TT3D: Table Tennis 3D Reconstruction

Supplementary Material

9. Table Segmentation

Off-the-shelf segmentation models such as SAM [24] and monocular depth estimation models like Depth Anything V2 [58] struggle to accurately identify the edges of the table, as illustrated in Figure 7. One major challenge with depth-based models is their difficulty in distinguishing between the top surface and the side of the table, leading to ambiguous table edge detection. Meanwhile, SAM tends to over-segment the table, producing multiple disjointed submasks instead of a single cohesive table mask. This fragmentation requires additional post-processing to merge the individual segments, adding complexity to the pipeline.

Given these limitations, we opted to fine-tune existing segmentation networks on our custom dataset using the DICE loss. The Segmentation Models (PyTorch) library [20] provided a flexible framework for experimenting with different architectures and loss functions efficiently. We evaluated a range of segmentation architectures, including DeepLabV3 [11], DeepLabV3+ [12], U-Net [42], U-Net++ [65], PSPNet [61], MANet [29], LinkNet [8], FPN [30], and PAN [28]. Additionally, we tested various backbone networks to assess their impact on segmentation accuracy. The performance of these models is summarized in Figure 8.

10. Extension of our method to tennis

While table tennis and tennis differ in scale and equipment, their underlying ball dynamics share significant similarities. Both sports involve a ball that is required to bounce once on the playing surface, which contrasts with sports like badminton, where the shuttlecock does not bounce. The core principles of ball motion and bounce dynamics, therefore, remain relatively consistent between the two sports.

For tennis, existing camera calibration methods are well-established [16, 17]. These methods typically rely on standard computer vision techniques, such as color filtering and the Hough line transform, to accurately detect court lines. The intersections of these lines are then used to calibrate the camera. Compared to table tennis, this calibration process is more straightforward due to reduced player occlusion and a larger number of visible features on the court.

Adapting our 3D reconstruction method from table tennis to tennis primarily involves adjusting the physical parameters that govern the ball's motion. The tennis environment introduces additional complexity due to the variability in court surfaces—namely, grass, clay, and hard courts (e.g., cement). Each surface type influences the ball's behavior through its unique coefficient of restitution (COR) and friction coefficient (μ). For instance, grass courts have a lower COR and friction coefficient, resulting in a lower and faster bounce compared to clay courts, where higher friction increases ball spin and reduces speed post-bounce. Hard courts typically provide a balance between these extremes. We provide these tennis constants in Table 2.

Variable	Tennis
m [kg]	5.7×10^{-2}
r [m]	0.033
μ	0.55 / 0.9 *
k_{COR}	0.68 / 0.85*

Table 2. Physical constants used for tennis. * We distinguish between grass and clay courts, which have different dynamics.

As for the aerodynamics, the fuzzy surface of the tennis ball makes the dynamics much more complex. There are multiple formulation for the drag and mangus forces [41, 52]. We chose to use the convention from [41]. The drag force is then defined as:

$$\boldsymbol{F_D} = \frac{1}{2} C_D \rho \pi r^2 \|\boldsymbol{v}\| \boldsymbol{v} \tag{14}$$

where:

$$\begin{aligned} C_{\mathrm{D}} =& 0.6204 - 9.76 \times 10^{-4}(\boldsymbol{v} - 50) \\ &+ \left[1.027 \times 10^{-4} - 2.24 \times 10^{-6}(\boldsymbol{v} - 50)\right] \boldsymbol{\omega}, \end{aligned} \tag{15}$$

And the Magnus force is defined as:

$$\boldsymbol{F_M} = \frac{1}{2} C_M \rho \pi r^2 \|\boldsymbol{v}\| \frac{\boldsymbol{\omega} \times \boldsymbol{v}}{\|\boldsymbol{\omega}\|} \tag{16}$$

where:

$$\begin{aligned} C_M =& ||\boldsymbol{\omega}|| \left[4.68 \times 10^{-4} - 2.10 \right. \\ & \left. \times 10^{-5}(||\boldsymbol{v}|| - 50)\right. \end{aligned} \tag{17}$$

With the updated ODE, the same 3D ball trajectory method can be used.

Figure 7. Comparing different segmentation methods for table tennis table. From left to right: Input image, Depth Anything V2 [58], SAM [24], Custom trained Unet++ (ours)

