Semi-automatic ground truth annotation in videos

An interactive tool for polygon-based object annotation and segmentation

Julius Schöning
Institute of Cognitive Science
University of Osnabrück
Osnabrück, Germany
juschoening@uos.de

Patrick Faion
Institute of Cognitive Science
University of Osnabrück
Osnabrück, Germany
pfaion@uos.de

Gunther Heidemann
Institute of Cognitive Science
University of Osnabrück
Osnabrück, Germany
gheidema@uos.de

ABSTRACT

Knowledge extraction from video data is challenging due to its high complexity in both the spatial and temporal domain. Ground truth is crucial for the evaluation and the adaptation of algorithms to new domains. Unfortunately, ground truth annotation is inconvenient and time consuming. Common annotation tools mostly rely on simple geometric primitives such as rectangles or ellipses. Here we propose a novel, interactive and semi-automatic process, which actively asks for user input if the result of the automatic annotation appears to be incorrect. After a brief review of related tools for video annotation, we explain our proposed semi-automatic method *iSeg* using a prototype implementation. *iSeg* has been tested on two visual stimulus datasets for eye tracking experiments and on two surveillance datasets. The experimental results and the usability are compared to existing annotation tools. Finally, we discuss the properties and opportunities of polygon-based video annotation.

Keywords

ground truth, annotation, polygon based, semi-automatic, user in the loop

1. INTRODUCTION

According to the official press statistics of *Youtube*, about 300 hours of video are uploaded every minute [14] just to this platform. In case all uploaded videos would be watched and annotated manually in real time, 18,000 operators would be necessary. This illustrates the importance of automatic knowledge extraction and acquisition from video data. In some cases, the successful detection of semantic concepts such as "persons", "buildings" or "cars" is already possible with current automatic content analysis and understanding tools [3]. As detection methods are rapidly developing, manual ground truth annotation becomes even more important for the development of algorithms — both for training and evaluation.

In the architecture of video visual analytics [12] the computer assists the user on the two lowest levels of the reasoning process: the *extraction of meaningful artifacts* and the *assessment of situations*. In common annotation methods for video and image data [3], no such techniques are used. With *iSeg* we try to transfer this architecture to combine the computational power of a computer with the high level abilities of the human user. By this cooperation between human and computer the quality of the results improves significantly and annotation speed increases slightly.

While current freely available video annotation tools [4, 13, 10, 11] usually provide only simple geometric primitives like rectangles or ellipses, a significant improvement in video annotation are polygon-shaped areas. Another problem of current tools is that they provide no or little support for an easy concurrent annotation of several frames. To overcome these drawbacks, we propose our novel interactive, polygon-based and semi-automatic method *iSeg*. *iSeg* actively asks for user interaction if the result of the automatic annotations seems to be incorrect. *iSeg* creates annotations in every polygon based shape with more than three edges. For better usability, it provides intuitive and easy to use interaction metaphors.

2. STATE OF THE ART

Dasiopoulou et al. [3] reviewed and compared several image and video annotation tools, using several criteria like in- and output formats, metadata types, granularity, localization, and expressivity of the annotations. According to their review, four of seven image annotation tools provide polygon-based annotations, but in contrast only one of seven video annotation tools provides polygon-based annotations, namely the Video Image Annotation (*VIA*) tool [10]. However, in our own test of *VIA* it turned out that we were not able to activate polygon-based annotation. Therefor only rectangular annotation markers were available in the user interface. In addition, we noticed *VIA* only displays a clipped region of the input video if the resolution is FullHD.

One quite old but still in use tool for ground truth generation is the Video Performance Evaluation Resource (*ViPER*) [4]. An automatic 2D propagation of the annotated object can be used to speed up the annotation process. Due to its properly defined and specified XML output format with XSD schema, the usage of annotations done with *ViPER* is easy.

