Visualizing Skiers' Trajectories in Monocular Videos

Supplementary Document

Matteo Dunnhofer, Luca Sordi, Christian Micheloni
Corresponding author e-mail: matteo.dunnhofer@uniud.it

A. Further Motivations and Details about SkiTraVis

A.1. Trajectory Points $p_t^{(t)}$

To represent the skier localization point $p_t^{(t)}$, we did not use a human skeleton parsing algorithm, such as a 2D human pose detector [9, 21], to not increase the complexity and the efficiency of the SkiTraVis pipeline. In fact, such a procedure would have required the execution of an additional deep neural network at the cost of incrementing the overall time to visualize the trajectory. Moreover, even though alpine skier-specific pose detectors have been studied [1], we found them committing displacement errors in the prediction of the human and equipment key-points close to the ground surface (e.g. for prediction feet' or skis' positions). For instance, AlphaPose [21] fine-tuned and tested, respectively, on the SkiPose2D training and test datasets [1] achieves a Mean Per Joint Prediction Error (MPJPE) of 13.2 pixels at the skier's feet and of 9.4 pixels in the upper body parts. By calculating the average pixel distance between $p_t^{(t)}$ and its corresponding reference point (i.e. the point computed following Eq. (2,3,4) on the manually annotated bounding-box) we obtain a value of 5.7 pixels, which is much lower than the errors committed by the skier pose detector. These results demonstrate that the bounding-box tracker is more robust in localizing the point $p_t^{(t)}$ in practice. Nevertheless, we believe that SkiTraVis does not require substantial changes in the pipeline to compute more semantically meaningful $p_t^{(t)}$, since 2D human pose detector could be just run on the image patches extracted from the bounding-boxes predicted by the visual tracker. We hence leave the integration of 2D pose detectors for future work. Moreover, the selection of values $k < 0.9$ (e.g. $k = 0.5$) could be used to move the trajectory toward representing the athlete's center of mass, which is of interest for biomechanical analysis [58], but it should be noted that in 2D images such a point does not fall at the same depth level in which the homography is computed. Indeed, in nearby locations to the center of mass, the homography is computed by exploiting static key-points in the background that lie on the snow's surface behind the skier. Hence, transforming the eventual $p_t^{(t)}$ with such a homography would shift the point by a pixel amount that does not respect the camera motion at the depth level at which the skier is actually localized.

B. Experimental Details

B.1. Data and Annotations

As stated in the main paper, to determine the quality of SkiTraVis's trajectories, we make use of a dedicated evaluation dataset representing real-world application scenarios. The videos belonging to this dataset appear in the test-set of SkiTB. SkiTB ("Skiers from the Top to the Bottom") is a video dataset we collected to implement deep learning-based athlete-tracking methods for different skiing disciplines. It contains 300 broadcasting videos (for a total of 392K frames) of 196 professional skiers (alpine skiers, ski jumpers, and freestyle skiers) performing their gestures from the start to the finish. Due to the high spatial extent of skiing courses, each performance is captured across several different cameras put in sequence. Each frame of the videos was manually annotated by our research team with a bounding-box containing the appearance of the athletes' bodies and equipment, following the instructions commonly used for the creation of visual tracking datasets [14, 15, 20, 31, 33, 66]. All videos are labeled with the skiing sub-discipline performed, the course location, as well as the weather conditions. 100 of such 300 multi-camera videos feature the performance of alpine skiers, where the first 60 are set as training set and the remaining 40 as test-set, considering an ordering based on the date of the competition. This setting respects the real-world condition where the learning-based algorithms are applied to test data acquired at a later time than the training data. Such a training set, which comprises 662 monocular videos and 130934 frames, has been used to fine-tune the STARK [67] tracker, while the 439 monocular videos present in the test set have been exploited as candidates for the generation of the reference trajectories as described in Section 4 of the main paper. Figure 7 gives additional examples of the situations in which was possible to obtain a reference trajectory, and hence evaluate the error committed by SkiTraVis. The virtual graphics present in the video frames have been manually labeled with bounding-boxes in order to avoid the influence of their static visual features in the execution of the camera motion estimation step. All the video frames presented in the main paper and in this supplementary document (Figures 1, 2, 3, 4, 5, 7) are taken from video clips appearing in the dataset used for validating SkiTraVis.

