ESAIM: PROCEEDINGS AND SURVEYS, September 2014, Vol. 45, p. 467-474
J.-S. Dhersin, Editor

DETECTION OF AN IMAGE IN A VIDEO SEQUENCE *, **

Tamara El Bouti[1], Gwenael Mercier[2], Caroline Obrecht[3] and Giuseppe Benedetti[4]

Abstract. The purpose of this work is to discuss possible methods to retrieve a frame in a source video sequence starting from a given image query (for example, a screen shot of a movie scene). In order to do that, we develop and test different approaches that rely on SIFT local descriptors. First, we briefly recall the SIFT algorithm and propose a straightforward way to use it in order to find a few candidate frames which are similar to the query image. Then, we propose an improvement of this method by focusing on the moving regions to filter out the irrelevant features. Finally, we formulate a different way to detect the right candidate using a geometric approach that accounts for the relative position of the points of interest.

Résumé. Le but de ce travail est de discuter des méthodes pour retrouver un frame dans une séquence vidéo en partant d'une image requête (une photo tirée d'un film par exemple). Pour résoudre notre problème, on s'appuiera sur les descripteurs locaux SIFT. Dans un premier temps, nous présenterons brièvement SIFT et proposerons une manière directe de l'utiliser afin de trouver un certain nombre d'images candidates similaires à l'image requête. Dans un deuxième temps, on va améliorer cette méthode en se concentrant sur les régions de mouvement. Finalement, on va proposer une autre manière de détecter les bonnes images candidates en utilisant une approche géométrique qui prend en compte aussi la position des descripteurs locaux.

Introduction

This work is motivated by a problem presented by Technicolor, a company which provides products and services for the entertainment industry. The subject proposed by Patrick Perez, a researcher in this company during the week 'Maths & Entreprises'[1] was the following: how can we detect an exact frame in a movie, starting from a given image query? (typically, a poor-resolution screen shot from the movie found on the Internet). More precisely, Mr. Perez provided an example of a picture and a video sequence extracted from the movie *Children of Men* and asked to find a way to know exactly which frame of the sequence corresponds to the source picture. This was a real challenge, given that the target picture did not come from the same source as the video sequence,

* *We thank Patrick Pérez for this exciting issue*

** *We are very grateful to Simon Masnou and Antonin Chambolle for providing us some ideas to broach this subject.*

[1] LMV, Versailles St Quentin en Yvelines, France; e-mail: tamara.el-bouti@uvsq.fr

[2] CMAP, École polytechnique, France; e-mail: gwenael.mercier@cmap.polytechnique.fr

[3] Laboratoire de Mathématiques de l'Université Paris-Sud, France ; e-mail: caroline.obrecht@u-psud.fr

[4] CREST/CEREMADE, Universié Paris Dauphine, France ; e-mail: beppeben2030@gmail.com

[1] "Semaine d'Etudes Mathématiques et Entreprises (SEME)" in french. At the SEME, young researchers (usually Ph.D. students) in mathematics work in small groups during one week on problems of industrial relevance. The problems are presented by various companies.

© EDP Sciences, SMAI 2014

Article published online by EDP Sciences and available at http://www.esaim-proc.org or http://dx.doi.org/10.1051/proc/201445048

and the two did not have the same resolution nor the same contrast or quality. In this paper, we will base our presentation on another picture taken from the movie *Titanic*, which will be even more difficult to treat because of a higher similarity between frames.

In the past decade, image retrieval and visual search at large scales have experienced an impressive progress, as demonstrated by Google Image search. These algorithms are typically constructed to be invariant to geometric transformations, contrast changes, noise, and changes of brightness. Nonetheless, searching for images in a movie sequence remains a very difficult task because:

- a sequence is a family of images (frames) and even a small sequence is composed of a lot of images (for example 250 frames for a video of 10 sec). As a result, the search for the matching frame has to be performed in very large databases.
- images extracted from the same scene can be very similar, as one can easily imagine. To find the right frame one has to capture the "real", possibly fine differences between the query image and the other images of the scene, and not those which are artificially induced by image transformations as compression or scaling etc.

