Markerless Augmeted Advertising for Sports Videos

Hallee Wong[1] Osman Akar[2] Emmanuel Antonio Cuevas[3] Iuliana Tabian[4] Divyaa Ravichandran[5] Iris Fu[5] Cambron Carter[5]

[1]Williams College [2]University of California, Los Angeles [3]Universidad de Guanajuato [4]Imperial College London [5]GumGum Inc.

Objectives

We propose a novel system for augmenting sports videos that:

1. Automatically identifies viable **crowd** regions in a video clip
2. Places the asset with a **natural** perspective consistent with the scene
3. Does not require the presence of a known calibration object (e.g. standard field markings)
4. Is fully automatic

Related Work

- Existing systems for specific use cases (tennis [1, 2], soccer [3], baseball [4]) rely heavily on the presence of known, reliable markings on the ground and a rigid sporting-ground structure

Overview

The video augmentation system involved two steps:

1. Automatic augmentation of a static seed video frame
2. Tracking across video frames to place the asset consistently

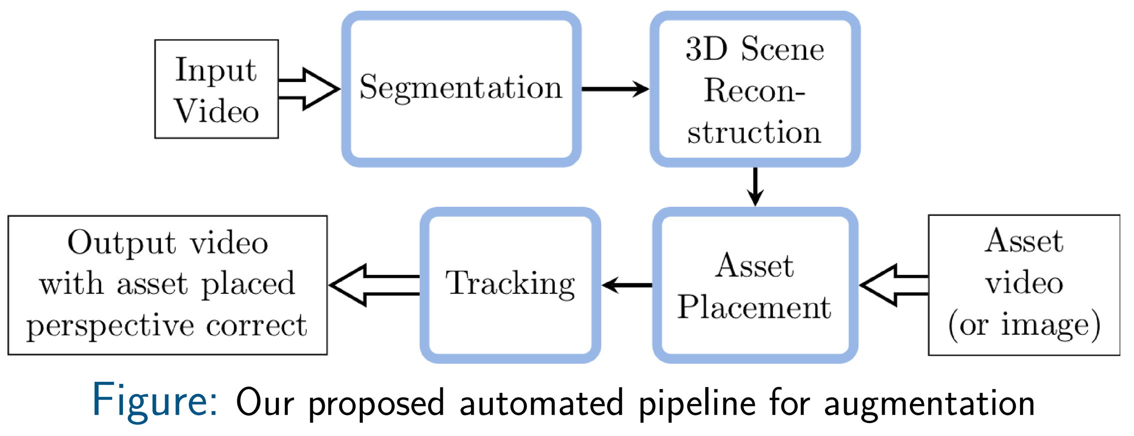


Figure: Our proposed automated pipeline for augmentation

Assumptions

- The input video is captured from a single viewpoint by a monocular RGB camera
- Intrinsic camera parameters may not be known
- The asset to be overlayed in the video will be a 2D image/video
- The shot(s) need not be static (i.e. may change location or angle)

Crowd Segmentation

1. **Sample static images** from the input video at the desired number of intervals
2. Use Pyramid Scene Parsing Network (PSPNet) [5] to **segment** areas with "person" and "grandstand" texture in each frame
3. Choose the frame with the lowest **Segmentation Quality Score (SQS)** as the seed frame

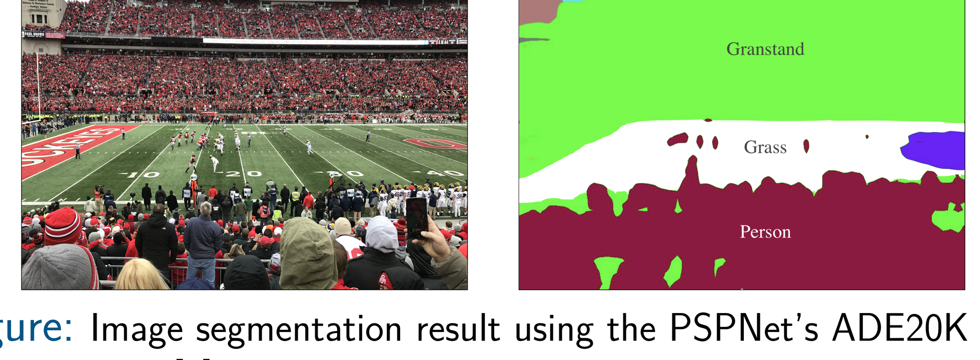


Figure: Image segmentation result using the PSPNet's ADE20K trained model [5]