Other already-published datasets [1, 55] were not used

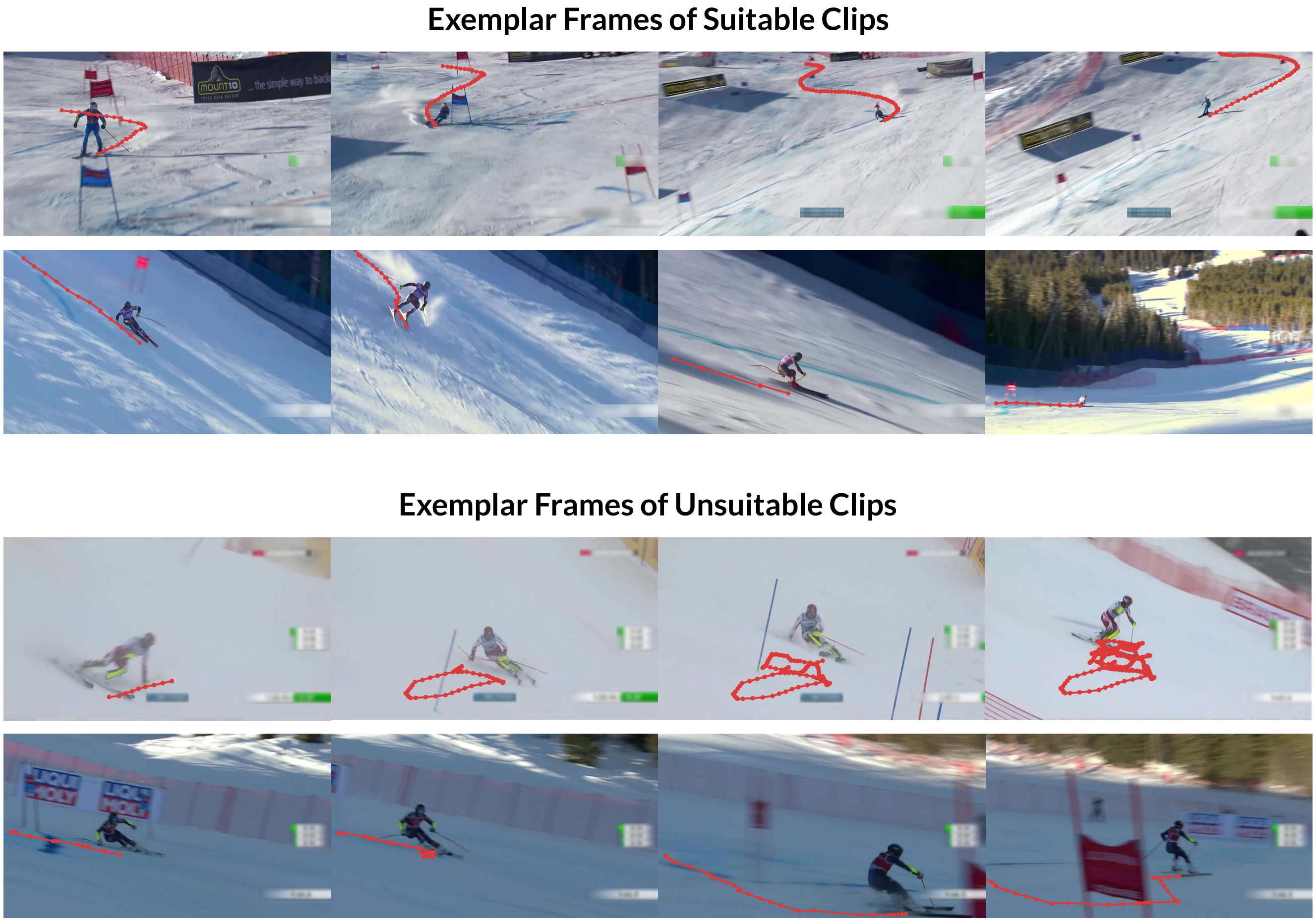
Exemplar Frames of Suitable Clips
Exemplar Frames of Unsuitable Clips