We assume that the first point has already been solved (as Mr. Perez declared, this is usually the easiest part). To deal with the second one, we will make use of a classical algorithm that detects and matches characteristic image fragments, i.e. the SIFT algorithm.

1. SIFT

The algorithm that we use as a basic building tool to compare the query image with the frames of the movie scene is called SIFT, i.e. Scale-Invariant Feature Transform. This algorithm was developed by David Lowe ([1], [2]) and is widely applied in computer vision[2]. SIFT characterizes and "summarizes" an image by a set of local descriptors which are invariant to image scale and rotation, and which are stable under other image transforms such as affine transformations, changes of viewpoint or illumination. We provide here a brief description of how it works.

The first step of SIFT consists in detecting keypoints (or interest points) of the image. Loosely speaking, an interest point is a point of the image which is surrounded by a region with a rich information content, i.e. a high variation of grey scale values. For example, the area around the interest point might contain corners or edges which delimit objects in the image. Ideally, the presence of the interest point should be stable under image perturbations such as brightness variations, rotation or scale/resolution changes. To detect the keypoints, D. Lowe uses a sophisticated procedure based on the classical difference-of-Gaussians (DOG) method [3]. The DOG method smooths the original image $I(x, y)$ by convolution with the Gaussian kernel

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

and then considers the difference between two close smoothings (n is a constant):

$$D(x, y, \sigma) = (G(x, y, n\sigma) - G(x, y, \sigma)) * I(x, y).$$

Keypoints are defined as extrema in $D(x, y, \sigma)$. Lowe repeats the keypoint detection by DOG at multiple scales of smoothing and at multiple levels of "downsampling" of the original image $I(x, y)$. Keypoints are only accepted if they are extrema in the current level as well as in the levels above and below. This procedure aims at providing the scale invariance of the interest point detection.

The second step of the algorithm consists in computing a local descriptor for each interest point. The size (scale) of the keypoint is already known from the first step. The neighbourhood of every keypoint is rescaled so that every keyzone has the same scale. At this scale, the gradient magnitude and orientation at every point of the keyzone is computed (using simple pixel difference). The orientations are saved, weighted by the gradient

[2]We use here a MATLAB implementation that comes with the VLFEAT library [5]

magnitude (and a circular gaussian kernel in order to limit the influence of too far gradients), in a histogram with 36 bins covering all the possible orientations. A peak in the histogram corresponds to a keypoint orientation (which can be non unique: in that case, several keypoints are created with a different orientation).
At this point, we are left with a collection of keypoints along with their scale and orientation. In the sequel, a scale and rotationally invariant descriptor is computed for the region around each keypoint. First, the keyzones are normalized to the same scale and orientation. Then, a 4×4 grid around the keypoint is considered for each region, so that each cell (i, j) of the grid contains 4×4 points $(A_{ij}^{kl})_{k,l=1}^{4}$ of the keyzone. For each cell (i, j) the gradient magnitude and orientation at every A_{ij}^{kl} is computed and the orientations weighted by magnitude (and a Gaussian kernel) are stored in a 8-bin histogram (using interpolation for non principal directions). Finally, a vector $(u_k) \in \mathbb{R}^8$ is created with u_k being the height of the k^{th} column of the histogram. The SIFT descriptor of the keypoint $U \in \mathbb{R}^{128}$ is nothing but the concatenation of the vectors (u_k) obtained for every cell.
The local descriptors can be used to match keypoints between two images I and J in the following way: first, SIFT computes sets (A_i) and (B_j) of keypoints of I and J. Then, one may consider that two local descriptors A_{i_0} and B_{j_0} match if the euclidian distance between A_{i_0} and B_{j_0} is "much" smaller than the distance between A_{i_0} and any other B_j (Lowe suggests $d(A_{i_0}, B_{j_0}) \leq 0.8 \cdot d(A_{i_0}, B_j)$).