The Semi-Automatic Ground Truth Annotation tool (*SAGTA*) [13] is designed for the rectangular annotation of pedestrians. The semi-automatic process relies on the as-

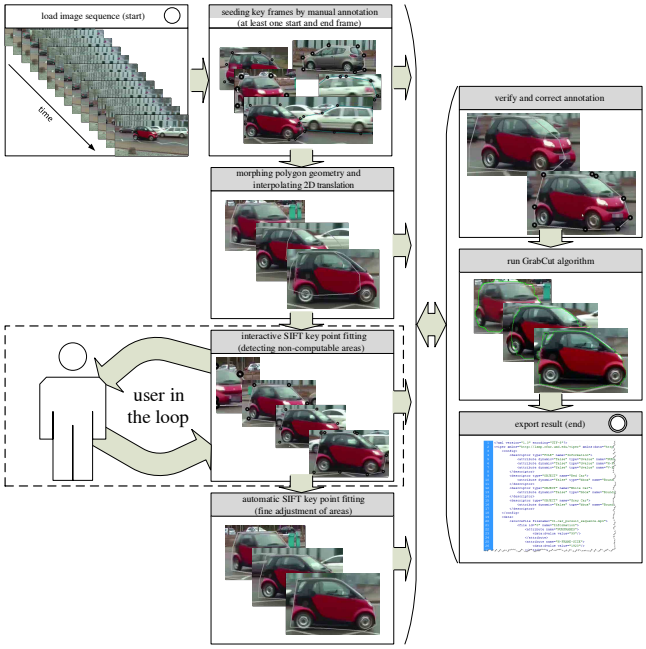

Figure 1: Overview of the interactive semi-automatic annotation and segmentation process. Process blocks with white headlines are obligatory to run *iSeg*. Blocks with gray headlines are optional and can be used in any order at any time. The block *interactive SIFT key point fitting* actively asks for user interaction in case the automatically generated annotations appear to be incorrect. Thus, the computer remains the "work horse" of the process, while the close cooperation with the user allows *iSeg* to achieve sophisticated results.

sumption of 3D linear motion and is technically based on ORB feature matching. Thus, it reduces the number of manually annotated frames. Due to the 3D linear motion assumption of *SAGTA* the camera needs to be stationary.

For performing annotations in real-time, collaborative approaches can pool the resources of several users. Vannotea [11] implemented this still exotic approach and thus multiple users can index, browse, annotate and discuss the same video sequences simultaneously.

3. iSeg

Reflecting the proposed architecture of video visual analytics [12], our *i*nteractive annotation and *Seg*mentation tool (*iSeg*) focuses on a semi-automatic architecture putting the user in the loop. As shown in Figure 1, *iSeg* consists of eight main process blocks. Two blocks of these eight are obligatory and must be processed in a specific order, marked with a white headline in Figure 1. The remaining process blocks, marked with a gray headline, can be executed by the user in any sequence and repeated as often as necessary until the intended annotation is achieved.[1] In the following, selected process blocks are described in detail.

3.1 Polygon morphing

[1]Demonstration video of the annotation process with *iSeg* https://ikw.uos.de/~cv/publications/k-cap15

The user needs to identify the areas of interest (AOI) only on a few frames. Therefore, the algorithm must estimate the positions and the contours of the AOI on the intermediate frames. To keep things as convenient as possible, the user can use polygons with varying numbers of vertices on different frames. Contours of AOI can be either convex or concave, are intersection free, and for simplicity holes in the AOI are omitted. Thus, the task is to morph two non self-intersecting polygons with different numbers of vertices. In addition, all intermediate polygons also must not be self-intersecting, since they represent contours as well.

There are several existing algorithms in the literature concerning polygon interpolation [2, 1, 6, 8]. Since none of the existing methods seemed to fit our needs, we implemented a new, very basic form of polygon matching to test whether the semi-automatic approach with the aid of computer vision is viable.

Given two polygons $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ and $\mathcal{B} = \{b_1, b_2, ..., b_m\}$ we start by separating out the translation component by centering the polygons on their center of gravity. In the future we would also like to extract a rotation component beforehand by means of computer vision. To cope with the problem of different vertex numbers, we introduce additional points in the polygons. For every point in $\mathcal{A}$ there will be an additional point on the contour of $\mathcal{B}$ and vice versa, so all intermediate polygons will have $n + m$ points. For every point $a_i \in \mathcal{A}$ the position of the matching additional point on the contour of $\mathcal{B}$ is the point on the contour with the smallest distance to $\mathcal{A}$: $match(a_i) = \arg\min_{c_i \in \pi(a_i)}(||a_i - c_i||)$ where $\pi(a_i)$ is the set of closest points to a_i on every line segment in $\mathcal{B}$. The points $\{a_i\}$ and $\{match(b_j)\}$ (as well as $\{b_j\}$ and $\{match(a_i)\}$) will then be collected into two new polygons called $\mathcal{A}'$ and $\mathcal{B}'$, representing the same shapes as $\mathcal{A}$ and $\mathcal{B}$, but now with $n + m$ points each.

