Vid2Player: Controllable Video Sprites that Behave and Appear like Professional Tennis Players (Supplemental Material)

ACM Reference Format:

1 BALL CONTACT POSITION ESTIMATION

Ball contact position is crucial for generating realistic ball trajectories and plausible player motion at contact time. We estimate the ball’s 3D court space position at the time of contact using a two-step process:

(1) We estimate the 2D screen space position of the ball in the frame of contact
(2) We use the player’s pose at the frame of contact to convert the 2D screen estimate of the ball into a 3D court space estimate.

2D (Screen Space) Ball Contact Point Estimation. We train a k-nearest neighbors regressor (k-NN) to automatically estimate the ball contact position in screen space from existing pose information. Specifically, we extract the following features from our database to train the k-NN:

1. Keypoint Features
   - 2D keypoints for ball contact and preceding frame, relative to the player bounding box: RShoulder, RElbow, RWrist, LShoulder, LElbow, LWrist, RHip, RKnee, RAnkle, LHip, LKnee, LAnkle.
   - Joint lengths (\(D_{ij}\)) denotes the pixel distance between two joints:
     \[ D_{ij} = |x_i - x_j| \]
   - Feature maps (\(\mathbb{R}^{4096}\)) generated by the second to last fully-connected layer of pre-trained VGG-19 [Simonyan and Zisserman 2014]. We input the cropped frame centered with player bounding box and then use principal component analysis (PCA) [Jolliffe 1986] to reduce dimension into \(\mathbb{R}^{50}\).

2. Visual Features
   - Shot type: 1 for forehand, -1 for backhand

3. Labeled Features
   - Shot type: 1 for forehand, -1 for backhand

The features are stacked in the order listed above to form a single feature vector (\(\mathbb{R}^{116}\)). We hand labeled contact positions (pixel space) to provide ground truth supervision for training three separate models for the three male players (for simplicity, we use Federer’s model for Serena Williams). We use sci-kit learn’s implementation of a k-NN regressor (KNeighborsRegressor) [Pedregosa et al. 2011]. For all players, we choose \(k=2\). Fig. 1 shows examples of our estimated ball contact position on examples from the validation dataset. Table 1 gives both dataset details and provides quantitative measure of contact point estimation accuracy on the validation set.

3D (Court Space) Ball Contact Point Estimation. We make the assumption that contact always happens at the same depth plane as the player’s court position. We can then calculate a 3D offset for the contact position \(x_c^t(t_c^t)\) relative to the player’s court position \(x_p(t_c^t)\) using the 2D offset between the estimated contact \(x_c^t\) and the player’s root position \(x_p^t(t_c^t)\), scaled by the player’s real-life height

<table>
<thead>
<tr>
<th>Player</th>
<th>(N_i)</th>
<th>(N_v)</th>
<th>(R^2) score</th>
<th>L2 error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federer</td>
<td>300</td>
<td>152</td>
<td>0.890</td>
<td>29.4</td>
</tr>
<tr>
<td>Nadal</td>
<td>300</td>
<td>141</td>
<td>0.896</td>
<td>27.7</td>
</tr>
<tr>
<td>Djokovic</td>
<td>300</td>
<td>144</td>
<td>0.910</td>
<td>24.9</td>
</tr>
</tbody>
</table>

Table 1. Quantitative evaluation of ball contact position estimation. \(N_i\) and \(N_v\) denote the number of shots used for training and validation. \(R^2\) denotes the coefficient of determination metric and L2 error measures the average error (in pixels) between the predicted and ground truth contact position.
2 BALL TRAJECTORY GENERATION

Our system relies on estimating the 3D trajectory of the tennis ball in two situations. During offline database preprocessing, the ball trajectory model is used to estimate the trajectory of the ball in all database clips given the time and 3D court space location of two consecutive shots. During novel point synthesis, the ball trajectory model is used to search for a trajectory that matches the specified ball start position and a specified time and location where the ball should bounce on the opponent’s side of the court.

As introduced in the main paper, we update the ball velocity using the following equations:

\[
\begin{align*}
\frac{dv_x}{dt} &= -kv(C_d v_x + C_L v_y) \\
\frac{dv_y}{dt} &= kv(C_L v_x - C_d v_y) - g \\
C_L &= 1/(2 + v/\nu_{spin})
\end{align*}
\]

In our implementation, we use \( m = 0.057 \, \text{kg}, \ R = 0.032 \, \text{m}, \ \rho = 1.21 \, \text{kg/m}^3, \ g = 9.81 \, \text{m/s}^2 \) and \( C_d = 0.55 \). We simulate the ball’s bounce as a collision where the coefficient of restitution (COR) is constant and the spin after bounce is always topspin. In our implementation, we set COR to be 0.65. If the shot before bounce is topspin, we set \( \nu_{spin} \) after bounce to be \( \text{COR} \cdot \nu_{spin} \) before bounce, otherwise we set \( \nu_{spin} \) after bounce to be \( 5 \, \text{m/s} \).