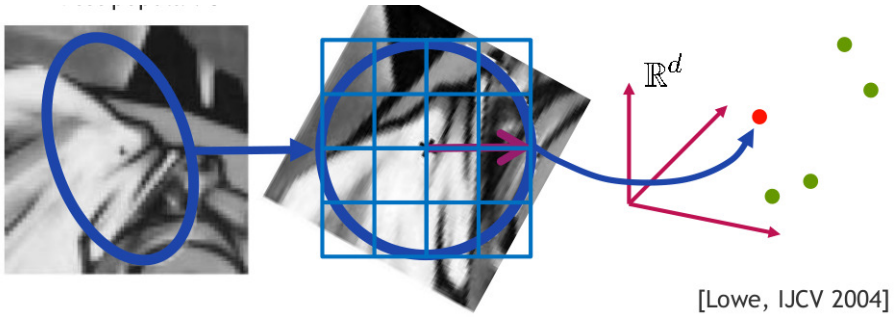

FIGURE 1. Sift detector.

In summary, the whole analysis consists of three steps:

- Detection of characteristic points.
- Computation of SIFT local descriptors for each point.
- Matching of points between different images, based on their descriptors.

2. Searching for Candidates using SIFT

Back to our problem, recall that we are equipped with a sequence of frames extracted from a movie scene, and a query image for which we want to find the corresponding frame in the sequence.
We assume the target sequence consists of N frames and we assume that the frame corresponding to the query image I is a part of the sequence. The goal is to match I with an image in $k \in \{1, ..., N\}$ of the sequence. In order to do so, we start by short-listing a set of $C(< N)$ candidate images, which will then be analyzed with methods that better allow to recover finer differences.
The choice of C should theoretically be done case by case by considering the total number of images N and the degree of similarity. However, our experiments suggest that the choice $C = 10$ is generally a good compromise. Our procedure consists of several steps:

- The SIFT algorithm is applied to the query image I and to each image of the N images of the sequence. We thus obtain, for each image, a set of key points (of variable size) identified by their spatial position and their associated SIFT vector of dimension 128.

- We localize the most relevant $n(k)$ points of interest between I and every image k, $k = 1, ..., N$. This is done thanks to an algorithm already implemented in Matlab (Vlfeat: [5]), which finds matches between the points of interest of two images using the distance between the associated vectors. The number $n(k)$ changes with k because all the images do not necessarily have the same number of points of interest.
- For each match i of two points of interest, a score $S(k, i)$, for $i = 1, ..., n(k)$ is computed. This score represents the distance between the local descriptors.
- We choose a number n of points of interest between I and every frame in the sequence (this number is of course smaller than $n(k)$ for every k). The goal here is to make homogeneous comparisons with each image of the sequence, in order to choose the best one. One possible choice can be $n = min_k n(k)$, but in general we preferred to choose a smaller number because the other matches in the list are typically not very reliable (i.e. they have a small score). In the following examples, we fixed $n = 30$. Anyway, we noticed that the results are typically not very sensitive to this choice.
- For each image k we define a distance $d(k) = \sum_{i=1}^{n} S(k, i)$, which measures the degree of global similarity between k and I. If the images do not have common points of interest we set $d(k) = \infty$.
- We sort the $d(k)$ for $k \in \{1, ..., N\}$, and we choose the C first ones in the list.

The proposed method has proved to be very efficient in finding the right candidate images (the result of this method has been plotted for "Titanic" in Figure 4, where the query image is in the fifth position). In both cases ("Children of men" and "Titanic") candidate images are actually part of the same scene of the film as the query image and are, visually, very similar to it (see Figures 2, 3 and 4). However, as it can be seen from the examples, the correct frame does not always have the best score among the candidates. Thus, we need to further refine the comparison.

The main disadvantage of the method just described is its computation time. To select the 10 candidates from a sequence of 2000 images, the algorithm requires at least 30 minutes of calculation, and this may become infeasible for larger data sets. Remark, however, that the procedure is very well fit to parallel computation, with a great margin for time efficiency.

