Tennis Stroke Detection

Khushdeep Singh (10351)

Aakash Verma (10002)

Advisor: Dr. Amitabha Mukherjee

Dept. of Computer Science and Engineering

{khushd, aakashv, amit} @ iitk.ac.in

April 12, 2012

Abstract

Human activity recognition has been studied in many research domains, and sports video analysis is one of them. This project proposes a method of stroke detection in tennis videos basically using particle filtering techniques. Stroke Detection is based on a combination of particle filters, motion descriptors and event detectors. Firstly, we track the player and extract player centered images. Second, we use the iterative application of the famous Lucas Kanade algorithm for analyzing the optical flow. And finally, we generate motion descriptors. The task of feature detection has been done using an extension of the Voila Jones algorithm in 3-D. Adaptive Boosting algorithm of machine learning has been used for the purpose of training.

1 MOTIVATION

Video Analysis now a days has become a very important field because of its application in day to day life. Its applications include fields such as security surveillance, human machine interface, sports, traffic monitoring, etc. Our project emphasizes on one of the many aspects of human activity recognition in videos; that is sports. Activity recognition in sports videos has a variety of uses like automatic classification of huge sports content, improving the performance of players, generating automatic commentary etc. Also, since sports activities like tennis strokes are defined explicitly, they are widely used for experiments on human activity recognition.

2 INTRODUCTION

The basic task of stroke detection in a tennis video can be divided into various subtasks which are as follows:

1. Player Tracking using particle filters, based on the methods proposed in [1].

2. Optical Flow Analysis: We have used **Lucas Kanade** Algorithm of optical flow analysis in this project.
3. Motion Descriptors: The motion descriptors are generated from the optical flow of the player centered images. We then also use the Gaussian filter to blur each motion descriptor.
4. Feature Detection: We have used a 3-D extension of the famous **Voila Jones** algorithm for object detection in 2-D as mentioned in [2].
5. AdaBoost: We have used adaptive boosting algorithm of machine learning for the purpose of training.

3 PLAYER TRACKING USING PARTICLE FILTER

In the project we have formulated tracking as inference in a Hidden Markov Model which contains two states; the observation state (from the image data) and the hidden data (e.g. object location, scale, etc.) as mentioned in [3]. What we are doing is that we are using the concept used in [1] for approximating the probability distribution by a weighted sample set $S = \{(\mathbf{s}^{(n)},\pi^{(n)}) | n=1.....N\}$. Each sample **s** represents one hypothetical state of the object, with a corresponding discrete sampling probability π, where $\Sigma^{N}_{n=1}\ \pi^{(n)} = 1$. The mean state of an object is estimated at each time step by $E[S] = \Sigma^{N}_{n=1}\ \pi^{(n)}s^{(n)}$.

- Prior distribution: $p(x_0)$
  - Describes initial distribution of object states.
  - We are doing it manually.

- Transition Model: $p(x_t | x_{t-1})$
  - Specifies how objects move between frames.
  - We use second order auto regressive model:

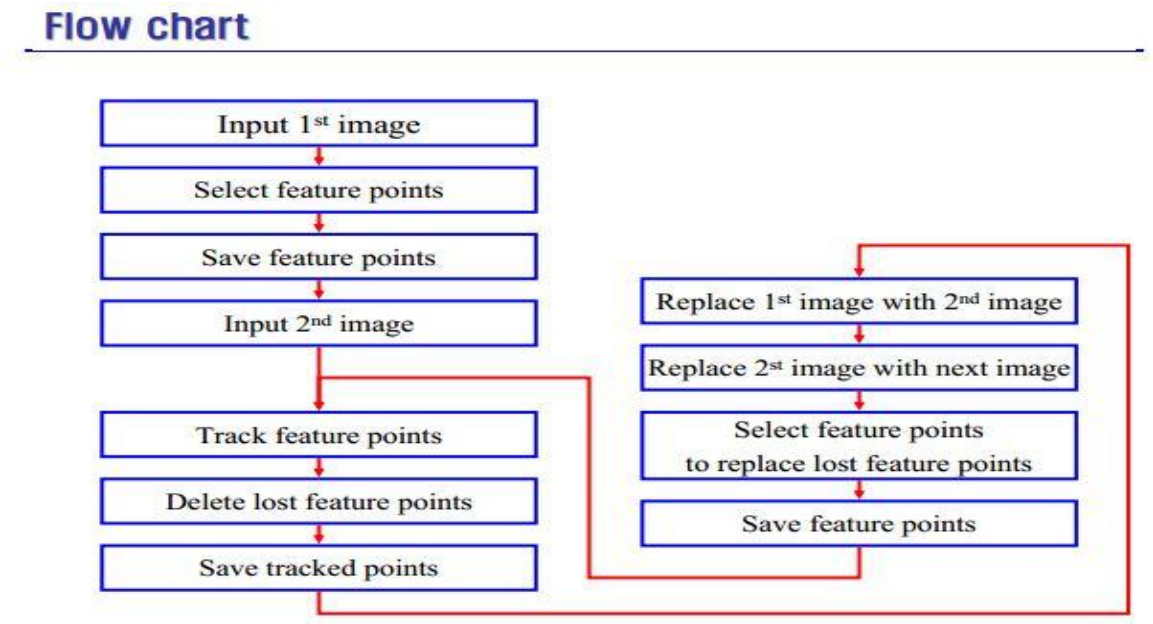
$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{x}_{t-2} + \mathbf{w}_t$$

- Observation Model: $p(y_t | x_t)$
  - Specifies the likelihood of an object being in a specific state (i.e. at a specific location)
  - We are using a color histogram- based model.

The distance between two distributions is calculated using the **Bhattacharyya distance** which is defined as $d = (1 - \rho[p,q])^{1/2}$ where $\rho[p,q] = \Sigma^{n}_{u=1} (p^{(u)}q^{(u)})^{1/2}$

4 OPTICAL FLOW ANALYSIS

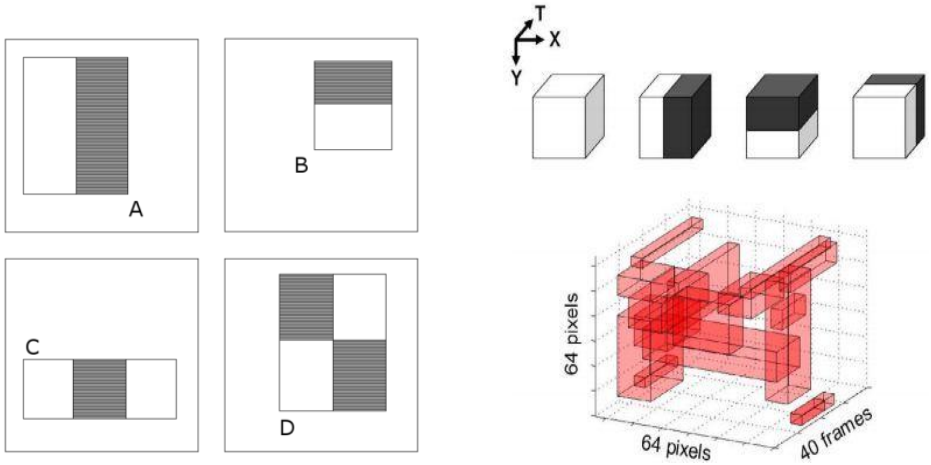
Optical flow estimation is of two types; namely, sparse and dense optical flow analysis. Sparse optical flow estimation involves computation over a subset of image points, that is, only some selected feature points. On the other hand dense optical flow involves estimation of optical flow over each and every image pixel. Sparse analysis results in quicker results but they are less accurate compared to dense analysis. There are various optical flow analysis algorithms which include Lucas Kanade algorithm [4], Farneback algorithm [5], etc. In this project we have used a pyramidal implementation of the classical Lucas Kanade algorithm as used in [4]. What the algorithm does is that it takes the video as a set of sequential images by taking out frames from the video and then selects feature points in the image which are then tracked in the next image of the sequence. An iterative implementation of the Lucas Kanade optical flow computation provides sufficient local tracking accuracy.