During the data annotation phase, given a pair of consecutive ball contact positions, and the time in between, we perform a grid search over the ball’s launch velocity (horizontal and vertical components of linear velocity, as well as spin velocity) to yield a trajectory which starts at the first contact point, clears the net, bounces in the court, and then closely matches the time and location of the second contact point. We search over the three parameters below to find the optimal trajectory. Fig. 2 shows aggregated results of this estimation.

- Launch velocity (magnitude of linear velocity): [50, 70] \, \text{m/s} for serve and [15, 35] \, \text{m/s} for non-serve shots.
- Launch angle (between linear velocity and ground plane): [−15, 0] \, \text{degrees for serves and [0, 15]} \, \text{degrees for non-serve shots.}
- Magnitude of spin velocity: [0, 20] \, \text{m/s} (we flip the sign before \( C_L \) if the ball undergoes topspin).

When synthesizing novel points, our system performs grid search over the same parameters to find the optimal trajectory given the ball start and target bounce positions, along with the target shot velocity.

3 POINT STATE DESCRIPTOR COMPUTATION

In our behavior model, we build a point state descriptor for a point’s current state by discretizing the features which influence player’s behavior decisions. The details of the discretization for each feature can be found in Table 2. We divide each side of the court into six regions. Horizontally, we split the court into three regions (Ad/Center/Deuce). The three regions are equally divided within the side boundaries for singles and the non-center regions are extended to their open areas. Vertically, we split the court into Front and Back given the middle line between service line and baseline. This simple discretization can capture player and ball characteristics such as cross-court vs. down-the-line shot and ground strokes vs. net plays. The origin of our court coordinate system lies at the court’s center, with a horizontal range spanning ±4.12 m and a vertical range spanning ±11.89 m. When we compute the descriptor for the player on the far side, we rotate the court by 180 degrees to make them consistent.

4 CLIP SEARCH COST

In the main paper, we have defined all the cost terms used for clip search. However, since these cost terms are measured in different units (e.g. \( C_{\text{pose}} \) in pixel distance and \( C_{\text{contact}} \) in meter), To make them roughly comparable, for each cost term we determine a threshold for visually unacceptable results and use the threshold to
Fig. 2. Estimated trajectories in database clips for all players separated by forehand/backhand shots. The red line indicates the mean of trajectories' maximum height before bounce. Nadal has the highest trajectories among the three male players.

Normalize the cost term following the equation below:

\[ c' = \begin{cases} 
  c/t & \text{if } c < t \\
  \text{inf} & \text{otherwise}
\end{cases} \]

\( c \) and \( c' \) denote the cost before and after normalization, \( t \) denotes the threshold. Once all the cost terms are normalized, we linearly combine them using weights that balance trade-offs between different constraints. We set the highest weight for \( C_{\text{recover,dir}} \) since the player recovering to the wrong direction leads to egregious behavior artifact. We also prioritize the weights for \( C_{\text{react,velo}} \) and \( C_{\text{react,dir}} \) to minimize the foot skating during the reaction phase. \( C_{\text{contact}} \) and \( C_{\text{shot,place}} \) are less weighted as it is harder to notice the match error for shot contact position and shot placement position due to the fast swing motion of the player. The final weights are determined experimentally to produce good results. We list the thresholds and weights in Table 3.

### Table 3. Threshold and weights used for each cost term.

<table>
<thead>
<tr>
<th>Thresh</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 pixels</td>
<td>0.05</td>
</tr>
<tr>
<td>0.2 m/s</td>
<td>0.5</td>
</tr>
<tr>
<td>1 m</td>
<td>0.05</td>
</tr>
<tr>
<td>1.2</td>
<td>0.5</td>
</tr>
<tr>
<td>0.5</td>
<td>1.2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25 mph</td>
<td>0.05</td>
</tr>
<tr>
<td>1 m</td>
<td>0.05</td>
</tr>
</tbody>
</table>

5 EXTENDED EVALUATION

5.1 Recreation of player style

Fig. 3 compares the shot placement and recovery position of all three male players in the in the real world data from our database with the results of simulated points generated using our behavior models. Note that in general, our simulated points share similar statistics in important player behavioral decisions, including shot selection (shot type, shot velocity and shot placement) as well as recovery positions, which are explicitly modeled by our behavior models.

