Demo: Semantic Human Activity Annotation Tool Using Skeletonized Surveillance Videos

Bokyung Lee
KAIST
Daejeon, Republic of Korea
bokyunglee@kaist.ac.kr

Michael Lee
Autodesk Research
Toronto, Ontario, Canada
michael.lee@autodesk.com

Pan Zhang
Autodesk Research
Toronto, Ontario, Canada
pan.zhang@autodesk.com

Alexander Tessier
Autodesk Research
Toronto, Ontario, Canada
alex.tessier@autodesk.com

Azam Khan
Autodesk Research
Toronto, Ontario, Canada
azam.khan@autodesk.com

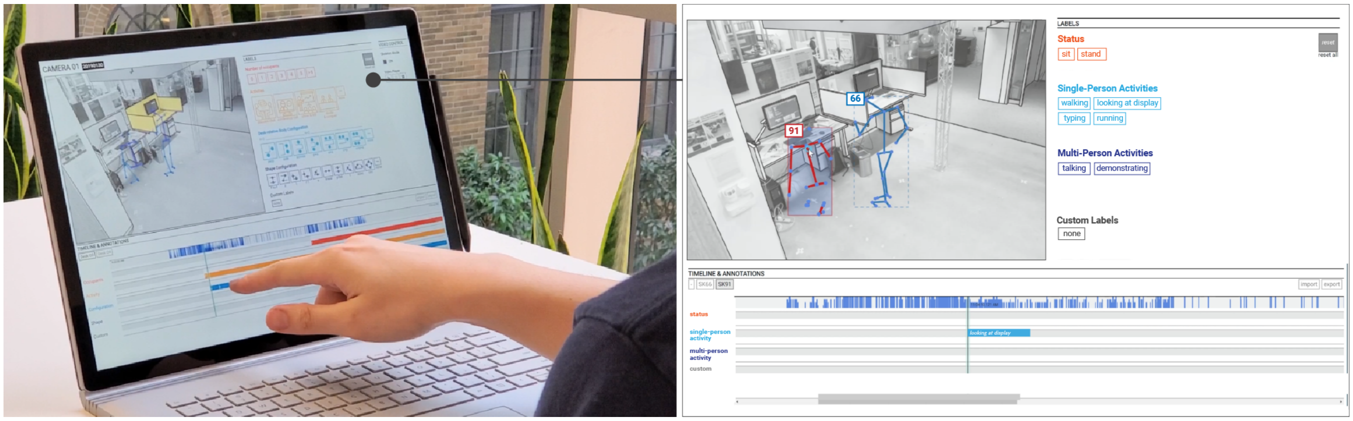


Figure 1: Skeletonotator is a web-based annotation tool that creates human activity data sets using 2D skeletonized poses.

ABSTRACT

Human activity data sets are fundamental for intelligent activity recognition in context-aware computing and intelligent video analysis. Surveillance videos include rich human activity data that are more realistic compared to data collected from a controlled environment. However, there are several challenges in annotating large data sets: 1) inappropriateness for crowd-sourcing because of public privacy, and 2) tediousness to manually select activities of people from busy scenes.

We present Skeletonotator, a web-based annotation tool that creates human activity data sets using anonymous skeletonized poses. The tool generates 2D skeletons from surveillance videos using computer vision techniques, and visualizes and plays back the skeletonized poses. Skeletons are tracked between frames, and a unique id is automatically assigned to each skeleton. For the annotation process, users can add annotations by selecting the target skeleton and applying activity labels to a particular time period, while only watching skeletonized poses. The tool outputs human activity data sets which include the type of activity, relevant skeletons, and timestamps. We plan to open source Skeletonotator together with our data sets for future researchers.

To appear in Ubicomp 2019. Authors pre-print version.

CCS CONCEPTS

• **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*; • **Computing methodologies** → *Activity recognition and understanding*.

