc(\mathcal{G})$ with GT_i by an

- coverability rate $CR(L, GT_i) := \frac{|L \cap GT_i|}{|GT_i|}$, and
- error rate $ER(L, GT_i) := \frac{|L - GT_i|}{|L|}$.

At the moment we have annotated the semantics "crossbar", "lower pane left" and "lower pane right" in windows to the corresponding feature positions in twenty-five images with buildings. This gives three sets of ground truth features, namely GT_1 = Crossbar, GT_2 = PaneLeft and GT_3 = PaneRight.

For each image and each ground truth GT_i, $1 \leq i \leq 3$, we choose the group L in $loc(\mathcal{G})$ with the highest coverability rate $CR(L, GT_i)$. We show mean and standard deviation of the coverability and error rate over all three groups and all 25 images in table 1a. Figure 2a shows graphically the distribution of CR and ER over the 25×3 ground truth feature sets. The chosen parameters for $N(f)$ are $t_o = 0.5$, $t_s = 2.0$, $t_v = 500$ and the overlap rate for similarity of two groups in the AGS has been set to 0.75. Only groups with at least two members have been regarded.

In one of the 25 images there are only two windows whose crossbar features are not grouped. A single mistake in such small groups gives high errors rates.

Table 1. Evaluation of AGS algorithm on 25 manually annotated images

(a) Evaluation Lowe-SIFT

	CR	ER
mean	0.8589	0.0504
standard deviation	0.1951	0.0939

(b) Evaluation SIFT$_{noi}$

	CR	ER
mean	0.8939	0.0411
standard deviation	0.166	0.079

This explains the bad results in some images in figure2a. However, even this simple version of AGS gives good results in our analysis of the semantic classes "crossbar", "lower pane left" and "lower pane right". On average, the locations $loc(G)$ of the best matching group G for one of those classes covers 86% of all semantic positions of that class with an average error rate of 5%, see table 1a.

7.2 SIFT$_{noi}$

As we are searching for objects with a similar semantics in a single image those objects should possess the same orientation, at least in our application scenario of buildings. Thus, the orientation invariance of SIFT is even unwanted here. We therefore have implemented a variant SIFT$_{noi}$ - *noi* stands for **n**o **o**rientation **i**nvariance - where the orientation normalization in the SIFT algorithm is skipped. As a consequence, the main orientation o_f plays no role and the algorithm for $N(f)$ has to be adopted, ignoring o_f and the threshold t_o. We have further changed the parameter t_v to 450 for SIFT$_{noi}$. The results of our AGS with this SIFT$_{noi}$ variant are slightly better and shown in table 1b and figure 2b. The mean of the coverability rate increases to 89% while at the same time the error rate decreases to 4%.

8 Résumé

We have presented a completely automatic approach to the detection of groups of image positions with similar semantics. Obviously, such a grouping is helpful in many image analysis tasks.

This work is by no means completed. There are many variants of the AGS algorithm worth to be studied. One may modify the computation of $N(f)$ for a feature f. To decrease the error rate, a kind of splitting phase should be tested where in case of a high overlap rate between two groups G, G' the union $G \cup G'$ may be refined by starting with $G'' := G \cap G'$ and adding to G'' only those features in $(G \cup G') - G''$ that are "similar" enough to G''. The AGS method presented in this paper uses Lowe-SIFT features and a novel variant of SIFT$_{noi}$ features without orientation invariance. AGS works well in images with many symmetries – as in the examples with buildings – but less good in chaotic images. This is mainly caused by the fact that both SIFT features are designed to react on symmetries. Therefore, a next task is the extension of AGS to other feature classes and combinations of different feature classes.

References

1. Hering, N., Schmitt, F., Priese, L.: Image understanding using self-similar sift features. In: International Conference on Computer Vision Theory and Applications (VISAPP), Lisboa, Portugal (to be published, 2009)
2. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision ICCV, Corfu, pp. 1150–1157 (1999)

3. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 20, 91–110 (2003)
4. Rehrmann, V., Priese, L.: Fast and robust segmentation of natural color scenes. In: Chin, R.T., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1351, pp. 598–606. Springer, Heidelberg (1997)
5. Slot, K., Kim, H.: Keypoints derivation for object class detection with sift algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS, vol. 4029, pp. 850–859. Springer, Heidelberg (2006)