5.2 Database Clip Usage

We also evaluate how frequently database clips are used during our novel point synthesis. For this evaluation, we randomly generate 1500 points between the three male players. Fig. 4 plots a histogram of the number of times each clip was used in these simulations. 89.1% of the database clips are used at least once. The most frequently selected clips are cross-court ground stroke rallies, which is the most common pattern in tennis. Never chosen clips are in general depicting atypical behaviors, especially net play, for which corresponding game situations are less likely to occur in our synthesis.

6 RENDERING

6.1 Player Shadows

We generate player shadows using a purely image-based approach that warps the player’s alpha matte to approximate projection on the ground plane from a user-specified lighting direction (as shown in Fig. 5). Specifically, we choose three points to control the image warping: two ankles points and one head point (Fig. 5a). To obtain
Federer Djokovic Nadal
(a) Shot placement (forehand) (b) Shot placement (backhand) (c) Recovery position

Avg vel 75 kph Avg vel 82 kph Avg vel 79 kph

Federer Avg vel 81 kph
Djokovic Avg vel 83 kph
Nadal Avg vel 81 kph

Mean (-1.0 m, -1.0 m) Mean (-0.9 m, -1.8 m) Mean (0.5 m, -1.9 m)

Fig. 3. Shot placement (forehand/backhand) and recovery positions of Federer (right-handed), Djokovic (right-handed), and Nadal (left-handed) in our simulated points (top) and database points (bottom). By modeling behaviors from real-world data, we can simulate points that capture player-specific styles that are consistent with well-known tendencies of the real-life players.

Fig. 4. The statistics of how frequently database clips are used during the synthesis of 1500 randomly generated points. Almost 90% of the database clips are used at least once.

Fig. 5. Image-based approach for player shadow rendering. We warp the player’s alpha matte using three control points (red dots) from (a) to (c) to approximate projection on the ground plane from a user-specified lighting direction. We warp the matte ten times around the give lighting direction to simulate soft shadow (d) and darken the pixels in the warped matte to render the final player shadow.

the positions of these control points in the target image, we fix the two ankle points and rotate the head point around the middle of the two ankle points given a user-specified angle, then use moving least squares [Schaefer et al. 2006] to warp the player’s alpha matte to compute the shadow region (Fig. 5c). We simulate the effect of soft shadows by warping the matte ten times around the given angle (Fig. 5d). Finally, the renderer darkens pixels of the background image that lie within the warped matte to render the player shadow (Fig. 5b). This simple approach yields plausible shadows for the lighting conditions in our chosen background images, and avoids the challenge of robust 3D player geometry estimation to perform physically correct shadow rendering or leveraging learning based method to train a generative model for shadow synthesis.

6.2 Appearance Normalization and Body Completion

Training details. For appearance normalization, we directly use the Cycle-GAN implementation from [Zhu et al. 2017]. We trained 12 models for both player relighting and re-clothing tasks from scratch, including: 10 relighting models for the five broadcast videos in our database (each model for near/far side of the court) and two re-clothing models for Serena Williams to modify her clothing from one video to look like her clothing from the other (plus each model for near/far side of the court). To train each model, we collect 10K images from each visual domain. Training each model takes approximately 24 hours to converge using the default training parameters from [Zhu et al. 2017] on a TITAN V GPU.

For body completion, we use the Pix2PixHD implementation from [Wang et al. 2018] to convert a stick figure rendering of a players skeleton to a realistic sprite. We trained four models from scratch, for each of the players standing on the near side of the court. To train each model, we collect 5K image pairs. The training takes approximately 12 hours to converge using the default parameters of the code on a TITAN V GPU.

Extended results. Fig. 6 illustrates the overall performance of our appearance normalization for relighting the tennis players into uniform lighting condition. For each source video, we uniformly
sample the nearside players throughout different times of the match and make a montage of the cropped frames around the player. We visualize the montages before and after normalization in the top and bottom rows. Despite the dramatic lighting variance throughout the first two matches and the overall brightness/saturation difference for the other three matches, our appearance normalization is capable of bringing these diverse lighting distributions back to the same lighting condition.

Fig. 7 shows randomly sampled results of relighting the three male players, as well as both relighting and re-clothing Serena Williams.

Fig. 8 shows the randomly sampled results of our body completion approach for Nadal and Djokovic, who have more cropped frames as they usually stand further back compared to the other players. Our approach successfully complete most of the cropped bodies.

REFERENCES


Fig. 7. Appearance normalization results for relighting the male players (top) and relighting and re-clothing Serena Williams (bottom). The normalized sprite is shown in the black box and the original sprite is shown to its left. Notice how normalization corrects for significant sunlight differences that occur over men’s matches that take place outdoors over the span of hours, and adds long sleeves to William’s shirt. (high-resolution image—please zoom in)
Fig. 8. Body completion results for Nadal (top) and Djokovic (bottom). The completed sprite is shown in the black background and the original cropped sprite is shown to its left. (high-resolution image—please zoom in)