Figure 7. **Suitable and unsuitable clips for trajectory visualization.** This figure extends the plots of Figure 4 of the main paper by giving other examples of the frames where was possible to obtain the reference trajectory. As can be noticed, suitable video clips include frames where different visual features contrast with the whitish appearance of the snow and are evenly distributed over the whole frame.

for their unsuitability to the task of interest. SkiPose2D [1] provides just 2D images annotated with sparse pose annotations, thus it is not suitable for dense per-frame tracking as it is required by continuous trajectory visualization in each frame. SkiPosePTZ [55] was not employed because of its limited representation of real-world scenarios. Such a dataset was collected by capturing six alpine skiers on a single course composed of just three turning gates. Moreover, poles with markers have been densely placed on the side of the track in order to register the video frames to a geo-spatial reference system. Such marking poles, which are clearly visible in the frames, are not present in normal conditions of alpine skiing training or competition. Thus, in SkiPosePTZ, the frame matching methods for camera motion estimation would have had the opportunity to exploit visual features not present in real conditions, ultimately leading to a wrong assessment of the error committed by `SkiTraVis`.

B.2. Performance Measures

In the following, we give more information about the employed performance measures. The proposed Mean Per Point Trajectory Error (MPPTE ↓) is defined, for a video clip $\mathcal{V}$, as follows:

$$\text{MPPTE} \downarrow_{\mathcal{V}} = \frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{1}{|\sigma_t|}\sum_{i\in\mathcal{I}_t}||p_i^{(t)} - q_i^{(t)}||_2^2\Big), \qquad (9)$$

where $q_i^{(t)} \in \sigma_t$ and $\mathcal{I}_t = \{i\} \subseteq \{0,\cdots,t\} : q_i^{(t)} = [x_i^{(t)}, y_i^{(t)}], 0 \leq x_i^{(t)} < w, 0 \leq y_i^{(t)} < h$. $\mathcal{I}_t$ is the set of the indices of points $q_i^{(t)}$ that remain visible in F_t. To obtain a single score that summarizes the performance of `SkiTraVis`, the average MPPTE $\downarrow_{\mathcal{V}}$ value across all evaluation video clips is used. In simple words, in each frame, MPPTE $\downarrow_{\mathcal{V}}$ computes the Euclidean distance between the points of the reference trajectory σ_t and the corresponding ones, according to the temporal index of insertion, of the predicted trajectory τ_t. Overall, this measure gives information on the spatial distance in pixels occurring between the reference trajectory's points that are visible and the respective but predicted by `SkiTraVis`.

The MPPTE ↓ does not take into account the fact that the $p_i^{(t)}$ could be removed from τ_t because of Eq. (6). Due to such a condition, τ_t and σ_t could have different lengths since points that are discarded are not inserted back later on. Thus, to compare trajectories of different lengths we employed Dynamic Time Warping (DTW ↓) which measures the alignment of two trajectories in such conditions [50]. For a video clip $\mathcal{V}$, the score is defined as

$$\text{DTW} \downarrow_{\mathcal{V}} = \frac{1}{T} \sum_{t=0}^{T-1} dtw(\tau_t, \sigma_t) \tag{10}$$

where $dtw(\tau_t, \sigma_t)$ is the function that computes the Dynamic Time Warping distance in pixels between the predicted and reference trajectories. The overall DTW ↓ measure is obtained by averaging across all the evaluation videos.

The MSE ↓ measure used to compare $\mathbf{H}_t$ with the respective reference $\mathbf{H}_t^{(\text{LOFTR})}$, was implemented for a video clip $\mathcal{V}$ as:

$$\text{MSE} \downarrow_{\mathcal{V}} = \frac{1}{T} \sum_{t=0}^{T-1} \left(\frac{1}{9} ||\mathbf{H}_t - \mathbf{H}_t^{(\text{LOFTR})}||_2^2 \right). \tag{11}$$

As for the MPPTE ↓ and DTW ↓ measures, the average across all the video clips was employed to obtain a single score.

For further details about the AUC ↑ measure, please see [66].

