

Qualitative Analysis of Spatio-Temporal Event Detectors

Benedikt Kaiser

*University of Karlsruhe, Institute for Process Control and Robotics
Building 40.28, Kaiserstr. 12, D-76128 Karlsruhe, Germany
Email: kaiser@ira.uka.de*

Gunther Heidemann

*Stuttgart University, Intelligent Systems Group
Universitätsstr. 38, D-70569 Stuttgart, Germany
Email: ais@vis.uni-stuttgart.de*

Abstract

Interest point detection is an established method to select relevant image regions. Such techniques use features like corners or edges, which are known to indicate regions likely to hold patterns of interest. Selection of such regions increases processing efficiency. For the recognition of motion, however, such context-free methods are still very rare. Though there are numerous methods to find space-time volumes of motion in image sequences, most aim at finding just motion as a such, not volumes which are more promising for analysis than others. Therefore Laptev and Lindeberg (2005) generalized the Harris detector to the spatio-temporal domain. But the problem remains to evaluate what kind of motion is captured by a detector. For example, the detector of Laptev and Lindeberg should capture “corners” — like the original 2D-version of Harris and Stephens (1988) — but what does that mean for motion? Therefore we present an approach to visualize events which were selected by a spatio-temporal interest point detector. Since the analysis of single examples is not fruitful, we use clustering to analyze large quantities of space-time volumes selected by a detector. The resulting cluster centers are prototypical events, representing the types of events the detector responds to. Thus a qualitative yet statistically exhaustive analysis of detector properties is possible.

1 Introduction

One of the main problems in Computer Vision is still computational efficiency, despite all developments in increasing computer power. Therefore, practically

all approaches in image analysis rely on some selection of relevant image regions before computationally costly analysis sets in. For single frame analysis, lots of methods exist, most of which belong to one of the following two categories: Region segmentation or interest point detection. Segmentation usually searches for regions which are homogeneous with respect to some feature such as color or texture (e.g. [8]), or their boundaries (e.g. [9]). By contrast, interest point detectors search for features like symmetry [11] or corners [2], and represent such regions by a single point. So both segmentation and interest point detection single out regions that exhibit features which are *object-related yet context-free*. This concept is highly efficient, since there is no need for domain adaptation, no costly computation or implementation, and yet a high selectivity, i.e., increase in the likelihood to filter out candidates relevant for further processing. For interest point detection, there is also significant evidence that the human visual system works in the same way and responds to similar features, e.g., symmetry [7, 10]. Applications range from image retrieval [13] over active vision [1] to object recognition [3]

For single frames, the evaluation of segmentation and interest points is straight forward (though requiring some effort): We can simply look whether or not a detector filters out the objects we are after. But for image sequence analysis, the situation is more difficult. It is usually not an object we want to filter out — this could be done by repeated single frame analysis and tracking techniques — but motion. Filtering out motion raises the question what “type” of motion is relevant. This question should be answered from a context-free point of view, not for a particular task.

We think that a major obstacle in developing context-

free event detectors is that one can not tell what they really detect. While 2D interest points can be easily related to the underlying patterns, 2D+1D (i.e. space plus time) interest points denote a mixture of a spatial with a temporal pattern. To make the results of a spatio-temporal detector interpretable, we suggest to cluster the detected space-time volumes and visualize the obtained reference vectors (cluster centers) as prototypic events. In the following section 2 we will briefly review an algorithm for 2D+1D corner detection. In section 3, we will evaluate the results of the detector by clustering and visualizing its selected space-time volumes. The results will be discussed in section 4.

2 An algorithm for spatio-temporal interest point detection

Since 2D-corners are salient points for single frames, carrying even more information than region boundaries, it is promising to search for 2D+1D corners in the spatio-temporal domain. Laptev and Lindeberg [6] have therefore proposed a temporal extension of the Harris detector [2]. But since the Harris detector requires on the often problematic computation of derivatives, we have extended the SUSAN detector proposed by Smith and Brady [12] to the spatio-temporal domain instead [5]. The advantage of the SUSAN detector is its robustness, since it does not depend on derivatives. We will briefly report on our approach, for details see [5]. We have to provide that the reader is familiar with the SUSAN detector [12].