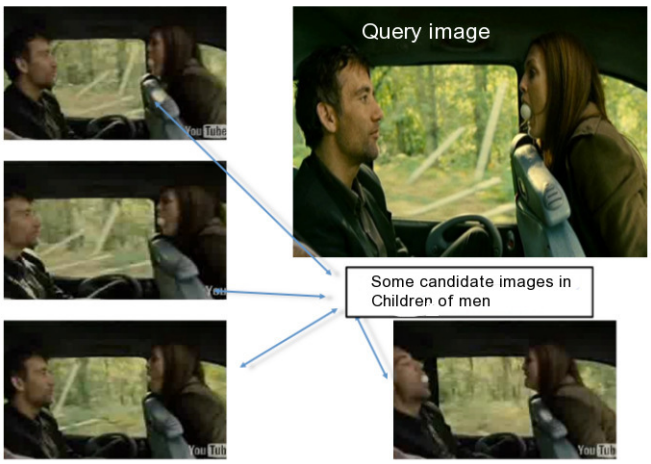

FIGURE 2. Query image of Children of men with some candidate images.

FIGURE 3. Query image of Titanic.

FIGURE 4. 10 Candidate Images in Titanic

3. MOTION-BASED REFINEMENT

In this section (and in what follows) we only deal with grayscale images. However, the procedure can be easily extended by suitably applying it sequentially to every color channel of the image.

In order to compare two images, one of the first ideas that usually comes to mind is to perform a pixel-by-pixel comparison and get the areas where the two pictures are different. Unfortunately this simple procedure is weak in our case as it is not invariant under image transformations. Indeed, we cannot compare the query image with the video sequence because of rescaling, compressing, changing of contrast, brightness, etc. However, comparing frames from the video sequences among themselves seems to be meaningful: those images come from the same sequence and have the same size and quality.

The idea is therefore to compare the candidate images in order to capture an "interest area" in which the images are different (this corresponds to areas in the movie scene frames where some movement takes places).

FIGURE 5. Regions of Interest in Children of Men and Titanic

More precisely, we will say that a point is in a *moving area* if there are two images where this point has different grey scale values (in fact, where the grey levels are above a given threshold). Examples of moving areas detected in this way are shown in Figure 5 for our two test videos. One can immediately see that the frames from *Titanic* are much more similar than the ones from *Children of Men* (the moving area is much smaller than in the latter).

Assuming that the preliminary SIFT procedure has already chosen ten relevant candidate images (as explained in the section above), we are able to compute the moving area and then delete all non-moving-area points from the candidate images. Then, we perform another SIFT comparison between the query image and the modified candidate frames. The detected points are now constrained to be in the regions where the candidate images differ (as we threw away the others), therefore they are more relevant.

In the movie *Children of Men*, this sequential procedure enables us to get the right frame in the sequence. To confirm this efficiency, we tested it on the more difficult Titanic case. The results are, however, less satisfactory in this case, which has led us to look for another method.

4. A GEOMETRIC APPROACH

We now want to tackle the problem of invariance to various geometric transforms. It seems reasonable, in our setting to deal only with the following transformations: scaling, crop and eventually some slight change of aspect ratio. We want to exploit more specifically this geometric rigidity to choose the right image among the ten candidates.

Mathematically, we consider that the query image I is equal the target $\tilde{I}$ up to an affine transformation F which consists in a translation and scaling with respect to the horizontal or vertical axes (see Fig. 6) :

$$\tilde{I} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} I.$$

Since a pair of triangles is enough to uniquely get the coefficients λ_1, λ_2, a_1 and a_2, it can be natural to consider every pair of triangles associated to three corresponding interest points and to compute the coefficients on it.

To deal with this collection of coefficients, our first idea is to proceed by a simple statistical analysis: we choose the image which minimizes the standard deviation for the whole collection of λ_1 and λ_2 corresponding to each pair of triangles of the image. To avoid some meaningless results, we decided to delete the highest and the lowest quarters of results.

For the film *Children of Men*, the results are satisfying, and the right frame is detected by this method (Fig. 7).

