A Low-cost Approach Towards Streaming 3D Videos of Large-scale Sport Events to Mixed Reality Headsets in Real-time

Kevin Marty* Prithvi Rajasekaran† Yongbin Sun‡ Klaus Fuchs§

Auto-ID Labs MIT & ETHZ

ABSTRACT

Watching sports events via 3D- instead of two-dimensional video streaming allows for increased immersion, e.g. via mixed reality headsets in comparison to traditional screens. So far, capturing 3D video of sports events required expensive outside-in tracking with numerous cameras. This study demonstrates the feasibility of streaming sports content to mixed reality headsets as holographs in real-time using inside-out tracking and low-cost equipment only. We demonstrate our system by streaming a race car on an indoor track as 3D models, which are then rendered in an Magic Leap One headset. An onboard camera, mounted on the race car provides the video stream used to localize the car via computer vision. The localization is estimated by an end-to-end convolutional neural network (CNN). The study compares three state-of-the-art CNN models in their respective accuracy and execution time, with PoseNet+LSTM achieving position and orientation accuracy of 0.35m and 3.95°. The total streaming latency in this study was 1041ms, suggesting technical feasibility of streaming 3D sports content, e.g. on large playgrounds, in near real-time onto mixed-reality headsets.

Index Terms: Augmented Reality—Visualization—Head mounted display—Sport streaming; Deep learning—Image processing—Pattern recognition—Localization

1 INTRODUCTION

Despite allowing for increased immersion, research on capturing and streaming three-dimensional (3D) video to consumer devices has been under-discussed and not yet been adopted by content creators or developers. With the advent of augmented, virtual and mixed reality devices (altogether referred to as XR) , consuming three-dimensional (3D) video content has become more accessible than ever before. Compared to watching two-dimensional (2D) video on traditional screens, 3D video streaming on XR devices allows for increased perceived immersion in relation to the displayed content. In fact, 3D video streaming allows for content to be perceived as more vivid, salient, enjoyable and interactions as more natural compared to 2D videos [2]. In order to be perceived as enjoyable, such XR applications have to run at least 60 frames per second for a decent user experience and at least 30 frames per second as a minimum requirement to ensure stable and smooth movements of the displayed holograms [15]. A large latency in the streaming pipeline would result in a poor update of the live event and the user would miss fast-changing situations. Therefore, it is key to get the actual position of the sport agent frequently, requiring fast capturing and processing of video feeds to produce a 3D video feed in near real-time, allowing viewers to observe sports events as they unfold. Surprisingly, 3D streaming of applications or videos in real-time has been lacking focus of attention among researchers and practitioners [14]. Despite the recent developments and the rapidly increasing advances in hardware and software, the commercial breakthrough to towards mass adoption has not yet been observed, as most applications still remain rather simple prototypes [11].

A significant barrier towards capturing 3D video content of sports event is the technical requirements, as the practice requires outside-in tracking via numerous, expensive cameras, preventing most sports events to stream 3D content in real-time. Examples of 3D video applications in the sports domain include the commercial product FreeD's Replay [1]. To create volumetric replay, high tech camera equipment is necessary. For example, to stream a 3D tennis game match, 28 cameras with a 5K resolution have to be installed around the tennis court [22]. The equipment cost for 3D modeling with multi-view cameras to stream 3D sport events scales with the size of the playground. As an example, for a soccer stadium, 38 ultra-high-definition cameras are necessary to capture the entire soccer field as an outside-in tracking shown in figure 1. Unfortunately, most current approaches of modeling 3D sport events still rely on outside-in capturing of 3D video provided by multiple static cameras around the sports field.

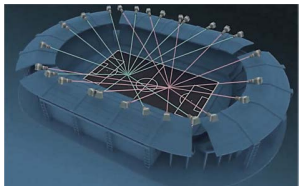

Figure 1: Outside-in tracking of a soccer stadium with 38 ultra-high-definition cameras

Figure 2: Inside-out tracking of a race track with one onboard camera mounted on the race car

Computer vision can support generating 3D videos of sports events at substantially lower costs by enabling systems to infer 3D content by leveraging the current location of players (e.g. humans, race car) from 2D video feeds. Recently, a study demonstrated feasibility of converting 2D YouTube videos of historic soccer matches into 3D videos [18], thereby not only enabling reviewing old soccer matches in 3D, but also allowing generating 3D videos at low costs. To infer the location of players on a field, a convolutional neural network is trained on 3D data, extracted from soccer video games to estimate the depth map of each player in every pose. After localizing the player on the field and estimate the pose, the trained neural network calculates the corresponding depth map. Their solution uses the field localization approach, which only works on fields with dominant visual features, such as a soccer field that has pre-defined layouts (i.e. white lines, green grass, four corners). Therefore, the approach of [18] is limited for playgrounds with dense static mono cameras around the field, not allowing for capturing sports content on larger outdoor areas, e.g. race tracks, and requiring the system to be calibrated for a specific field outline only.