Appendix

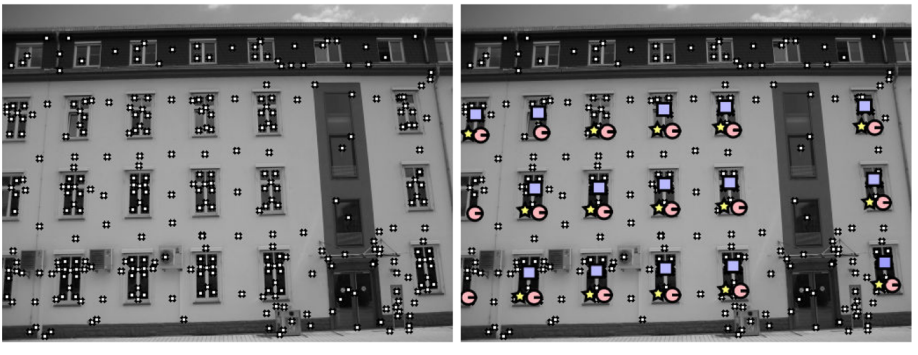


Fig. 3. #group = 10; shown are crossbars, lower right pane, lower left pane

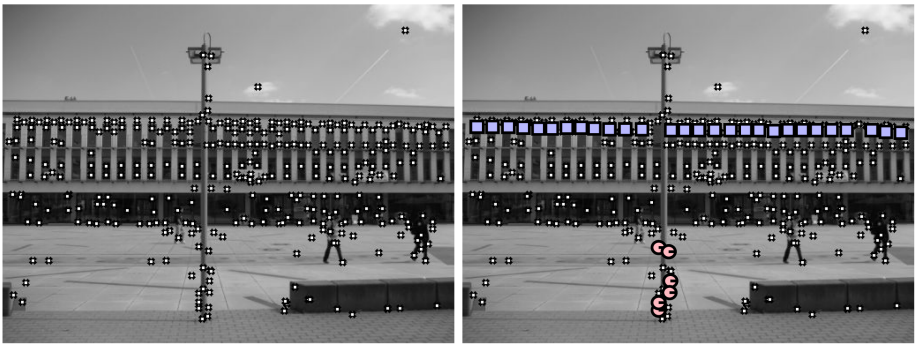


Fig. 4. #group = 21; shown are upper border of pane, lower border of post

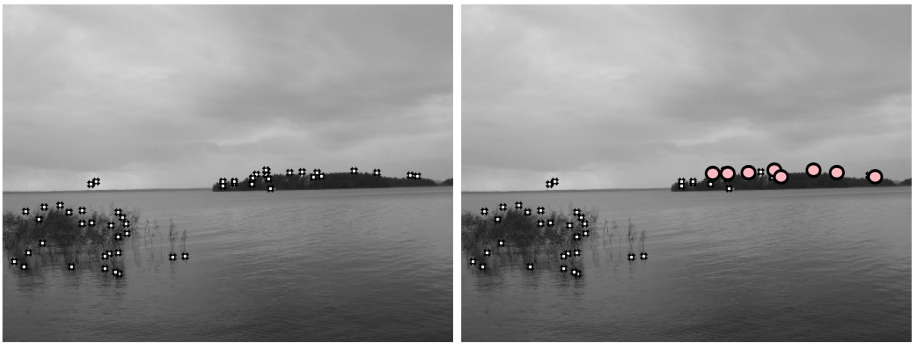


Fig. 5. #group = 1, namely upper border of forest

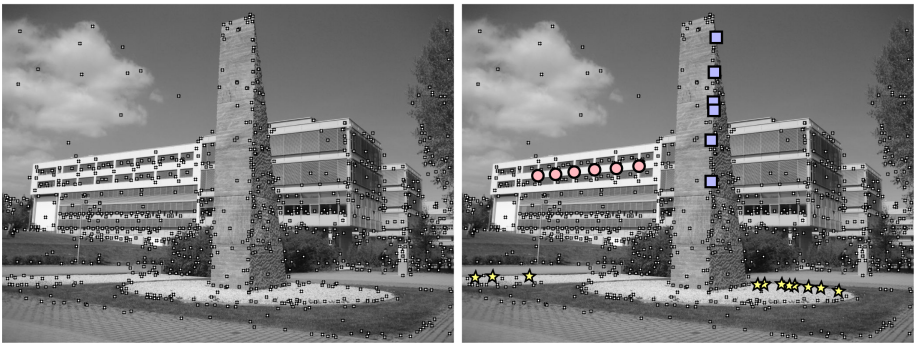


Fig. 6. #group = 24; shown are window interspace, monument edge and grass change

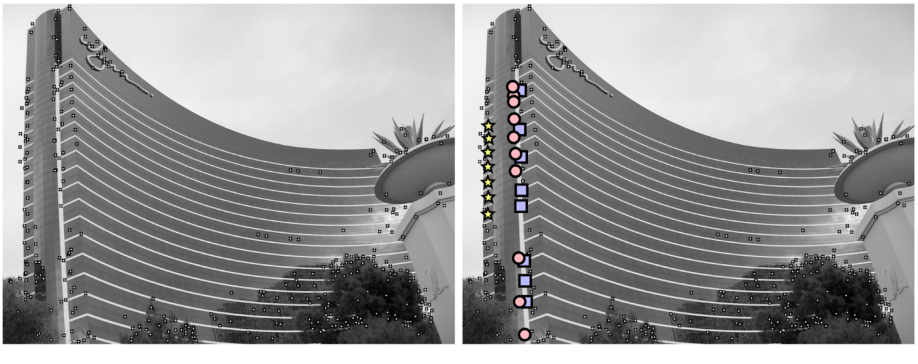


Fig. 7. #group = 7; shown are three different groups of repetitive vertical elements