B.3. Implementation Details

To obtain the fine-tuned version of STARK [67] (STARK-ft), we exploited the frames of the aforementioned training set. We used the original code provided by the authors to adapt the STARK-ST50 model pre-trained for generic object tracking. Except for the number of epochs in the stage-one training (which was set to 200), default values have been used for the hyper-parameters. Also for the SiamRPN++ [36] and SuperDiMP [2,32] visual trackers we used the code provided by the authors along with their pretrained models. MOSSE [3] instead was implemented by the pyCFtracking library [22]. Regarding the camera motion tracking estimation methods, the code provided by the authors was used to implement SuperPoint [13] and SuperGlue [51] with the pre-trained weights for outdoor conditions. The Kornia library [48] was instead used to implement LOFTR [57] (the instance for outdoor environments has been exploited). For the ST + LK [39, 53] and ORB + BF [49] methods as well as for RANSAC [23] the OpenCV implementations [4] were employed.

Table 4. `SkiTraVis`**'s performance when sharpening filters are applied.** A small filter of size 3 × 3 slightly improves the MPPTE ↓ results, while the Unsharp mask slightly decreases the quality of trajectories.

Sharpening	w/o	filter S	Unsharp mask
MPPTE ↓	12.1	11.6	12.2
DTW ↓	313	314	343

C. Additional Results

Sharpening Frames. Considering the limited availability of visual features in video frames capturing winter scenarios, we explored the impact of the following convolutional filter

$$S = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{12}$$

and of the Unsharp mask to sharpen each video frame before the key-point ST + LK detection and tracking step. The results are reported in Table 4 and show that a 3 × 3 filter slightly improves the trajectories in terms of MPPTE ↓. The other filter instead reduces the performance.

Initialization with a Skier Detector. We performed experiments to understand the impact on `SkiTraVis` of an initialization bounding-box given by a detector rather than a human annotator. In such initialization conditions, `SkiTraVis` becomes a fully automatic system that does not require any human intervention. To implement the detector, we used a YOLOv5x instance [29] fine-tuned (with default hyper-parameters) on the same SkiTB training-set used to fine-tune STARK-ft. With the initialization given by the detector, the `SkiTraVis`'s configuration with STARK-ft as skier tracker and ST + LK as camera motion estimation method achieves MPPTE ↓ and DTW ↓ errors of 15.8 and 344 pixels, respectively. The quality of the initial annotation hence influences negatively the trajectory visualization, since the two measures degrade by 3.7 and 31 pixels, respectively. We hypothesize that this behavior is due to the visual object tracker when initialized with the detector's localization. Such a noisy bounding-box prejudices the initialization step that builds models of the target skier, and in turn, such noisy models affect the localization ability of the tracker.

More Qualitative Examples. Figure 8 shows enhanced trajectory visualizations for different disciplines and application scenarios. Additional qualitative videos showing the trajectories generated by `SkiTraVis` can be reached through this link `tinyurl.com/2raemvf5`.