*e-mail: martyk@ethz.ch
†e-mail: prithvir@mit.edu
‡e-mail: yb_sun@mit.edu
§e-mail: fuchsk@ethz.ch

978-1-7281-6532-5/20/$31.00 ©2020 IEEE
DOI 10.1109/VRW50115.2020.0-223

For larger playgrounds like a car or bicycle races, the costs for generating and streaming such 3D models with multi-view cameras would simply become unrealistically high. It is unlikely and economically unreasonable to set up the required number of cameras around such large tracks in order to capture 3D videos. Similar to generating 3D videos from 2D recordings of historic soccer matches through computer vision [18], we propose to create low-cost 3D models from the single, but non-static, inside-out tracking player-based point-of-view camera. This study's approach, therefore, demonstrates an inside-out tracking with only one camera, mounted on the race car as shown in figure 2. This paper provides a solution for streaming large-scale sports events, which is sparse or not fully covered with cameras around the playground such as racing tracks, alpine skiing or public streets [5, 26]. Our solution uses only one onboard camera to localize the sports agent, for example, a race car as seen in figure 2. The camera is attached to the sports player and is a standard mono camera, which is widely used in mobile phones or GoPros. With only one onboard camera per sports agent, our approach does not depend on the size of the sports playground, but on the number of players participating in the event. To improve the localization robustness of the sports agent, we use deep learning technology and compare several state of the art convolutional neural networks (CNN) in regards to their accuracy performance and observed latency. Harsh environments like big illumination changes, blurred images or big viewpoint changes are handled better with deep learning-based localization instead of feature-based localization. To handle the deep learning drawback of large computing power demand for fast execution, we run our model on a cloud instance which delivers computing power on-demand. Our system includes two data streams: 1) Camera video stream from the sports agent to the cloud instance, 2) the predicted position vector of the sports agent from the cloud instance to the mixed reality headset.

In this paper, we propose a novel computer vision based system that infers a players position on a playground in near real-time via a single player-mounted inside-out camera. In the study, we therefore primarily focus on the localization problem and the related latency. Our experiments indicate a clear relation between position accuracy and calculation time. However, the most accurate position calculation is still 10x faster than the video data stream. We provide a first demonstration of a remote-controlled race car, driving around a small race track and stream the event to the augmented reality headset. The main contributions of our work are summarized as below:

- Novel end-to-end pipeline to stream sports events practiced on large-scale playground with low-cost equipment
- Comparison between different deep learning architectures in their position and orientation accuracy as well as in their execution time
- Time analysis between the position calculation and the streaming latency

2 Related Work

2.1 Sport Analysis

Computer vision (CV), convolution neural networks (CNN) and augmented reality (AR) are extensively used in academia [10, 19] as well as in commercial applications [1, 16] for sports analysis. Analyzing sport content includes many tasks like segmentation, localization, detection or environment modeling. Because most of the sports events are practiced in stadiums, the environment is static and therefore can be pre-modeled with RGB-D cameras [29] or with visual odometry technique [6]. To capture free-viewpoint navigation or 3D replays in small playgrounds including soccer, football or basketball, multiple high tech cameras are used. Intel FreeD provides a commercial solution for streaming volumetric data (voxel), which can be used for sports scene reconstruction in 3D [25]. With this outside-in tracking method, 38 cameras are necessary to capture a soccer game. To cover a larger playground like a car race track would require more expensive high tech cameras. A different approach uses 2D YouTube videos to create 3D videos at low costs by applying deep learning technology to estimate the 3D shape of the sports agent [18]. This approach reduces the number of necessary cameras to create 3D videos. However, it relies still on the outside-in tracking method and requires multiple cameras around the field to track the individual sports agents.

In our approach, we use a player-mounted inside-out camera as an input stream to localize the sports player which in our experimental setup is a race car. Localization the camera is equivalent to localize the race car's position. The position vector can then be used to virtual place the pre-modeled car in the virtual environment. Our approach requires much less equipment (one camera per sports player) and is hence less expensive for large scale playgrounds.

2.2 Feature Based Localization

The localization problem has been studied extensively in the last couple of years. One solution takes geotagged images which are stored in a database [3]. Comparing the stored images with the query image by using different retrieval techniques gives the approximated position of the query image. A more accurate method to localize the camera is to use the Structure-from-Motion (SfM) technique [21]. SfM estimates all 6 degrees of freedom (DOF) by extracting and describing features with algorithms including the scale invariant feature transform (SIFT) or binary robust independent elementary features (BRIEF), of the query image and calculate the similarity [27] to the keyframe features. Random sample consensus (RANSAC) is used to reduce the outliers and the PnP or direct linear transformation (DLT) algorithm estimates the camera pose. The accuracy heavily relies on the extracted features. Therefore the SfM fails in texture less environment, blurred images, strong illumination change or strong viewpoint changes.

2.3 Machine Learning Based Localization

Considering image retrieval, convolutional neural network are competitive with the feature-based methods [17]. Depending on the application, CNN shows great potential to handle difficult lighting or motion blur better than the traditional technique. Kendall presents the first end-to-end, real-time pose estimation with a convolutional neural network [12]. The paper regresses a 6-DOF pose of the camera, using the spatial information of an RGB image. Based on [12], an extended version is using the additional temporal information along with the spatial information including a Long-Short-Term-Memory (LSTM) [7, 8].

3 System Design

In this paper, we propose a new system to stream large playground sports events to mixed reality headsets in near real-time. Our solution uses a single mobile player-mounted inside-out camera instead of a magnitude of fixed pre-installed, static outside-in cameras. We use convolutional neural network (CNN) models to infer the location and orientation of the movements of the sports agent. The camera feed is uploaded to a cloud-based computing infrastructure to leverage high-speed computing on-demand for the CNN models in real-time. A system overview is shown in figure 3. The camera that is mounted on the sports agent (i.e. race car) with the mounted onboard camera (race car in real-world perspective) streams images (onboard perspective) to the cloud instance. The CNN model is