Each point from the contour of $\mathcal{A}'$ matches to a corresponding point on the contour of $\mathcal{B}'$ and vice versa. Unfortunately, there can be cases where predecessor-successor relations are violated. Checking the order of vertices in A and B and exchanging conflicting vertices can correct some of these violations. To obtain the intermediate polygons, we interpolate linearly between matched points as well as along the translation vector (cf. video[1] 01 : 20).

The algorithm is very simple and straightforward and already works in many cases, especially when the two polygons are not completely different in their shape, which they rarely will be in practice. Still, it is only a heuristic approach and there are problems with some polygons where the point matching is not possible such that predecessor-successor relations are preserved. In these cases, self-intersecting polygons may appear on intermediate frames. With time complexity $\mathcal{O}(nm)$ the algorithm is rather fast, which makes it useful for testing purposes. We are aware our approach is still at an early stage and might not be adjustable to work without errors in all situations.

3.2 Interactive semi-automatic AOI fitting

Within this process block the 2D linearly interpolated and morphed polygon-shaped AOI will be adjusted to fit the real object on each frame. Thus the AOI follows the non-linear movements of the real object. We use the scale-invariant feature transform (SIFT) algorithm [8] to extract z key points

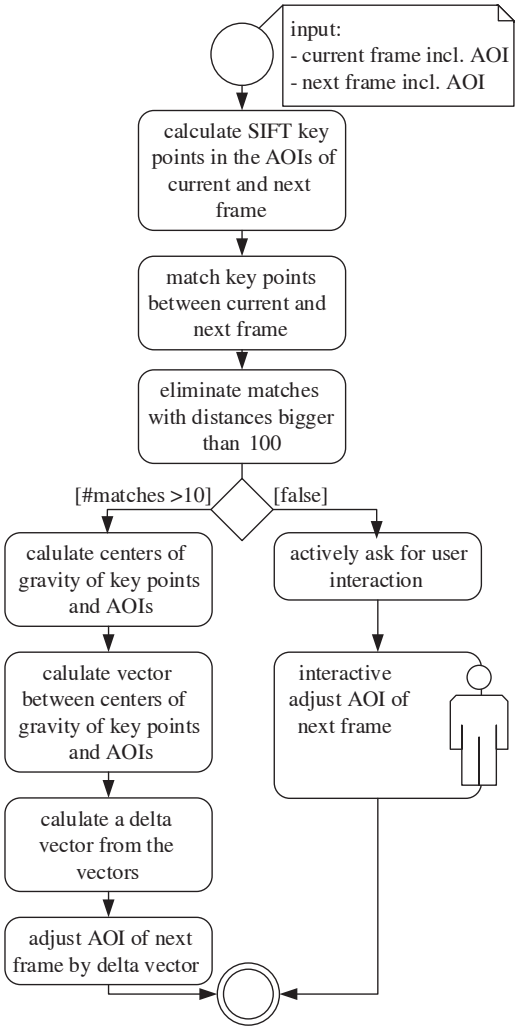

Figure 2: Activity diagram of the interactive semi-automatic AOI fitting. It has to be repeated until all AOI are adjusted.

$\mathcal{F}_1 = \{f_1, f_2, ..., f_z\}$ within the AOI $A_1 = \{a_1, a_2, ..., a_n\}$. Note that computing SIFT key points on the whole image would increase the processing time such that human computer cooperation becomes infeasible, since the "waiting time" for the user would raise to an unacceptable level.