KEYWORDS

activity recognition; annotation tool; data set; 2D skeleton

ACM Reference Format:
Bokyung Lee, Michael Lee, Pan Zhang, Alexander Tessier, and Azam Khan. 2019. Demo: Semantic Human Activity Annotation Tool Using Skeletonized Surveillance Videos. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC ’19 Adjunct), September 9–13, 2019, London, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341162.3343807

1 INTRODUCTION

Vision-based human activity recognition is fundamental for context-aware computing and automatic video analysis. It can open up the possibility of natural human-computer interfaces, such as using natural body language as input to manipulate robotic interfaces [7], and can contribute to categorizing large amounts of video [5]. For reliable activity recognition, preparing an extensive annotated data set is an essential process but is often a tedious and time-consuming task.

To facilitate the video annotation process, several web-based tools have been introduced for the crowd-sourcing of annotations [1, 5, 6]. However, these tools still use raw videos as input, which is challenging for annotating large surveillance data sets collected *in-the-wild* in terms of public privacy. Ciliberto et al [3] used anonymized 3D skeletons to maintain occupant’s privacy for activity annotation of a single person. We build upon this work by using 2D skeleton representations of videos and applying it to scenes with many people. Individual skeletons can be tracked over multiple frames eliminating the need for users to manually mark bounding areas when targeting individuals for specific annotations [1].

In this demonstration, we present Skeletonotator, a web-based annotation tool that we developed to annotate a human activity data set collected in our office while maintaining privacy. Our tool generates 2D skeletons from surveillance videos using computer vision techniques, visualizes, and plays back the skeletonized poses. Skeletons are tracked between frames, and a unique id is automatically assigned to each skeleton. For annotation, users can annotate single activities as well as collective activities by selecting corresponding skeletons. The tool outputs human activity data sets, which include the type of activity, relevant skeletons, and timestamps.

2 SKELETONOTATOR

Skeletonotator is a web-based annotation tool that supports creating crowd-sourced human activity data sets using anonymous skeletonized poses generated from surveillance videos (Figure 5). The system generates skeleton data frame-by-frame from video input using computer vision (CV) results from the OpenPose library [2]. The tool enables users to watch and playback skeletonized frames akin to manipulating videos, and supports timeline-based annotation similar to existing video annotation tools [4] (Figure 1).

Interface

The interface is composed of four panels: a) source panel, b) video panel, c) label panel and d) timeline panel, as shown in Figure 3. Users can select a target data set from a drop-down menu in the *source panel*; then the *video panel* displays skeletonized video frames. Users can playback skeletons with custom speed by scrubbing the *control bar* or pressing the hotkeys (<,>,-.+). The *label panel* on the right displays predefined labels as buttons, but users can also create new custom labels at any time. The system automatically generates and tracks unique id for each occupants’ skeletons, and displays different colours for each (Figure 2). For annotation, users can click the relevant skeletons from the *video panel* (singular or multiple), then click corresponding labels. The annotated results are shown in the *timeline panel* to keep track of annotations, and each skeleton owns its timeline panel which can be viewed by clicking the skeleton id on the top of the timeline panel.