Algorithm 1 SUSANinTime(x, y, t)

```

nucleus  $\leftarrow$  getpixel( $x, y, t$ )
for  $t' = -R$  to  $R$  do
  areas[ $t'$ ]  $\leftarrow$  0
  for  $y' = -r$  to  $r$  do
    for  $x' = -r$  to  $r$  do
      if mask[ $x$ ][ $y$ ][ $t$ ] = 1 then
        pixel  $\leftarrow$  getpixel( $x + x', y + y', t + t'$ )
        areas[ $t'$ ]  $\leftarrow$  areas[ $t'$ ] +  $c_1$ (nucleus, pixel)
      end if
    end for
  end for
end for
nucleus2  $\leftarrow$  areas[0]
value  $\leftarrow$  0
for  $t' = -R$  to  $R$  do
  value  $\leftarrow$  value +  $c_2$ (nucleus2, areas[ $t'$ ])
end for
return value

```

The straight forward extension of SUSAN would be

to generalize its 2D circular mask to a 2D+1D ellipsoid around the time axis. But this “naive” extension of SUSAN fails, since geometrical and dynamical features are coupled implicitly. We have therefore developed the SusanInTime approach, which generalizes SUSAN using a cylindrical space-time volume around the nucleus (Alg. 1). The USAN-area (see [12] for the definition) is computed within this cylinder, the single x-y-slices of the cylindrical mask are evaluated to find the USAN-areas for every frame. These values are saved in a 1D-array (areas []). Then the SUSAN principle is applied to the 1D-array areas [] in the following way: The USAN-area at the current time is considered to be a (second) nucleus value (nucleus2). Note the second nucleus value is an *area*, no gray value. Then the array areas [] is binarised with respect to the nucleus2 value, and the final detector response is the sum of the now binarised array. Interest points are the minima of this response.

3 Event analysis by clustering

In the following, we apply SusanInTime to a video collection. The detector selects salient space-time volumes, which we visualize using a clustering method.

3.1 Image sequence data

The image sequences should represent no particular domain, instead, a wide variety of real world motion performed by many different kinds of objects should be covered. Therefore, we have invested the available computing power in processing as many different sequences as possible, instead of going for long sequences. We used 50 sequences of length 3 to 7 seconds, at a frame rate of 25 frames per second. The sequences are partly from the TV, partly movies, and partly self-made.

3.2 Sequence analysis

In the first step, all sequences were processed using SUSANinTime to obtain spatio-temporal interest points, around which cylindrical space-time volumes were cut out. This set of space-time volumes was then partitioned into clusters using a vector quantization algorithm. Here we used the Activity Equalization algorithm proposed in [4] due to its computational efficiency and easy to adjust parameters, however, other methods for vector quantization might be equally used. The result of the vector quantization is a set of reference vectors, which are visualized in Fig. 1. The example shows

10 reference vectors, time is top to bottom. The reference vectors can be viewed as “prototypical events”. Vectors (b, c, f, g) show a white-black edge moving across the “pinhole”, whereas (e) has captured a moving corner and (d) a more complex event.

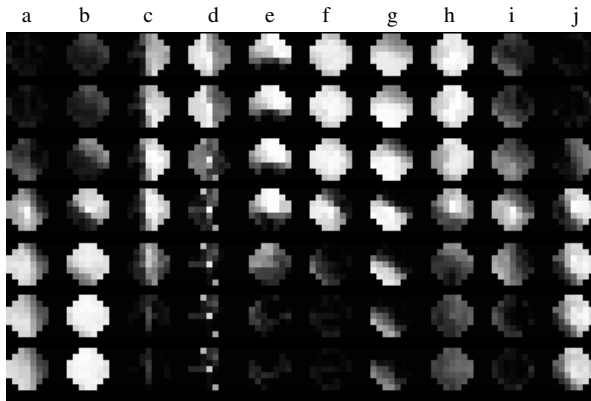


Figure 1. Ten reference vectors obtained from clustering. Time is top to bottom. As there was no rotational alignment of the processed space time volumes, several reference vectors are “wasted” on representation of the same event (moving edge). See text.