Segmentation Quality

The Segmentation Quality Score (SQS),

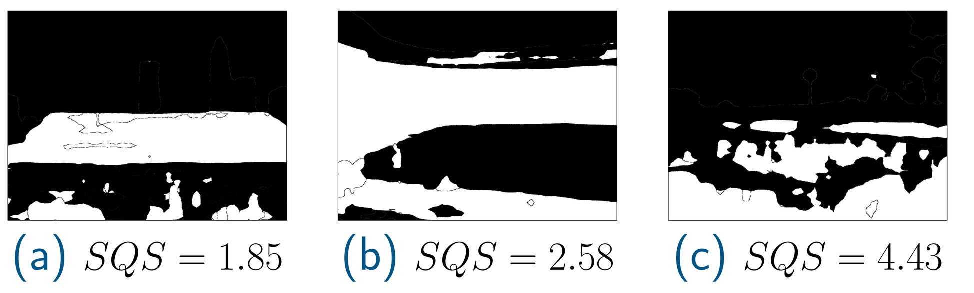
$$SQS = S_{cl} \cdot S_{cp} \cdot S_{sp} \ ,$$

quantifies whether the segmented crowd area:

(a) is connected: $S_{cp} = \frac{A_{\text{crowd}}}{A_{\text{largest component}}}$

(b) contains no significant holes: $S_{cl} = \frac{A_{\text{filled}}}{A_{\text{original}}}$

(c) is compact: $S_{sp} = \frac{P_{\text{largest component}}}{2\sqrt{\pi \times A_{\text{largest component}}}}$



(a) $SQS = 1.85$ (b) $SQS = 2.58$ (c) $SQS = 4.43$

Figure: Segmented images and the SQS associated with the quality of the segmentation. The smaller the SQS value, the better the quality

3D Reconstruction

1. Estimate the **focal length** from the image using vanishing point detection [6]
2. Estimate **depth** using MegaDepth [7]

Figure: The original image and its (inverse) depth map predicted by MegaDepth. The darker the pixel color, the larger the estimated distance from the camera.

3. Convert the relative depth map to a **3D point cloud** using the pinhole camera model

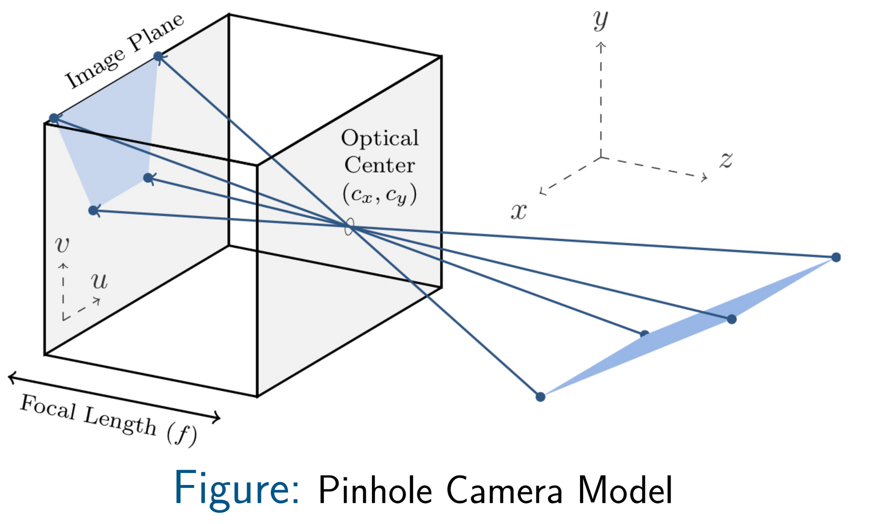
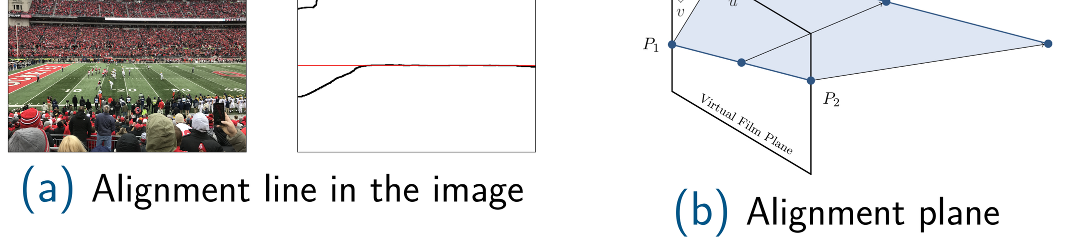


Figure: Pinhole Camera Model

4. Fit a plane using RANSAC to the crowd region in the 3D point cloud

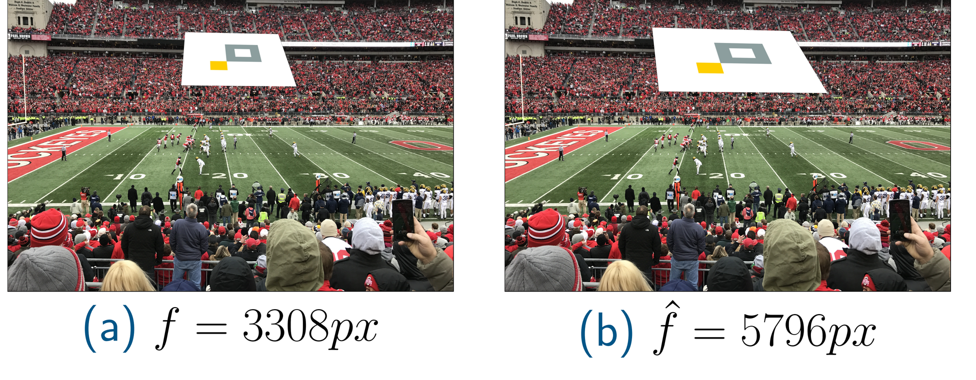
Asset Placement

1. Identify an **alignment line** in the 3D scene:
   (i) Apply Canny edge detection to the segmented image
   (ii) Find the longest line using Hough line detection
   (iii) Use the estimated focal length to project the alignment line to a plane in the 3D reconstruction