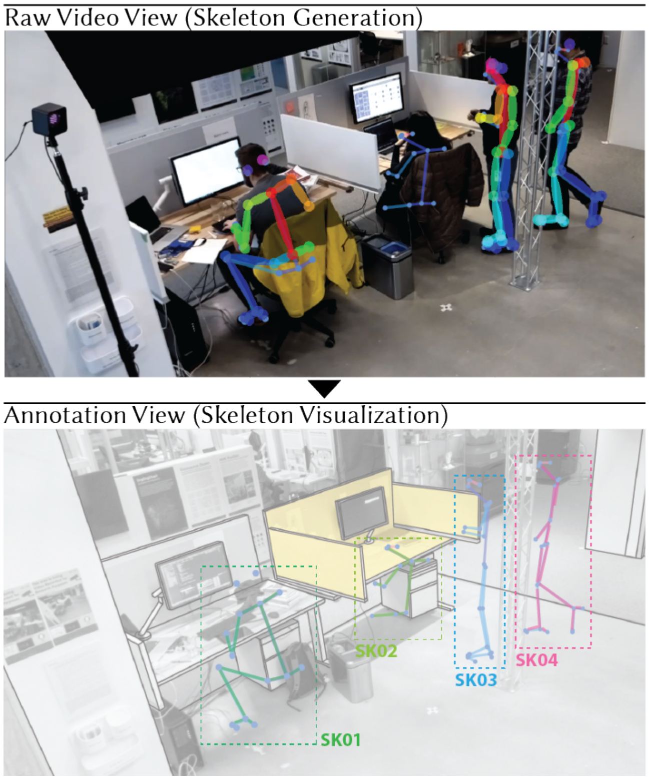

Raw Video View (Skeleton Generation)
Annotation View (Skeleton Visualization)
SK01
SK02
SK03
SK04

Figure 2: Annotation Tool Workflow

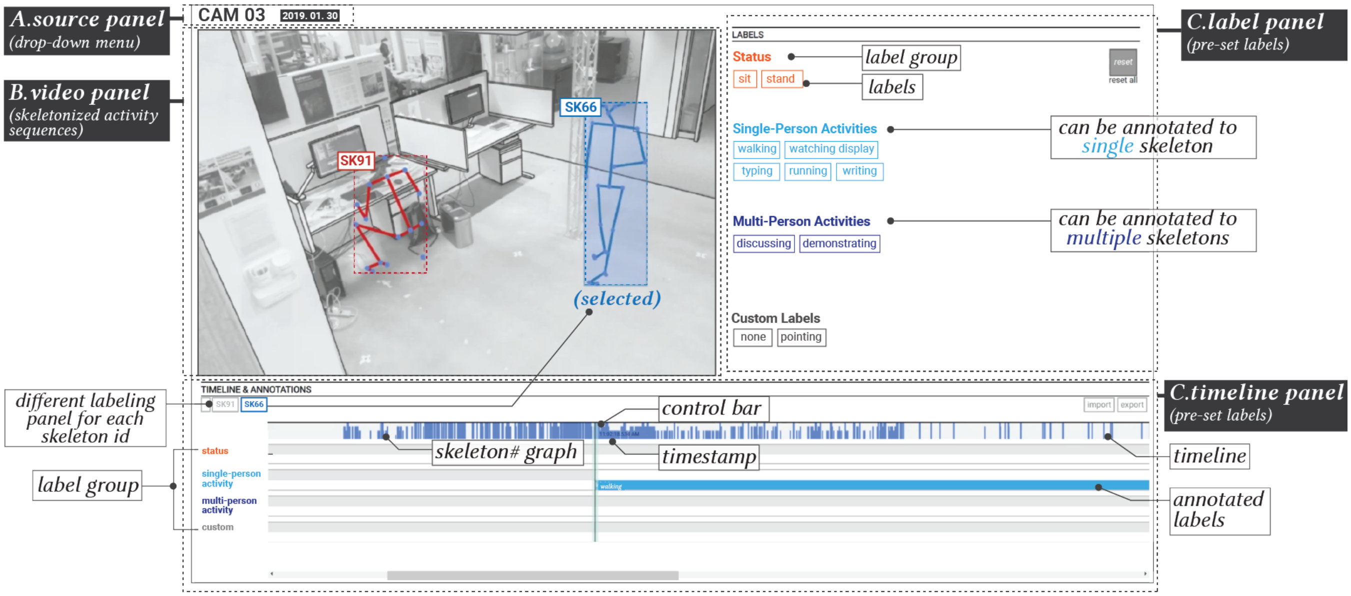
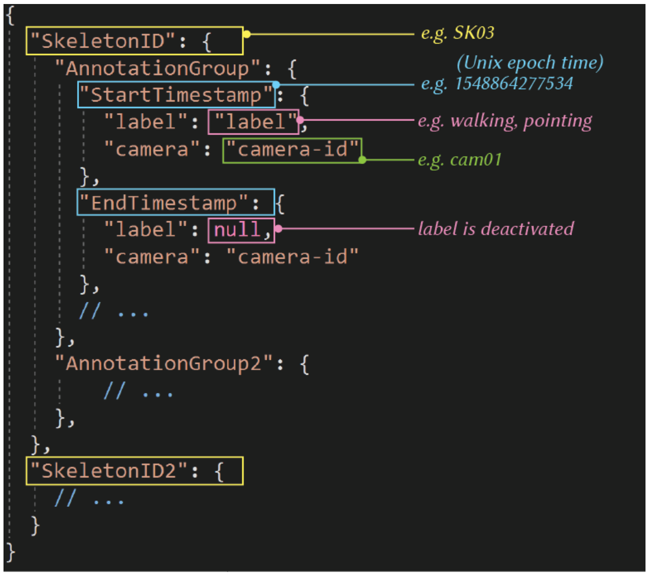