The key points $\mathcal{F}_i$ are calculated for the current $\mathcal{F}_1$ and for the next frame $\mathcal{F}_2$, highlighted with white circles in Figure 3. Under the assumption that rotation and scaling of the object within the AOI is small between successive frames, the key points from the current frame $\mathcal{F}_1$ are matched with the next frame $\mathcal{F}_2$ using the FLANN algorithm [9]. Based on the matching result $\mathcal{M}_{1-2} = \{m_1, m_2, ..., m_z\}$, all key points f_i with FLANN distances bigger than 100 are eliminated. In case more than 10 key points of $\mathcal{F}_1$ and $\mathcal{F}_2$ are left after key point elimination, the centers of gravity of the remaining key points $\mathcal{C}_\mathcal{F} = \sum_{g=1}^{i} f_g$ for $\mathcal{F}_1$ and $\mathcal{F}_2$ are calculated, shown as blue points in Figure 3(a). Then the centers of gravity $\mathcal{C}_A = \sum_{g=1}^{n} a_g$ of the AOIs in the current and next frame are computed, in Figure 3(a) marked by a gray point. Next the vector $\vec{v} = (\mathcal{C}_\mathcal{F} - \mathcal{C}_A)$ between the center of gravity of the key points and the center of gravity of the AOI is determined. If rotation and scaling are small, the vector $\vec{v}$ of the current and of the next frame are approximately the same. Under the assumption that the center of gravity of the key points $\mathcal{C}_\mathcal{F}$ matches the non-linear movement of the object, it is used to adjust the AOI. Therefore, the new center of gravity of the AOI $\mathcal{C}'_A$ in the next frame is calculated with reference to vector $\vec{v}$ of the current frame. The new center $\mathcal{C}'_A$ is marked as a red point in Figure 3(a). With the difference $\Delta = \mathcal{C}_A - \mathcal{C}'_A$, an affine transformation of the AOI using homogeneous coordinates is performed. Thus the AOI is transformed and the non-linear motion of the AOI is taken into account.

As shown in Figure 3(b), in case less than 10 key points of $\mathcal{F}_1$ and $\mathcal{F}_2$ are left after the elimination, the algorithm actively asks the user for interaction. The user can now adjust the polygon AOI with three intuitive metaphors described in Section 3.4 and continue the interactive semi-automatic AOI fitting process. Possible reasons for having less than 10 key points left are that the size of the AOI is too small, the AOI is occluded, the AOI mainly contains textureless areas, or the object in the AOI changes between frames. The AOI fitting is continued in close cooperation with the user until all AOI are detected that cannot be computed automatically (cf. video[1] 01 : 34).

3.3 Automatic AOI fitting

The automatic AOI fitting employs almost the same algorithm as the semi-automatic AOI fitting, as illustrated in the activity diagram in Figure 2. The difference is that in case less than 10 key points are left after the elimination, the next frame remains unchanged and the process continues. The idea of this process block is that it is performed after the *interactive semi-automatic AOI fitting* has been performed once. Thus all non-automatically computable AOI have been detected. On that basis the automatic AOI fitting increases the accuracy of the result iteration by iteration (cf. video[1] 02 : 45), because all difficult cases are resolved in cooperation with the user.

3.4 AOI verification and correction

The intuitive user interface enables the user to verify and correct the result in every stage of the process. Currently, three metaphors are available: i) the relocation of the whole AOI by clicking inside the AOI and dragging the AOI to the designated area (cf. video[1] 02 : 18), ii) the adjustment of vertices (single click, then drag) and adding vertices (double click between two existing vertices; cf. video[1] 02 : 05), and iii) the complete re-creation of the AOI by deleting the vertices and creating new vertices by double-clicking (cf. video[1] 02 : 10).

3.5 Data export

At any time the user can trigger data export (cf. video[1] 04 : 26). In the current prototype we only export the annotated AOI in an XML format valid to the XSD schema of the *ViPER* tool [4]. The exported XML includes the polygon annotations of all AOI for all frames.

4. EXPERIMENTAL TESTS

The performance of *iSeg* was evaluated on the car dataset

(a) ≥ 10 key points – automatic adjustment of the AOI

(b) < 10 key points – actively request for interaction

Figure 3: Example of used SIFT key points for the AOI fitting. On the left: AOI of current frame; On the right: AOI of next frame; White circles: SIFT key points $\mathcal{F}$; White lines: FLANN matches of key points; Blue points: Center of gravity of key points $\mathcal{C}_{\mathcal{F}}$; Gray points: Center of gravity of the AOI $\mathcal{C}_A$; Red point: New center of gravity of the AOI $\mathcal{C}'_A$

01-car pursuit [7] and three more datasets for which annotations already exist. We tested the reliability of the process blocks illustrated in Figure 1 including the semi-automatic AOI fitting based on unknown datasets.