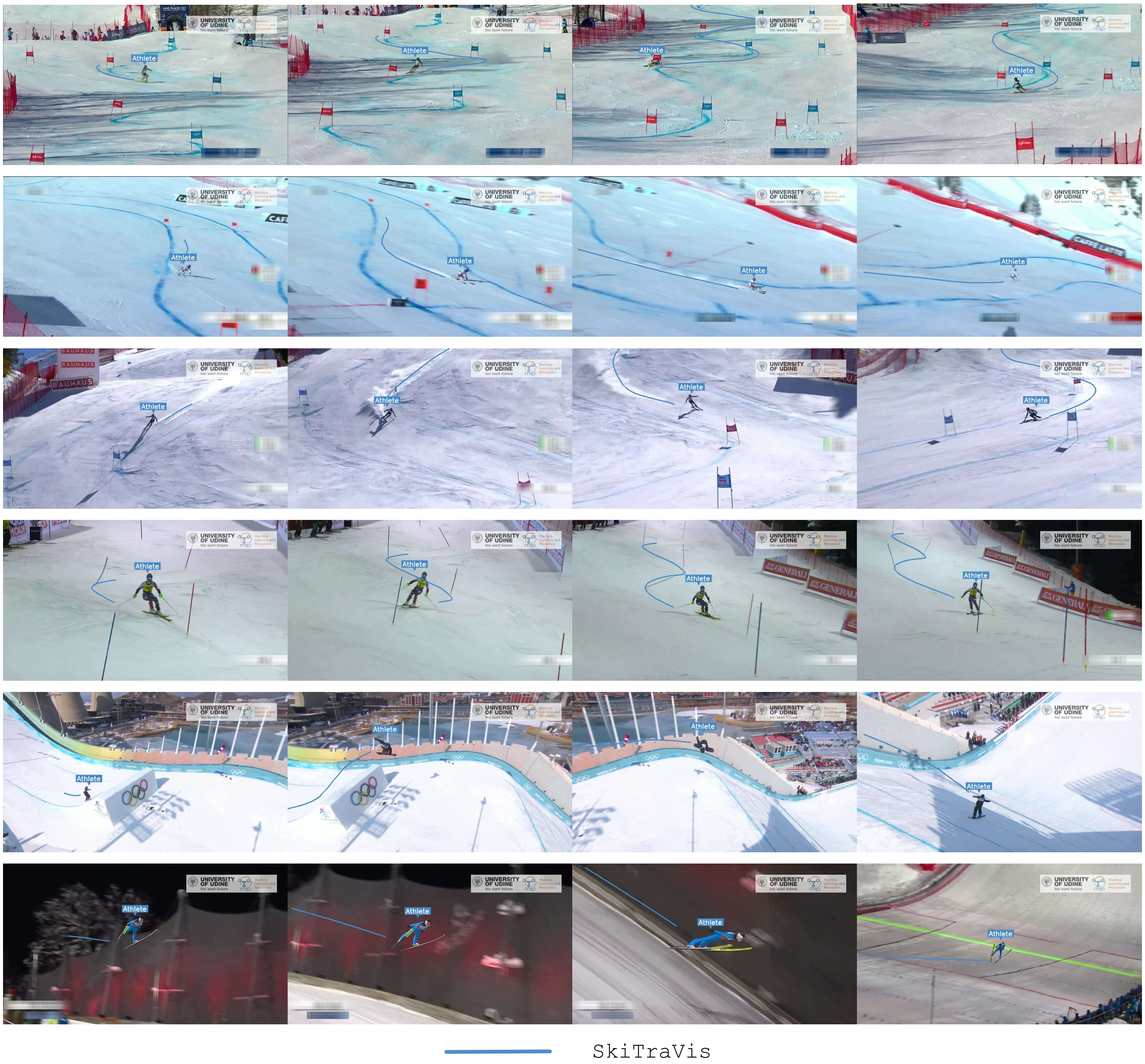

Figure 8. **Aesthetically pleasant trajectory visualizations.** This figure shows examples of enhanced visualizations based on the trajectory generated by `SkiTraVis` for alpine skiers, snowboarders, and ski jumpers. In these cases, in each frame of the video clips, the trajectories were smoothed using the Savitzky–Golay filter [52] and the trajectory points falling inside the area defined by the bounding-box b_t were filtered out.

D. Limitations and Future Work

We think `SkiTraVis` can serve as a baseline for future research on trajectory visualization and reconstruction in monocular video-based skiing performance analytics. We hypothesize that the capabilities of `SkiTraVis` could be improved by better integrating the different modules of the pipeline, and potentially through an end-to-end optimization stage of the learning modules and backbone networks involved. Datasets to train such an approach should be also investigated. The system could be also enhanced by exploiting human pose trackers instead of bounding-box ones. A human pose representation could be exploited to compute a more consistent point of contact between the athlete and the snow surface. Furthermore, the motion modeling of the different human body parts could enable the development of solutions able to simultaneously reconstruct the trajectory of disparate parts of the athlete (e.g. hands or feet) or of its equipment (e.g. skis). This direction should be investigated in parallel with research for more accurate human pose detection and tracking in skiing. Finally, we think that the better exploitation of the specific cues appearing on the slope and in training/competition scenarios could lead to an enhanced trajectory visualization. Methods that extend the applicability conditions of the proposed `SkiTraVis` to situations in which the employed LOFTR-based validation process failed, should be also studied in the future. In some contexts (e.g. for broadcasting applications), such a process could also exploit the repetitive movements of camera operators that follow different skiers during competitions on the same track to develop more accurate camera motion tracking models.