Figure 3: Annotation Tool Workflow

Annotation Output

Once users have finished annotating a data set, they can export the annotations as a JSON file. The structure of the annotation is shown in Figure 4. The annotations are assigned to each skeleton with unique person id. Each type of activities are labelled with corresponding timestamp (starting and ending time).



```
{
  "SkeletonID": {
    "AnnotationGroup": {
      "StartTimestamp": {
        "label": "label",
        "camera": "camera-id"
      },
      "EndTimestamp": {
        "label": null,
        "camera": "camera-id"
      },
      // ...
    },
    "AnnotationGroup2": {
      // ...
    },
  },
  "SkeletonID2": {
    // ...
  }
}
```

Figure 4: Structure of annotated data from Skeletonotator

Implementation

The *OpenPose* library was used to recognize the occupants' embodied poses from the collected videos and generate skeletons based on 25 key points as shown in Figure 2. JSON files corresponding to each frame in the video are generated, containing key points for each of the occupants in the video.

To facilitate annotation, we need a reliable and consistent identifier between frames during sequences. We obtain this identifier by tracking the head of each individual. After finding initial head keypoints in a series of frames, we track the head forward and backward during those sequences. When we momentarily lose detection, we perform matching between sequences by determining if the head in frame n is the same as that in an earlier frame series: at frame $n - i$ (for $i > 1$). We use the heads found at the boundary of sequences and use a distance threshold. Frame u contains the head at position h_u, which is the head we have before losing tracking. Frame v contains the head that appears in a new sequence after the tracking error. The heads match when they are within a heuristically determined threshold T, that depends on both time i and distance h:

$$||h_u^{n-i} - h_v^n|| < T(i, h_u^{n-i})$$

Three local workstations with NVIDIA Quadro P6000 graphics cards were used to process 1,920 hours of video data. We decomposed the processing of files into tasks and developed a job management script in Python to distribute them as jobs. Synchronization was performed using Amazon's Simple Queue Service (SQS). During processing, each