A drawback of the above approach is that several reference vectors are wasted to represent the same event: (b, c, f, g) just show the edge under different angles. Consequently, a large number of reference vectors must be used. This is a problem for two reasons: (i) A correspondingly large number of samples is required, since every reference vector should stand for a reasonably sized set of samples. (ii) Events that differ only by rotation must be manually grouped to allow interpretation.

As a solution, we rotate each volume extracted from the sequence in the image plane to achieve rotational alignment. For this we use the vector to the center of mass of the USAN, which points along the x-axis after rotation. Thus, we get rid of one “degree of freedom”, i.e., the rotation around the time axis.

Fig. 2 shows 19 reference vectors, which were obtained after rotation of the space time cylinders. At first glance, it seems that still several reference vectors represent just the same moving edge, now aligned to a vertical direction. But in fact, different events are captured: (a), (d), (k), (m), (n), (p) and (r) show moving edges with constant yet different velocities. Moving bars are represented by (e), (l) and (q). Reference vector (j) can be interpreted as a black object which appears, or a white object which disappears. Reference vector (i) is

a temporal impulse (“lamp switched on and off”). Most interesting are (b) and (s), which represent turn-around movements.

4 Discussion of results

The visualization of the “typical” events to which a detector responds should answer the following questions:

1. Is the sensitivity of the detector dominated by dynamical patterns?
2. What are the semantics of the events?
3. Are the events associated with particular 2D-patterns?

We will discuss these questions (a) for the results obtained with the particular detector used here (SusanInTime) and (b) in general. The first question is easily answered: The SUSANinTime detector selects primarily dynamic patterns (events), since the spatial patterns change over time for all reference vectors. Likewise, it would be easy to find out if a different detector responds to static patterns — some reference vectors would display constant patterns in this case.

The semantics of the events (question 2) is also easy to find out: We find several very basic events, such as moving bars, appearance / disappearance, impulses and turn-around events. However, it is not yet clear whether these events are all the detector responds to, because the evaluated space time volume is small. An event such as a double turn would probably be invisible because of both the spatial and temporal borders. But this problem can be mended with a larger space-time cylinder.

Finally, do the events “mix” with 2D object structure (question 3)? Naturally each event is based on the motion of stationary patterns, but, as far as visible, SusanInTime does not prefer objects of a particular structure. But again this may be due to the limited volume under consideration. Another problem is that the prototypes represent the average of each cluster, so stationary patterns might just be averaged out. Therefore, the last question can probably not be answered by clustering, other methods have to be developed.

5 Conclusion

We have presented a method to visualize the events filtered out of videos by an event detector. The visualization clearly shows the types of events captured by the detector, however, the space-time volume considered here is too small to admit more complex events.