5 MOTION DESCRIPTOR GENERATION

The motion descriptors are generated from the optical flow of the player-centered images. Horizontal and vertical components of the optical flow field, F_x and F_y, are separated into the positive components F_x^+, F_y^+ and the negative components F_x^-, F_y^-. Note that if one of the positive components is nonzero, the corresponding negative component will be set to zero and vice versa. Each of the components is then blurred using a 2-D spatial Gaussian filter to construct the motion descriptors Fb_x^+, Fb_x^-, Fb_y^+ and Fb_y^-. Once the motion descriptors are generated, each motion descriptor is cropped and rescaled to a fixed size according to the size of the tracker.

6 FEATURE DETECTION USING VOILA JONES

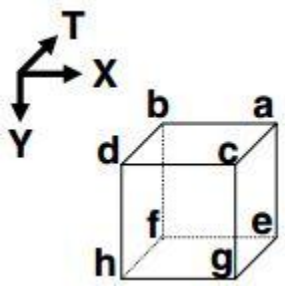


***Efficient Visual Event Detection using Volumetric Features* by Yan Ke, Rahul Suthankar, Martial Herbert [ICCV'05]**

In this project we extend the concept of 2-D features [2] to 3-D by introducing a temporal feature as well as mentioned in [6]. As shown in the first row of the second figure, our volumetric features consist of one-box or two-box volumes. The value of the one-box feature is simply the sum of the pixels within the volume. Correspondingly, the value of a two box feature is the difference of their individual sums. We use an "integral video" data

structure [6] to efficiently calculate the features. An integral video at pixel location (x, y) and time t is defined as the sum of all pixels at location less than or equal to (x, y, t).

More formal definition of integral video is given below.



The volume of this box can be computed from the integral video with eight array references: e – a – f – g + b + c + h – d.

The integral video *iv*, is defined as

$$iv(x,y,t) = \sum_{x' \le x} \sum_{y' \le y} \sum_{t' \le t} i(x',y',t'),$$

Where *i(x, y, t)* is the pixel value at the original image. In our framework, we compute two integral volumes for two optical flow components, and use $v_x(x, y, t)$ and $v_y(x, y, t)$ in place of *i(x, y, t)*. *iv* can be easily computed using the following recurrences:

$$\begin{aligned} s_1(x,y,t) &= s_1(x,y-1,t) + i(x,y,t) \\ s_2(x,y,t) &= s_2(x-1,y,t) + s_1(x,y,t) \\ iv(x,y,t) &= iv(x,y,t-1) + s_2(x,y,t), \end{aligned}$$

Where $s_1(x, -1, t) = s_2(-1, y, t) = iv(x, y, -1) = 0$. Therefore, computing the integral video structure is only marginally more expensive than computing a series of integral images in a video sequence.

7 ADAPTIVE BOOSTING

AdaBoost is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire. It is used in conjunction with many other learning algorithms to improve their performance. What adaboost does is that it constructs a "strong" classifier as a linear combination of many "weak" classifiers.

AdaBoost generates and calls a new weak classifier in each of a series of rounds t = 1,…,T. For each call, a distribution of weights D_t is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far been misclassified.

ADABOOST PSEUDOCODE

Initialize distribution $D_t(i) = \frac{1}{m}$ for all i (same weight for every example) For t = 1…T

1. Call *WeakLearn* using distribution D_t.
2. Get back a classifier $h_t : X \rightarrow Y$
3. Calculate error of h_t, $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$. If $\epsilon_t > \frac{1}{2}$, then set T=t-1 and recommence loop
4. Set α_t. (For example $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$).
5. Update D_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \\ e^{-\alpha_t y_i h_t(x_i)} & \text{, general formula} \end{cases}$$

Lecture on AdaBoost by Jan Sochman and Jiri Matas, Center for Machine Perception, CTU, Prague

8 IMPLEMENTATION AND RESULTS

The entire code was implemented in C++ using libraries OpenCV and OpenCVX.