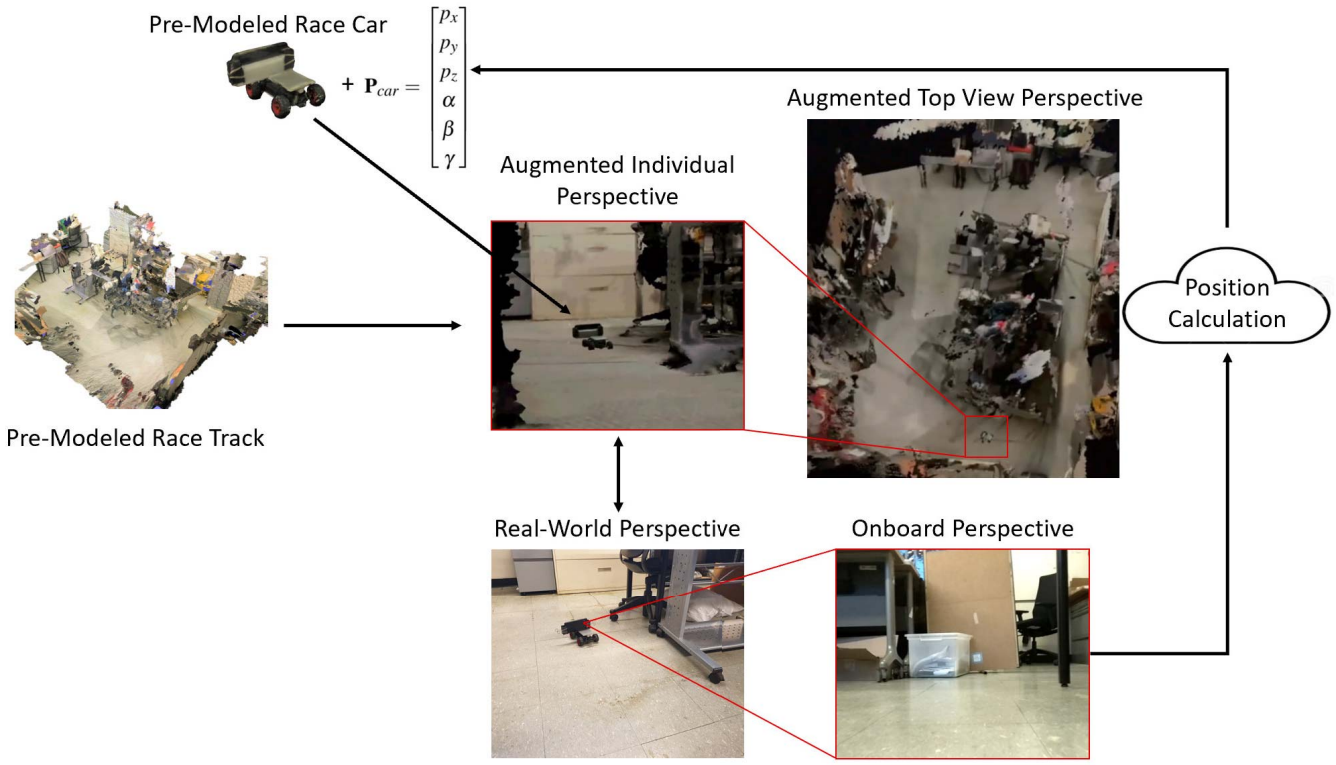

Figure 3: Streaming pipeline for live sports events to mixed reality headset. The pre-modeled player is placed in the pre-modeled environment as seen in the augmented individual perspective. The spectator can choose any individual perspective, including the top view perspective. The correspondent real-time, real-world player is shown in the real-world perspective. The onboard perspective is used for the position calculation, running on the cloud instance. The output is the position vector $\mathbf{P}_{car}$

running on the cloud infrastructure and calculates the position vector $\mathbf{P}_{car}$. The pre-modeled sports agent (race car) is placed in the pre-modeled environment (race track). The sport event (car racing) is rendered on the mixed reality headset and can be watched in different angles including top view (augmented top view perspective) or any individual perspective (augmented individual perspective). Similar to related study [28], we benchmark four different CNN architectures in their localization accuracy as well as in their execution time. We also analyze the streaming latency. Decomposing the total streaming pipeline in three components: 1) T1 video stream, 2) T2 position calculation and 3) T3 vector streaming, gives a more detailed inside over the total streaming latency. An overview of the streaming latency analysis can be seen in figure 4.

3.1 T1 Video Stream

Our approach uses one onboard camera, mounted on the sports agent. The camera is a low-cost RGB mono camera, oriented in the direction of translation. Therefore, localize the moving camera is equivalent to localize the sports player. The number of used cameras depends on the number of sports players and not on the size of the playground and for that reason, it is especially interesting for large-scale sport disciplines such as Formula 1, Tour de France or marathons.

The player-mounted camera uploads the video stream through a website. The frame rate of the streaming video is 30 images per second. We use a WebSocket protocol to send the images from the camera to the cloud instance, where our deep learning model calculates the player's position. The images, streamed from the

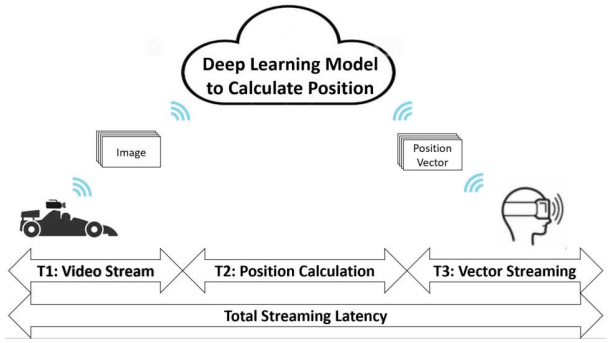

Figure 4: Overview of the streaming latency analysis

camera are stored in a memory buffer on the cloud instance. The CNN model takes the input image from the memory buffer as soon as it finishes with the latest position calculation. The number of images stored in the memory buffer depends on the camera frame rate, image upload speed and the execution time of the chosen deep learning model.