To test highly dynamic boundaries, we chose the *03-dialog* [7], because face boundaries exhibit the desired rapid change. Further, we used frames $0 - 60$ of the *S1 L1 PET2009* [5] video and the *road* [13] dataset. We chose the last two videos to be able to compare our *iSeg* with the *SAGTA* [13] tool. As an example, Figure 3 shows one frame of every dataset.

In the test series, the same objects were annotated with the use of *iSeg* as the existing annotation. To give an idea how long the annotation process using *iSeg* needs, the processing time is determined (cf. video[1] $05:07$).

5. RESULTS AND DISCUSSION

Our interactive semi-automatic tool *iSeg* for polygon-based object annotation and segmentation works well on different annotation scenarios. We compared the annotation time of *iSeg* with the annotation time of *SAGTA* [13]. In total, the annotation using *iSeg* took $83min$, $33min$ longer than using *SAGTA*, but this is unsurprising since polygon-shaped AOIs (*iSeg*) contain much more information and are more accurate than rectangular AOI. Correspondingly, adjusting polygon-shaped AOI takes more time.

With the interactive semi-automatic AOI fitting, non-linear AOI tracking is possible under the assumption that the object in the AOI does not rotate or change shape strongly. To overcome these restrictions, we will extend the activity diagram (Figure 2) with features which detect deformation and rotation. Thus the affine transformation of the AOI can be improved substantially. From the usability point of view, a time-line should be implemented for three reasons: firstly to improve the navigation in the video file, secondly to arrange the AOI such that occluded AOI are behind other AOI, and thirdly to describe relations between the AOI.

The question whether polygon annotation is worth the trouble we would answer with a clear yes: A polygon describes the boundaries much more precisely than a rectangle. Should a rectangle be sufficient — mind that rectangular areas are easier to process for subsequent methods — it can be easily derived, e.g., for the analysis of gaze data. The prototype version of *iSeg* can be downloaded[2] for *Ubuntu*, *Mac OS* and *Windows*.

6. ACKNOWLEDGMENTS

This work was funded by the German Research Foundation (DFG) as part of the Scalable Visual Analytics Priority Program (SPP 1335).

7. REFERENCES

[1] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation: A survey. Technical report, Handbook of Computational Geometry, 1996.

[2] F. Cobos and J. Peetre. Interpolation of compact operators: the multidimensional case. *Proceedings of the London Mathematical Society*, 3(2):371–400, 1991.

[3] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. A survey of semantic image and video annotation tools. *Lecture Notes in Computer Science*, pages 196–239, 2011.

[4] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. *International Conference on Pattern Recognition (ICPR)*, 2000.

[5] J. Ferryman and A. Shahrokni. PETs2009: Dataset and challenge. *Performance Evaluation of Tracking and Surveillance*, Dec 2009.

[6] C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. *Computers & Graphics*, 25(1):67–75, Feb 2001.

[7] K. Kurzhals, C. F. Bopp, J. Bässler, F. Ebinger, and D. Weiskopf. Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli. *Workshop on Beyond Time and Errors Novel Evaluation Methods for Visualization (BELIV)*, 2014.

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

[9] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2, 2009.

[10] Multimedia Knowledge and Social Media Analytics Laboratory. Video image annotation tool | multimedia knowledge and social media analytics laboratory `http://mklab.iti.gr/project/via`, May 2015.

[11] R. Schroeter, J. Hunter, and D. Kosovic. Vannotea - a collaborative video indexing, annotation and discussion system for broadband networks. *Workshop on Knowledge Markup and Semantic Annotation (K-CAP)*, pages 1–8, 2003.

[12] P. Tanisaro, J. Schöning, K. Kurzhals, G. Heidemann, and D. Weiskopf. Visual analytics for video applications. *it-Information Technology*, 57:30–36, 2015.

[2] https://ikw.uos.de/~cv/projects/iSeg

[13] S. Wu, S. Zheng, H. Yang, Y. Fan, L. Liang, and H. Su. SAGTA: Semi-automatic ground truth annotation in crowd scenes. *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, Jul 2014.

[14] YouTube. Statistics - youtube `https://www.youtube.com/yt/press/statistics.html`, May 2015.