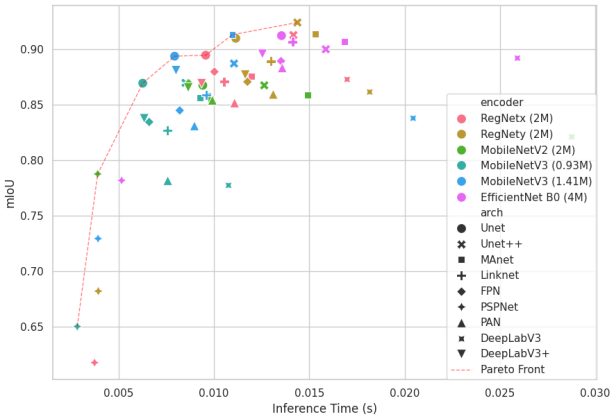
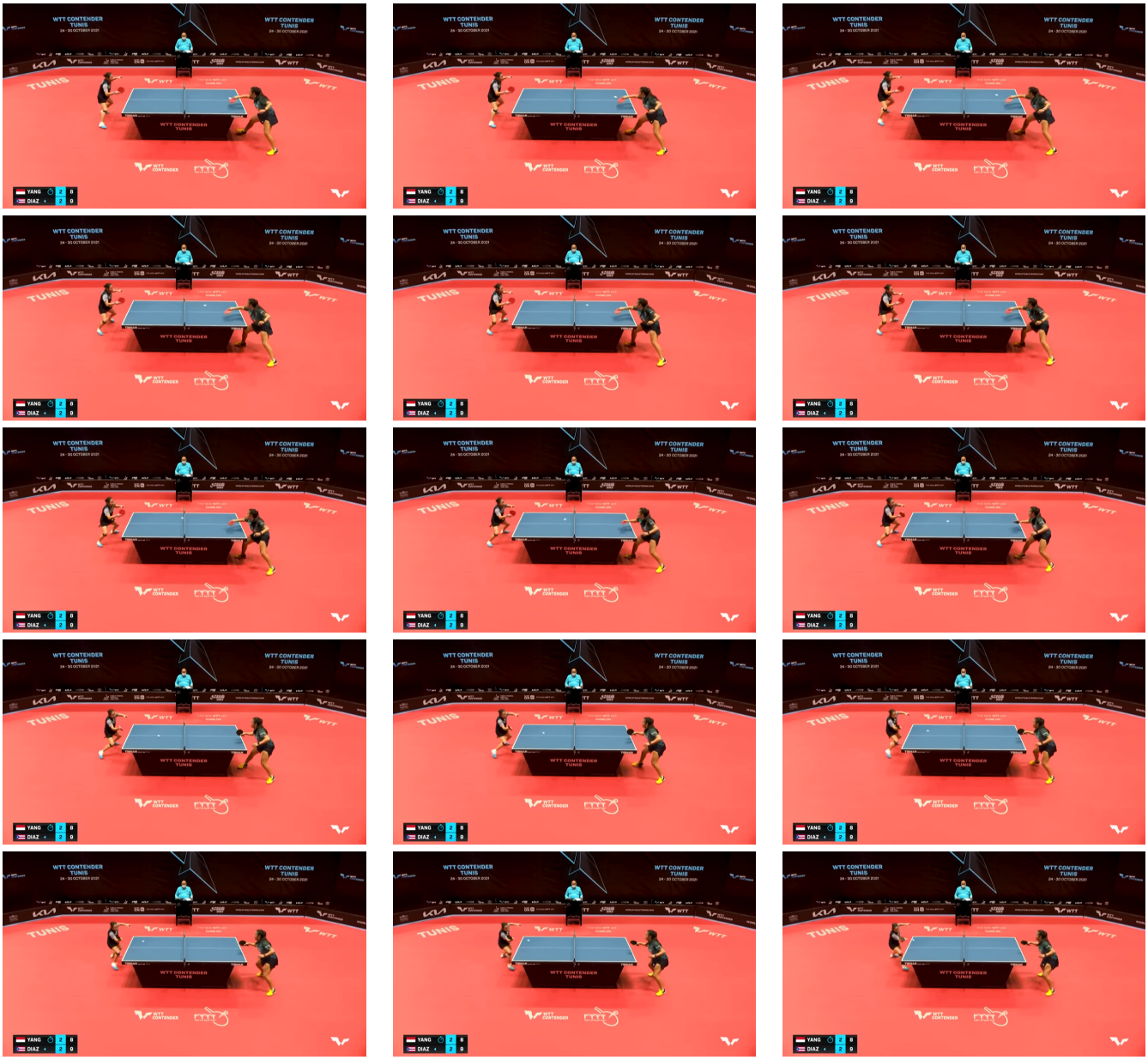
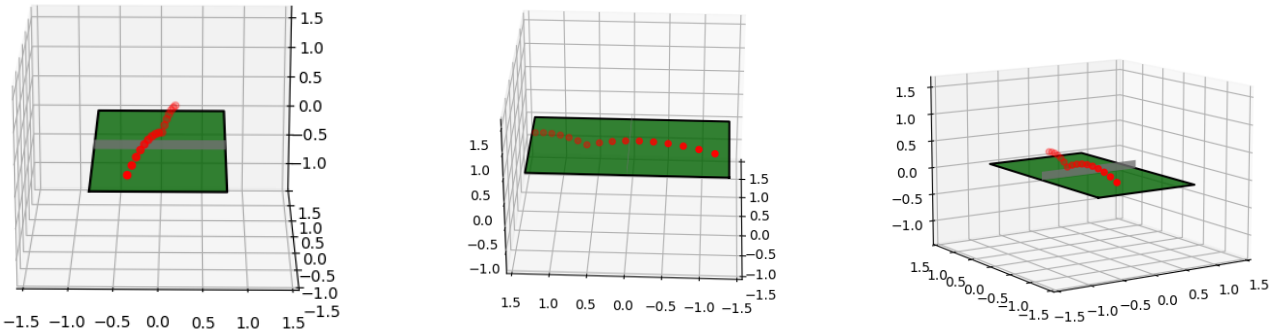


Figure 8. Benchmark of the table segmentation models for different model architecture and encoders. The Pareto from is draw to show the optimal models with regards to mIoU and inference time.