3.2 T2 Position Calculation

Convolutional neural networks require a lot of computing power for training and fast execution. Usually, a graphics processing unit (GPU) is necessary to achieve rapid computing time for real-time applications. Latest mixed reality headsets, as well

as current used onboard cameras, do not have the hardware capacity to run very deep CNN on their device. Therefore, to provide enough computing power and to ensure scalability of the solution, our deep learning model runs on cloud infrastructure.

The CNN model estimates the position $\mathbf{P}$ of the sports player, based on the input image I. The vector $\mathbf{P}_i = [\mathbf{p}_i, \mathbf{q}_i]$ contains the relative position vector $\mathbf{p}_i \in \mathbb{R}^3$ (equation 1) and rotation vector $\mathbf{q}_i \in \mathbb{R}^4$ (equation 2). The rotation vector $\mathbf{q}_i$ is represented in quaternions because of the singularity problem or better known as gimbal lock.

$$\mathbf{p}_i = p_x \cdot \mathbf{e}_x + p_y \cdot \mathbf{e}_y + p_z \cdot \mathbf{e}_z \tag{1}$$

$$\mathbf{q}_i = q_r + q_i \cdot \mathbf{i} + q_j \cdot \mathbf{j} + q_k \cdot \mathbf{k} \tag{2}$$

To calculate the absolute position $\mathbf{P}_{abs,i}$ with respect to the world coordinate system, we transform the quaternions to the rotation matrix $\mathbf{R}_i$ and multiply it with the relative position vector $\mathbf{p}_i$ as seen in equation 5.

$$\mathbf{R}_i = \begin{bmatrix} 1-2s(q_j^2+q_k^2) & 2s(q_iq_j-q_kq_r) & 2s(q_iq_k+q_jq_r) \\ 2s(q_iq_j+q_kq_r) & 1-2s(q_i^2+q_k^2) & 2s(q_jq_k-q_iq_r) \\ 2s(q_iq_k-q_jq_r) & 2s(q_jq_k+q_iq_r) & 1-2s(q_i^2+q_j^2) \end{bmatrix} s = ||\,\mathbf{q}\,||^{-2} \tag{3}$$

$$\mathbf{p}_i = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{4}$$

$$\mathbf{P}_{abs,i} = \mathbf{R}_i \cdot \mathbf{p}_i \tag{5}$$

3.3 T3 Vector Streaming

We make the assumption, that the race track environment does not change dramatically over the time of the event (e.g. during a race) and therefore can be considered to be static. Therefore, we pre-modeled the environment as shown in figure 6. For the race car, we assume the car to behave as a rigid body and pre-modeled the entire car as illustrated in figure 7.

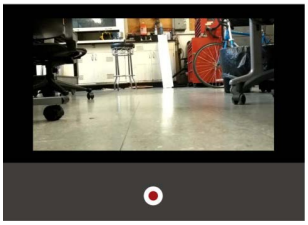

Figure 5: Screenshot of the streaming webpage, running on the player-mounted camera

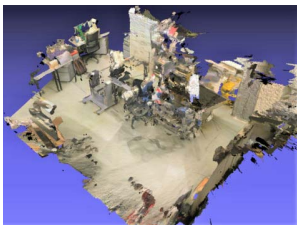

Figure 6: Virtual model of the research lab which is used as an indoor race track

The mixed reality headset (i.e. Magic Leap One) is connected to the internet via a wireless network connection. The headset sends hypertext transfer protocol (HTTP) requests to the cloud instance to get the position vector $\mathbf{P}$ of the player. The placement of the virtual player in the virtual environment, as well as the holographic rendering, is directly executed on the mixed reality headset itself.

4 Experimental Setup

4.1 3D Modeling and Video Streaming

To validate our proposed approach towards 3D streaming of sports events, practiced on large-scale fields, we chose to emulate a racing scenario, similar to Nascar races via an indoor round race track inside an office building. We make the assumption, that the race track and the body of the race car do not change over time and therefore can be assumed as rigid bodies. For the race track, this assumption is valid. However, for the race car, this is a strong assumption because the moving wheels while driving and steering violate the rigid body assumption. But considering the car model resolution, the rigid body assumption for the car does not decrease the user experience significantly.

To validate the localization accuracy and computing time of our CNN models as well as the streaming latency, we set up a small indoor race track in our lab with a remote-controlled race car of scale 1:16 (figure 8) driving around. On top of the race car, we mount a mobile phone, Xiaomi Redmi 6Pro, which is used as the onboard camera. As a race track, we chose an 18 meters long, rectangular path in our lab. The race track and the race car are pre-modeled with the low cost depth sensor from Structure Sensor (figure 7 and figure 6).

A streaming website as seen in figure 5, runs on the mobile phone to stream the onboard view of the race car. The website is hosted on the closest Amazon Web Service (AWS) cloud server. It is the equivalent instance as our deep learning model is running. We use the WebSocket protocol to send the images from the onboard camera to the cloud server. In our latency analysis, the segment T1 measures the time it needs to send an image from the onboard camera to the input of our deep learning model.

Figure 7: Virtual model of the race car

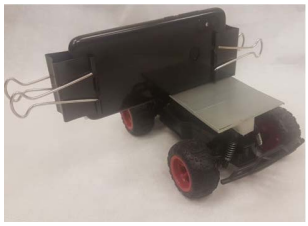

Figure 8: Race car with mounted onboard camera

4.2 Position Estimation with different CNN Models

To estimate the position of our race car, we use the CNN-based approach. Similar to [28], we estimate the position of the race car with an end-to-end convolutional neural network. The input for the CNN model is an image from the onboard camera and the output is the position vector $\mathbf{p}$ and orientation vector $\mathbf{q}$. The initial position is a distinctive position, similar to the start grid in a real car race.