(a) Alignment line in the image (b) Alignment plane

2. Position the asset on the crowd plane in 3D, parallel to the intersection of the crowd and alignment planes
3. Limit the size of the asset so that it fits within a **convex hull** on the inliers of the crowd plan in 3D
4. Project the asset back to 2D using a **homography transformation**



(a) $f = 3308px$ (b) $\hat{f} = 5796px$

Figure: Experimentally, we found the perspective consistency of the augmentation was most sensitive to the focal length estimation step

Tracking across video

1. **Identify features** in the crowd plane (from 3D reconstruction) using SIFT, KAZE and SURF

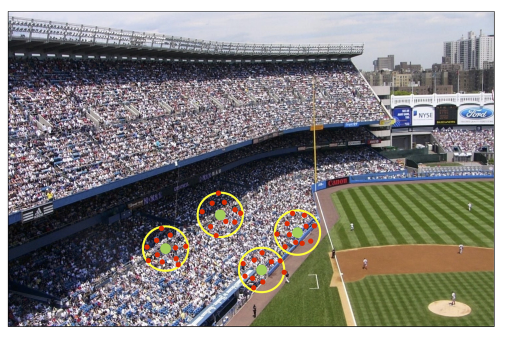


Figure: The red points are features to track, blue points are corners and the yellow circles are centered at each corner with a specified radius

2. Use **optical flow** to update the positions of the tracked features as the video advances [8]
3. Use the updated features' locations to estimate the homography matrix, to place the asset in the new frame

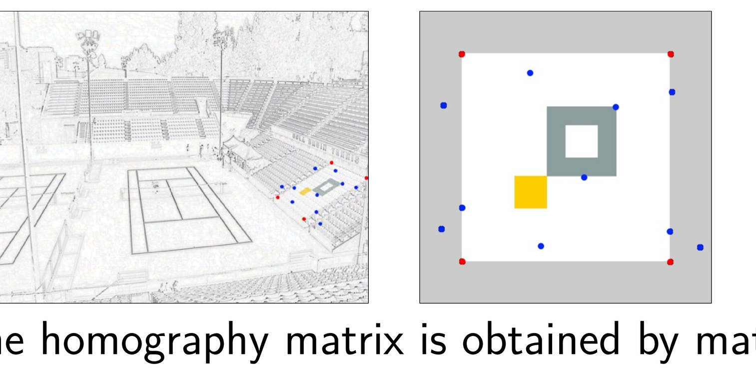


Figure: The homography matrix is obtained by matching features between the destination image and the asset

4. If a **shot change** is detected [9], freeze the features' locations and check each new frame for matching features as the video advances

Results

Check out an example of the pipeline's video output using this QR code:

Table: Approximate timing per step in pipeline for a single input image size of 1920 x 1080 px.

Time(s)	Process (coding platform/language)
11.491	PSPNet on GPU (Python, Tensorflow)
0.374	SQS on 25 segmented frames (Python)
12.159	Run MegaDepth (Python, Torch)
0.922	Mask image to "crowd" region (Python)
1.117	Identify alignment vector (Python)
6.627	Estimate focal length (Octave)
8.664	Fit plane using RANSAC (C++)
0.008	Display inliers (C++)
0.090	Warp asset using homography (C++)

References

[1] C. H. Chang *et al.*, "Virtual spotlighted advertising for tennis videos," *J. Visual Communication and Image Representation*, vol. 21, pp. 595–612, 2010.

[2] C.-H. Chang *et al.*, "Visa: Virtual spotlighted advertising," in *ACM MM*, pp. 837–840, 2008.

[3] C. Xu *et al.*, "Implanting virtual advertisement into broadcast soccer video," in *PCM*, pp. 264–271, 2004.

[4] Y. Li *et al.*, "Real time advertisement insertion in baseball video based on advertisement effect," in *ACM MM*, pp. 343–346, 2005.

[5] H. Zhao *et al.*, "Pyramid scene parsing network," in *CVPR*, 2017.

[6] B. Li *et al.*, "Simultaneous vanishing point detection and camera calibration from single images," in *ISCV*, 2010.

[7] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *CVPR*, 2018.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981.

[9] Y. Yildrim, "Shotdetection." `https://github.com/yasinyildirim/ShotDetection`, 2015.

Acknowledgements

- Professor Shantanu Joshi, UCLA
- The Institute for Pure and Applied Mathematics (IPAM) at UCLA
- GumGum Inc.
- US National Science Foundation Grant DMS-0931852