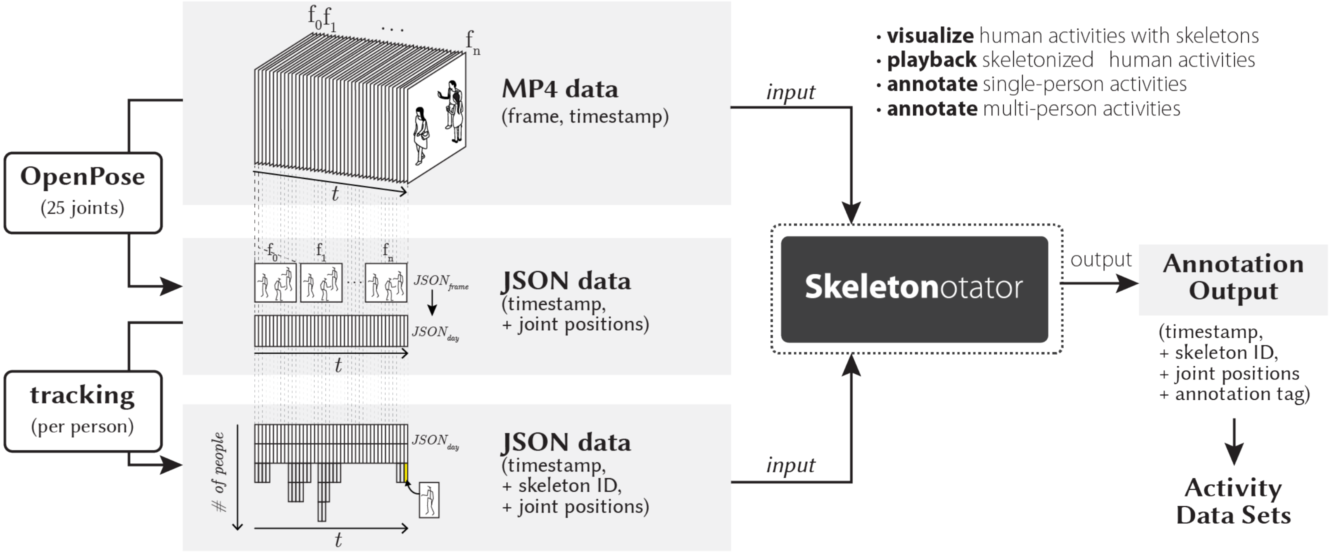


Figure 5: Annotation Tool Workflow

workstation retrieves a video from the file server, performs *OpenPose* [2] processing, then finally stores the results back onto the file server. To efficiently manage a large amount of data, We used a custom *Node.js* server to serve the web-based playback tool as well as a concatenated version of the selected skeleton data set. The tool was written in JavaScript and HTML5.

3 DISCUSSION & FUTURE WORK

Our web-based Skeletonotator tool proposes a solution to anonymize surveillance videos to support a crowd-sourced annotation process. Our tool generates and tracks skeletons for each occupant in the video, and lets users annotate labels to corresponding occupants without creating bounding areas on the video images. Using single or multiple selection, users can annotate both single activities, and group activities, like discussions or queuing. Skeletonotator will be open-sourced to support Ubicomp researchers in preparing datasets of human behaviours and activities for vision-based activity recognition.

We internally applied our tool to annotate group human activities within office contexts and validated that we can label human activities using body orientations, head orientations, and poses of legs and arms derived from 2D skeletons. Bench marking of our tool using ground truth data still needs to be performed in order to validate how accurate it is to annotate activities using observations based on abstract 2D skeletons only. Future work will investigate which type of human activities are specifically feasible to be annotated using our tool.

REFERENCES

[1] Federico Bartoli, Giuseppe Lisanti, Lorenzo Seidenari, and Alberto Del Bimbo. 2017. PACE: Prediction-based Annotation for Crowded Environments. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR ’17)*. ACM, New York, NY, USA, 121–124. https://doi.org/10.1145/3078971.3079020

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1302–1310.

[3] Mathias Ciliberto, Daniel Roggen, and Francisco Javier Ordóñez Morales. 2016. Exploring Human Activity Annotation Using a Privacy Preserving 3D Model. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp ’16)*. ACM, New York, NY, USA, 803–812. https://doi.org/10.1145/2968219.2968290

[4] Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. 2008. VCode and VData: Illustrating a New Framework for Supporting the Video Annotation Workflow. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI ’08)*. ACM, New York, NY, USA, 317–321. https://doi.org/10.1145/1385569.1385622

[5] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–970. https://doi.org/10.1109/CVPR.2015.7298698

[6] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. 2014. Glance: Rapidly Coding Behavioral Video with the Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST ’14)*. ACM, New York, NY, USA, 551–562. https://doi.org/10.1145/2642918.2647367

[7] Bokyung Lee, Sindy Wu, Maria Jose Reyes, and Daniel Saakes. 2019. The Effects of Interruption Timings on Autonomous Height-Adjustable Desks That Respond to Task Changes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI ’19)*. ACM, New York, NY, USA, Article 328, 10 pages. https://doi.org/10.1145/3290605.3300558