In our experiments, we compare four different CNN architectures in their position and orientation accuracy as well as in their computational time. The network architecture can be split into three different main steps:

- extracting spatial information
- extracting temporal information
- regression

An overview of the convolutional neural network architecture is shown in figure 10.

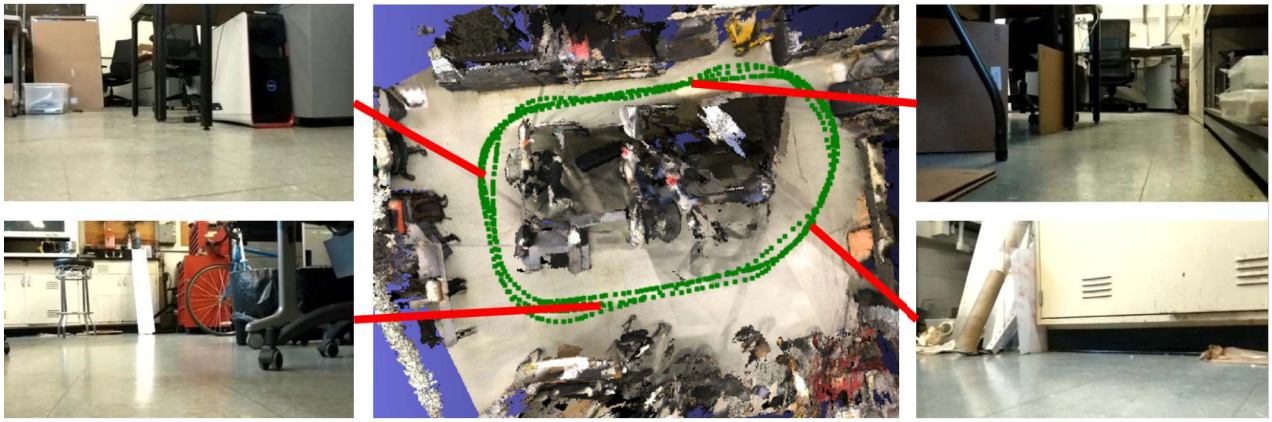

Figure 9: Top view of the research laboratory which is used as an indoor race track with the position labels (ground truth) in green and four correspondent onboard perspectives

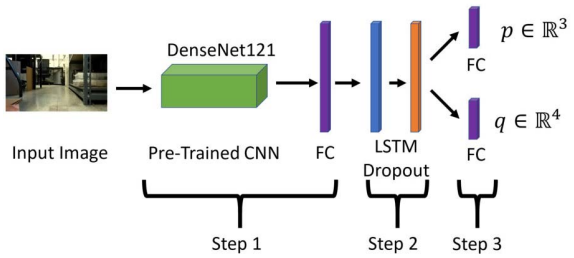

Figure 10: Deep neural network architecture

4.2.1 Step 1: Spatial Information

Many different CNN are available to extract spatial information. We compare four different architectures: GoogleNet [23], InceptionV3 [24], MobileNet [20] and DenseNet121 [9]. These networks are popular ones for features extraction tasks. Training those models would require a large training set with corresponding ground truth. For practical reasons, we use transfer learning. All the used models are pre-trained on the ImageNet data set. The models are fine-tuned by reshaping the last fully connected (FC) layer to have an output vector of 2048 dimensions. The output vector can be seen as a unique description of the input image.

4.2.2 Step 2: Temporal Information

In a sequence of input images, recent works show an improvement by using the temporal information as an additional resource [4]. Similar to [4], we use a long-short-term-memory (LSTM) layer with hidden size 1024 after step 1 to capture the temporal information. During training, after the LSTM layer, an additional dropout layer is used with a dropout rate of 0.5.

4.2.3 Step 3: Regression

The third step contains two dense layers. One dense layer for the output relative position vector **p** and one dense layer for the output rotation vector **q**.

The required input format for the pre-trained CNN models are 3x224x224 for the DenseNet121, MobileNet, and PoseNet and the InceptionV3 3x299x299. Therefore, we scale the image from the live stream with format 3x1920x1080 down to 3x456x256 for DenseNet121, MobileNet and PoseNet and 3x568x320 for InceptionV3. For the correct input format, we apply a random crop on the downscaled images.

To train the whole end-to-end deep neural network, which maps an input image I to the position vector $\mathbf{P} = [\mathbf{p}, \mathbf{q}]$, we use the Adam optimizer [13] with the L2 loss function from [28]:

$$L_i = ||\mathbf{p_i} + \hat{\mathbf{p}}_\mathbf{i}||_2 + \beta \cdot ||\mathbf{q_i} - \frac{\hat{\mathbf{q}}_\mathbf{i}}{||\hat{\mathbf{q}}_\mathbf{i}||}||_2 \quad (6)$$

p and **q** are the ground truth and $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ the correspondent estimated position and orientation. Because of the singularity problem, we represent the orientation in quaternions and therefore normalize the predicted orientation to unit length. β is a hyper-parameter which relates the orientation error to the positional error. Because our car object has with the no-slipping constrain only one rotational degree of freedom, we weight the orientation error smaller than the position error and therefore set β equal to 0.33.

The position calculation is running on an AWS cloud server. The model runs on an EC2 instance with GPU power. The instance has a 2.3GHz Xeon E5-2686 v4 Processor and two NVIDIA Tesla M60 graphic units with each 8GB memory. For measuring the computing time T2 in our latency analysis, we estimate the executing time for the position calculation function with one image I as an input argument and the correspondent output vector $\mathbf{P} = [\mathbf{p},\mathbf{q}]$.