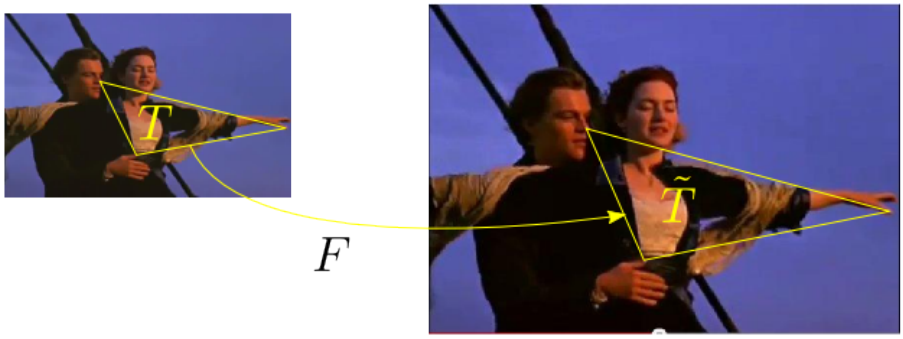

FIGURE 6. Geometrical Transformation on the query Image

Candidate Image	1	2	3	4	**5**	6	7	8	9	10
Standard Deviation λ_1	0.05	0.1	0.05	0.08	**0.01**	0.05	0.19	0.11	0.13	0.13
Candidate Image	1	2	**3**	4	**5**	6	7	8	**9**	10
Standard Deviation λ_2	0.17	0.4	**0.06**	0.25	**0.06**	0.19	1.81	0.09	**0.06**	0.16

FIGURE 7. Statistical analysis of λ_1 and λ_2 in *Children of Men*.

Candidate	1	2	3	4	5	**6**	7	8	9	10
Standard Deviation λ_1	0.44	1.35	0.46	0.67	0.45	**0.16**	0.85	0.58	0.35	0.31
Candidate	1	2	3	4	5	6	7	**8**	9	10
Standard Deviation λ_2	0.41	0.99	0.42	0.21	0.46	0.22	0.44	**0.16**	0.3	1.15

FIGURE 8. Statistical analysis of λ_1 and λ_2 in *Titanic*.

Concerning *Titanic*, the results are not so good (Fig 8), for a quite simple reason: most of interesting points are detected in areas where all the ten candidate images are similar.

In order to get a better grasp of the efficacy of this method, we also tried to find, in another video sequence, a frame which was extracted directly from this sequence. In this case, the standard deviation for the corresponding image is zero, as expected.

As we can see, this method is not perfect, and it is quite easy to imagine some possible improvements. For example:

- We could take into account only the "well sorted" triplets of points (we force the vertical or horizontal order to be kept under the transformation)
- To avoid detections on the non-moving area, we can use the technique introduced Section 3 to apply the latter method on the moving regions only.

We have presented here all the most relevant results that we obtained during our week of work. One can think of different things that could be done in order to improve our findings, and to reach the expected goal for every pair (query image, video sequence).

5. PROSPECTS

All the results above show that the sequence of "Titanic" remains difficult to treat, because of the very small differences between two consecutive images. Here are some unexploited ideas we could develop:

- There are more efficient tools (especially faster) than SIFT to detect similar regions between two images.

- The method based on pixel by pixel differences adds artificial edges which, unfortunately, may correspond to a region of interest of the query image and consequently distort the correspondence. It would be more relevant to modify SIFT to be able to seek the points of interest in the moving regions without changing the candidate images.
- It may be advantageous to use the geometric method to collect the values of λ_1, λ_2, a_1 and a_2 in order to fit the query image with candidate images. Thus, it would become possible to compare them to each other pixel by pixel.

References

[1] Lowe, David G. (1999). *Object recognition from local-scale invariant features*. Proceeding of the International Conference on Computer Vision. **2**. pp. 1150-1157.

[2] Lowe, David G. (2004). *Distinctive image features from scale-invariant keypoints*. International journal of computer vision. **60**. pp. 91-110.

[3] D. Marr and E. Hildreth (1980). *Theory of Edge Detection*. Proceedings of the Royal Society of London. Series B, Biological Sciences (The Royal Society) 207 (1167): pp. 215-217.

[4] Mikolajczyk, Krystian et Schmid, Cordelia (2002). *An affine invariant interest point detector*. In : Computer Vision - ECCV 2002. Springer Berlin Heidelberg. pp. 128-142.

[5] A. Vedaldi and B. Fulkerson (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/