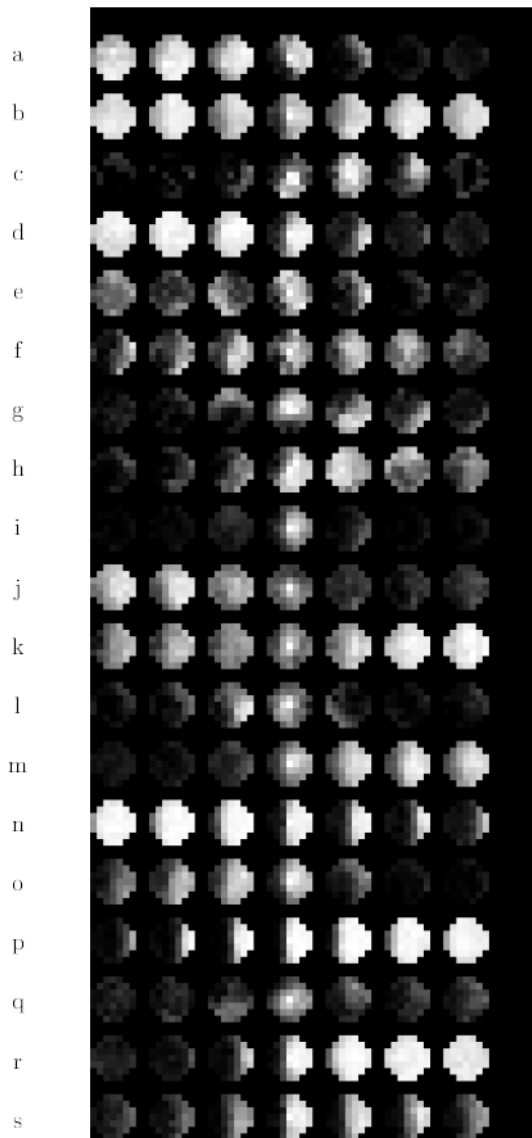


Figure 2. 19 reference vectors obtained from clustering after each space time volume has been rotated (around the time axis) to achieve common alignment such that the center of mass vectors of the USAN are all in the same direction. This results in more complex patterns than in Fig. 1.

This is no problem in principle and will be solved with increased volumes in future work. Also, the ratio of stationary vs. dynamical patterns is easy to determine. A more severe problem is that clustering can not decide how strong geometrical and dynamical features are coupled. Here we will try to fuse our approach with spatio-temporal principal component analysis.

Evidently, there are other desirable properties an event detector should exhibit, such as repeatability in the presence of distortions of the object or the trajectory, or varying illumination. But checking *what* a detector captures is naturally the first of all questions. As it is now possible to inspect the performance of event detectors in an intuitive way, we think that this simple method will facilitate future development.

References

- [1] G. Backer, B. Mertsching, and M. Bollmann. Data- and Model-Driven Gaze Control for an Active-Vision System. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1415–1429, 2001.
- [2] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conf.*, pages 147–151, 1988.
- [3] G. Heidemann. Focus-of-Attention from Local Color Symmetries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(7):817–830, 2004.
- [4] G. Heidemann and H. Ritter. Efficient Vector Quantization Using the WTA-rule with Activity Equalization. *Neural Processing Letters*, 13(1):17–30, 2001.
- [5] B. Kaiser and G. Heidemann. Context-Free Detection of Events. In *Proc. 15th Scandinavian Conference on Image Analysis SCIA 07*, pages 223–232, Aalborg, Denmark, 2007.
- [6] I. Laptev. On Space-Time Interest Points. *Int'l J. of Computer Vision*, 64(2-3):107–123, 2005.
- [7] P. J. Locher and C. F. Nodine. Symmetry Catches the Eye. In A. Levy-Schoen and J. K. O'Reagan, editors, *Eye Movements: From Physiology to Cognition*, pages 353–361. Elsevier Science Publishers B. V. (North Holland), 1987.
- [8] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and Texture Analysis for Image Segmentation. *Int'l J. of Computer Vision*, 43(1):7–27, 2001.
- [9] D. R. Martin, C. C. Fowlkes, and J. Makik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1), 2004.
- [10] C. M. Privitera and L. W. Stark. Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):970–982, 2000.
- [11] D. Reissfeld, H. Wolfson, and Y. Yeshurun. Context-Free Attentional Operators: The Generalized Symmetry Transform. *Int'l J. of Computer Vision*, 14:119–130, 1995.
- [12] S. Smith and J. Brady. SUSAN – A New Approach to Low Level Image Processing. *Int'l J. of Computer Vision*, 23(1):45–78, 1997.
- [13] Q. Tian, N. Sebe, M. S. Lew, E. Loupias, and T. S. Huang. Image Retrieval Using Wavelet-Based Salient Points. *J. of Electronic Imaging*, 10(4):835–849, 2001.