After the position calculation, the position vector $\mathbf{P} = [\mathbf{p},\mathbf{q}]$ is send with the hypertext transfer protocol (HTTP) to the mixed reality headset. We use the Magic Leap One headset to render and augment our virtual race track with the virtual race car on it. The time it needs to send the position vector from the cloud server to the mixed reality headset is named T3 in the latency analysis. The headset is via the wireless local area network (WLAN) connected to the internet.

4.3 Dataset

In order to train the computer vision pipeline and in particular the four CNN models (PoseNet, DenseNet121, InceptionV3, MobileNet), a labeled dataset is required. In fact, compared to other classification labeling tasks, position labeling is even more effort intensive and therefore expensive and sometimes not even a suitable tasks for humans. For training the end-to-end model, we therefore collect nine laps on the indoor race track. With a frame rate of eight images per second, we collect 1680 images in total for training. A

structure from motion (SfM) pipeline is used to calculate the correspondent position vector from the images. This position vector is then used as the ground truth for training the deep learning frameworks. An additional three laps represent the test set to evaluate the different models in their accuracy and respective computing time. Figure 9 shows the race track in a top view with the position labels in green and four onboard images.

5 Results

Table 1 shows the quantitative results for the different model accuracy in position and orientation. The presented position and rotation accuracy is the median value of the loss function L evaluated on the test dataset with 630 images. Figure 12 illustrates the qualitative results for the DenseNet121 + LSTM model, evaluated on the test data set.

Models	Position Accuracy	Orientation Accuracy
PoseNet + LSTM	0.41m	6.3°
DenseNet121 + LSTM	**0.35m**	**3.95°**
InceptionV3 + LSTM	0.49m	6.71°
MobileNet + LSTM	0.45m	7.79°

Table 1: Position and orientation accuracy for different CNN models

Three different scenes of our system can be seen in figure 11. Each line illustrates one scene with four different perspectives. The right column is the onboard perspective of the race car. The third column from the left is what a spectator sees if he attends the live event. At this stage, the spectator can only see one specific perspective of the whole track without moving around. The two columns on the left show the augmented race track and race car in different perspective. The spectator can see an overview of the race in the second column from the left or one specific part of the race track as seen in the left column. The augmented model can be rotated, shifted or zoomed to any individual perspective the spectator prefers to watch.

The computational time for the different models on the cloud instances is shown in column one in table 2. The last column represents the size of the weight file.

Models	Computational Time	Model Size
PoseNet + LSTM	**42ms**	**68Mb**
DenseNet121 + LSTM	94ms	478.3Mb
InceptionV3 + LSTM	91ms	221.4Mb
MobileNet + LSTM	69ms	125.8Mb

Table 2: Computing time for one position vector for different CNN models

In table 3, the results for the video stream latency T1, the position calculation T2 and the vector streaming latency T3 are presented. The last column in table 3 summarizes the total streaming time (T1+T2+T3) in milliseconds.

Models	T1	T2	T3	Total
PoseNet + LSTM	550ms	42ms	449ms	**1041ms**
DenseNet121 + LSTM	550ms	94ms	449ms	1093ms
InceptionV3 + LSTM	550ms	91ms	449ms	1090ms
MobileNet + LSTM	550ms	69ms	449ms	1068ms

Table 3: Latency analysis for the whole streaming pipeline for the different models

6 Discussion

In this paper, we propose a novel computer vision based approach towards streaming 3D video feeds of sports events in near real-time. In particular, we propose that sports events practiced on the large fields can be recorded in 3D by using a single player-mounted inside-out camera. This approach yields potential over contemporary approaches that are either expensive as they require a magnitude of static outside-in cameras or are impractical as they require a static court with pre-fixed visual references (e.g. soccer field). The main contribution of this study, therefore, lies in demonstrating that convolutional neural networks (CNN) can be used for inferring a player's relative position to a pre-modeled race track in order to generate a 3D video feed of an ongoing race event within 1041ms, indicating that near real-time 3D video streaming of large-field sports events via a player-mounted camera is possible. More concretely, by comparing four different CNN models in their respective performances of estimating the agents' position and orientation accurately as well as their required computational times required for location inference indicate that PoseNet+LSTM seems to be the fastest and DenseNet121+LSTM the most accurate architecture choice for such a task.

In addition, we decomposed the total streaming latency in three different segments (T1: Video Stream, T2: Position Calculation, T3: Vector Streaming) and evaluated each segment and the total streaming pipeline in their latency time. Considering the player's localization accuracy, the DenseNet121+LSTM outperforms the other CNN models with an achieved accuracy of localizing the player within 0.35m in position and 3.95° in orientation accuracy. DenseNet121+LSTM is a very robust solution with only a few outliers as seen in figure 12. PoseNet+LSTM has shown a slightly worse accuracy with respect to inferring player position and orientation compared to DenseNet121+LSTM. The most inaccurate model in position is InceptionV3+LSTM with an observed accuracy deviation of 0.49m. In respect to orientation, MobileNet+LSTM has inferred orientation least accurately with 7.79° from the true orientation. Overall, the position and orientation calculations of all CNN models seem accurate enough to provide a decent user experience in the augmented perspective. Taking into account the differences in execution times for inferring the player's position vector from the on-board image between the different CNN models, a clear dependency of model size and execution time can be observed. This seems obvious, as the model size corresponds to the number of weights stored in each model's file. PoseNet+LSTM is by far the fastest and lightest model with 42ms execution time and model size of just 68Mb. With 42ms, PoseNet+LSTM would be able to calculate the position in near real-time with a frame rate of 24 frames per second, indicating that a decent user experience is possible. MobileNet+LSTM is a slightly CNN model with 125.8Mb in size and already is 27ms slower than the PoseNet+LSTM. The longest execution time was observed with the DenseNet121+LSTM at 94ms, being equivalent to a frame rate of 11 frames per second and a correspondent weight file of 478.3Mb. Although the model size of PoseNet+LSTM is seven times smaller compared to DenseNet121+LSTM, the execution time is just two times faster.