Backview Sideview Oblic view

Figure 9. (Top)Input frames used to reconstruct the 3D ball trajectory. They are sorted in chronological order. (Bottom) 3D reconstruction results obtained from these frames for different points of view.

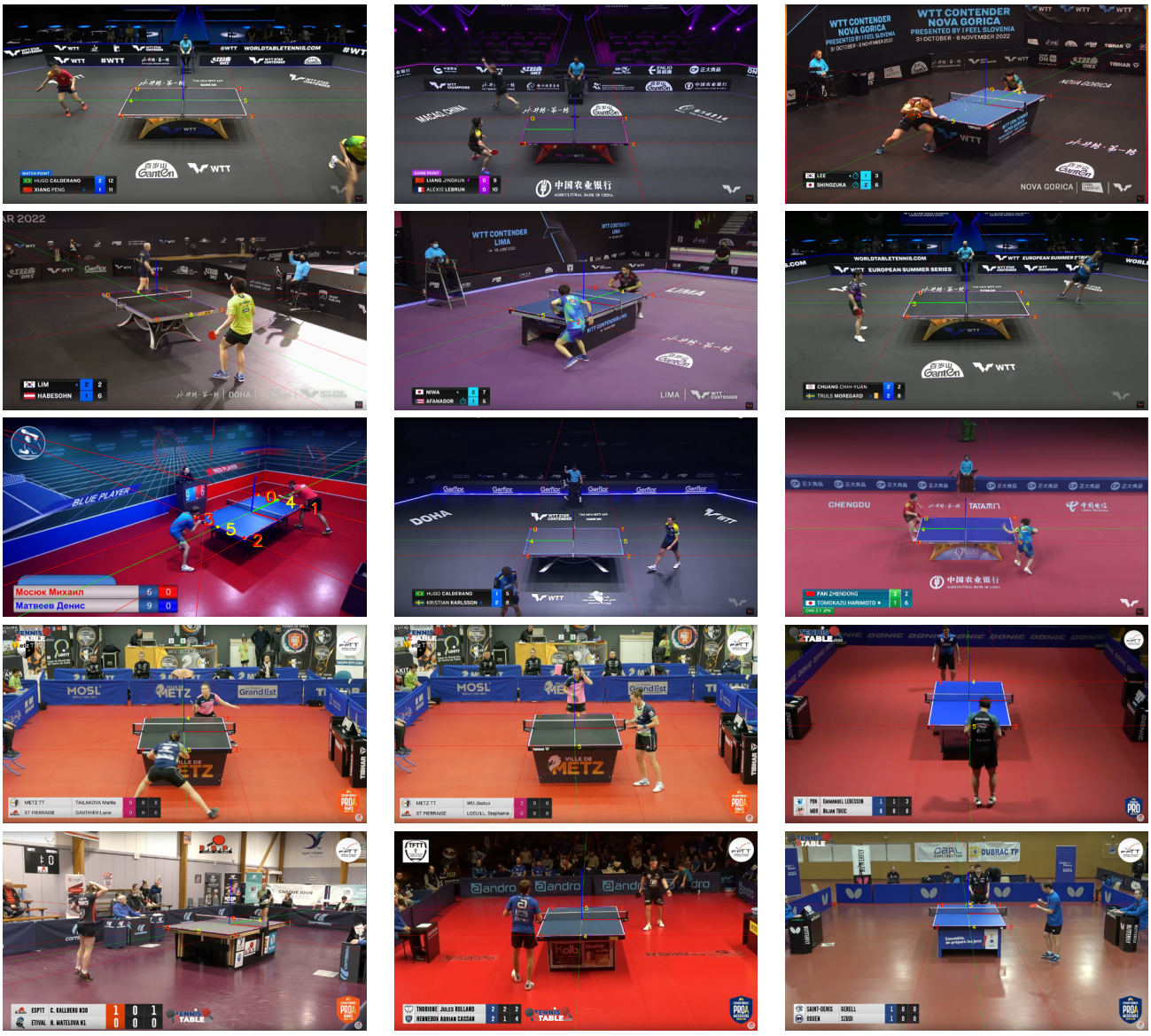


Figure 10. Camera calibration in different contexts. The red lines are the potential detected table edges. The green lines are the potential midlines. The numbers represent the feature number.

Figure 11. Camera calibration failures mostly because of obstruction from the table tennis player. The red lines are the potential detected table edges. The numbers represent the feature number. Most misdetections are due to the player's body being detected as the table edge.