For practical reasons, our system was tested on a 1:16 down-scaled setup, within an indoor research laboratory. The 18 meters test track would be equivalent to a 288 meters test track in a full-size layout, similar to roughly a third of a NASCAR race track. Despite the usage of a smaller setup for the purpose of this study, the streaming latency for a full-size layout and a down-sized layout can be expected to be in the same dimension. The accuracy in position and orientation for the full-size setup are comparable to the results that were observed in the study's down-scaled experiments. Therefore, the player-mounted camera-based approach can also be

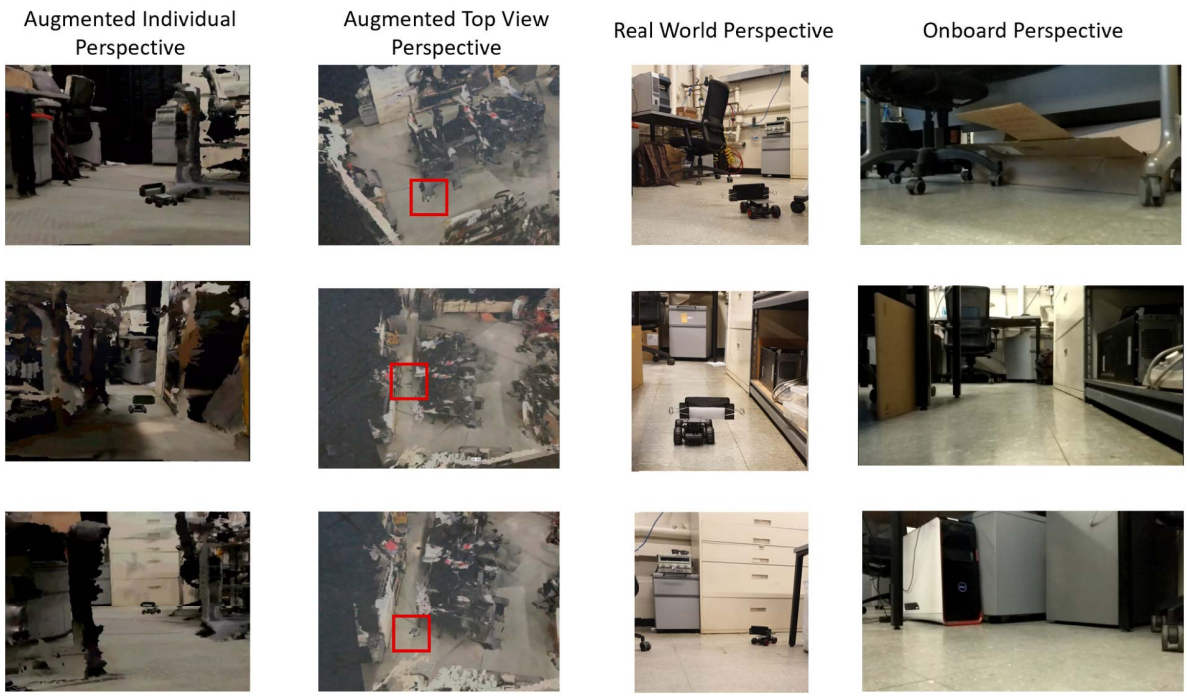

Figure 11: Each line shows a different scene from our system in four different perspectives. Left two columns: View through the mixed reality headset as a spectator; third column from left: Real world view of the race car; right column: Onboard camera view

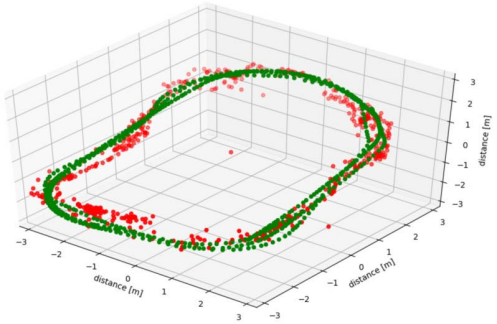

Figure 12: Calculated positions with the model DenseNet121+LSTM (red) versus the ground truth (green)

used for very large playgrounds like Formula 1 race tracks, given that visual features allow for inference of a player's position on the reference model. Analyzing the latency for the streaming pipeline clearly shows the bottleneck of streaming live sports events on large playgrounds. With an image upload time of 550ms, video streaming has the highest latency in our pipeline. To send the position vector from the cloud server to the mixed reality headset needs 449ms which is around ten times slower than the position calculation with PoseNet+LSTM. Vector streaming is 1.22 times faster than video streaming, although the size of the position vector is 8290 times smaller than the streamed image. The total streaming latency with the fastest CNN model is 1.041 seconds (i.e. PoseNet+LSTM). This 1.041 seconds streaming latency causes a delay in the position transmission and is a limitation in the frequent localization updates of the sports agent. As a consequence, the spectator does not observe what is happening between the first 1.041 seconds of an event. Depending on the sports player's speed and on the scale of the event, this streaming latency can cause an inferior user experience. Generally, the computing time for the position vector is negligible compared to the streaming of the images or the position vectors. For this reason, we implemented the DenseNet121+LSTM which gives the best localization accuracy and user experience in our demo.

The inside-out tracking approach demonstrated an economically superior, alternative solution to streaming large-scale 3D sports events by using a single agent-mounted camera. The possibility to place yourself in any individual perspective or watch a race in a top view creates new use cases for augmented as well as for virtual reality. The approach suggested in this paper, therefore, is especially interesting for non-professional sports disciplines and represents an opportunity to extend 3D streaming of sports to more inclusive domains (e.g. less financially affluent leagues or disciplines). Complementary to this paper, a demo video will be published to demonstrate the enriched experience for a spectator, watching a car race through the Magic Leap One headset.

7 Future Work

Although this study demonstrates the feasibility of streaming 3D renderings of large-scale sport events in near real-time, the results need to be understood as being subject to several limitations. First, the generalizability to other race tracks and sports arenas needs to be demonstrated. While it could be that similar race tracks could work equally well, sport arenas with limited visual feature variety might prove more challenging for inference of accurate player position from first-view inside-out tracking cameras. In addition, to further lower the latency in video streaming, further investigation in video compression has to be done. Third, faster network connections (e.g. 5G) are very promising solutions towards lowering the streaming latency of 3D video feeds. Therefore, the system proposed in this study might even show higher performance in the future given increased connectivity. Fourth, further research in combining different localization sensors (including wheel odometer,

differential global positioning systems) would increase the position and orientation accuracy and make it even more robust against anomalies. Another source of data for sports practice on streets like Tour de France could also be street-based image material, e.g. Google Street View. Fifth, investigation in dynamic modeling of 3D shapes instead of using one rigid body would increase the user experience, as the view could be customized based on a current perspective. For example, for video games (e.g. Formula 1), the user experience could be further increased by using the car model assets and race tracks from the video game directly. Finally, video games could also be used to generate more training data for the convolutional neural network.

As most of the limitations and future work packages would rather strengthen the accuracy of the computer vision framework and lower the latency observed in this study, it can be expected that the demonstrated performance of the proposed system could be further improved in the future, given improved accuracy from sensor fusion and increased connectivity.

REFERENCES

[1] A.J.McCarthy. *See the Groundbreaking Replay Technology Thats Coming to Professional Tennis*. url=https://slate.com/technology/2014/03/atp-s-new-replay-technology-freed-provides-360-degree-views-of-all-the-action.html, Accessed:2019-10-02.

[2] D. Anton, G. Kurillo, and R. Bajcsy. User experience and interaction performance in 2d/3d telecollaboration. *Future Generation Computer Systems*, 82:77–88, 2018.

[3] R. Arandjelović and A. Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pp. 188–204. Springer, 2014.

[4] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pp. 154–159. Springer, 2010.

[5] J. Dachman. *NBC Sports Tour de France Coverage Adds Augmented Reality, Live On-Bike POV Cameras*. url=https://www.sportsvideo.org/2019/07/19/nbc-sports-tour-de-france-coverage-adds-augmented-reality-live-on-bike-pov-cameras/, Accessed:2019-11-09.

[6] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968. Ieee, 2011.

[7] F. W. C. Hazirbas, L. L.-T. T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization with spatial lstms.

[8] T. Hu and H. Sun. Video stream relocalization with deep learning. In *2018 5th International Conference on Systems and Informatics (ICSAI)*, pp. 109–114. IEEE, 2018.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[10] Y. Jing. Vr, ar, and wearable technologies in education: An introduction. In *Handbook of Mobile Teaching and Learning*. Springer Verlag, 2019.

[11] H. Kaufmann and A. Dünser. Summary of usability evaluations of an educational augmented reality application. In *International conference on virtual reality*, pp. 660–669. Springer, 2007.

[12] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] E. Lamboray, S. Wurmlin, and M. Gross. Real-time streaming of point-based 3d video. In *IEEE Virtual Reality 2004*, pp. 91–281. IEEE, 2004.

[15] Microsoft. *Hologram Stability*. url=https://docs.microsoft.com/en-us/windows/mixed-reality/hologram-stability, Accessed:2019-10-02.

[16] Nanalyze. *7 Sports Technology Companies for Sports Analytics*. url=https://www.nanalyze.com/2017/03/sports-technology-analytics-companies/, Accessed:2019-10-02.

[17] U. Özaydın, T. Georgiou, and M. Lew. A comparison of cnn and classic features for image retrieval. *arXiv preprint arXiv:1908.09300*, 2019.

[18] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz. Soccer on your tabletop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4738–4747, 2018.

[19] H. Saito, N. Inamoto, and S. Iwase. Sports scene analysis and visualization from multiple-view video. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, vol. 2, pp. 1395–1398. IEEE, 2004.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[21] M. R. U. Saputra, A. Markham, and N. Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):37, 2018.

[22] S. Staff. *Replay Technologies freeD 360 Replay Brings New Perspective to US Open Tennis*. url=https://www.sportsvideo.org/2015/09/11/replay-technologies-freed-360-replay-brings-new-perspective-to-us-open-tennis/, Accessed:2019-10-02.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[25] I. Technologies. *See More Game Than Ever*. url=https://www.intel.com/content/www/us/en/sports/technology/true-view.html, Accessed: 2019-10-02.

[26] V. TURK. *Formula 1 Refuels To Take On The World Of Entertainment*. url=https://www.gqmiddleeast.com/culture/formula-1-refuels-to-take-on-the-world-of-entertainment, Accessed:2019-11-09.

[27] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE, 2007.

[28] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE, 2007.

[29] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, vol. 37, pp. 625–652. Wiley Online Library, 2018.