For player tracker based on particle filter, the code we used was available at [http://code.google.com/p/opencvx/wiki/ParticleFilter] [Copyright (c) 2008, Naotoshi Seo <sonots@sonots.com>]

The rest of the code has been written by us. We implemented the detector on two datasets:

KTH:

The motivation behind using this was the person centred videos which our particle tracker effectively generates. We used three actions: boxing for positive examples and hand wave and hand clapping for negative examples. The window used for finding features was 64X48.

Results:

No. of Training Examples	No. of Features	Misses	False Alarms
203	2,00,000	3%	60%
191	2,00,000	44%	21%
203	2,00,000	20.69%	40.6%

**first 3 results use 1000 week classifiers and the fourth result used 50000 week classifiers*

Tennis Dataset:

We created a dataset of 70 serves, 50 forehands and 50 backhands, by downloading videos from YouTube. The player was tracked using the particle tracker. An area which was double the area of that tracked region was cropped to generate figure centred images which were resized and optical flow was found for these to generate motion descriptors.

The window used for finding the features has been resized to 32X32. For the positive training examples, the time at which the ball strikes the tennis racket was manually found and a window of 20 frames centred at this time *t* was taken. We have used serves as positive training examples and forehands as negative.

Results:

No. of Training Examples	No. of Features	Misses	False Alarms
105	2,00,000	40%	22%
105	2,00,000	60%	22%

Golf Dataset:

A few initial frames which do not contain the golf swing were used as negative examples and a 20 frame window centred at the hit instant was used as positive examples. It was only tried on a limited dataset of 49 videos (from UCF Sports Action Datasets; http://www.cs.ucf.edu/~kreddy/Datasets.html). The results were not very satisfying as out of 9 positive examples, only 2 were detected while out of 44 negative examples 4 were false alarms.

In tennis the results are poor in comparison to those in the KTH due to smaller dataset and in golf we have even poorer results as the dataset was even smaller. Also, training was done only on 1000 weak classifiers, as training takes a lot of time (50,000 took a day and a half).

It is also very important to mention that this is only the preliminary step for event detection. Most of the event detecting techniques use further processing like particle filter used in [7] pg. 2, 3, and 4.

9 FURTHER WORK

This project provides a basic way of detecting and classifying strokes in a tennis video. On further improvements it can be very beneficial in several ways. It can also be used to improve the game of a player by creating a system that would count the percentage of errors the player made in different strokes. Strategic planning before a game can also be done by analyzing the game play of the opponent.

Outside the field of tennis also we can implement this method with slight changes. There have already been a lot of efforts to use the concept of human activity recognition for traffic monitoring and security surveillance.

Right now only court view videos are supported and analyzed by the system, but it would be a very good improvement if we can incorporate different camera angles. This would make the system very flexible and more useful as well since actual tennis videos are not from just one particular camera angle. But this task is beyond our scope right now as the actual videos have too many camera angles, they sometimes zoom in and out or focus on the umpire or the audience, incorporating all this in the project did not seem possible.

Right now what we are doing is just detecting the stroke, further we can use various particle filtering to classify the strokes as well. Advancements can be made by introducing methods which can also tell whether the stroke was hit correctly or not, this would help analyze the performance of the players. Also such a system can be used for automatic scoring in a match as well as for commentary.

Large improvements in the results can be made by using a greater number of training examples, and also increasing the number of week classifiers. Since, we are using the AdaBoost algorithm which is very expensive both in terms of time consumption and memory usage; we could not use sufficient number of classifiers as well as examples. Therefore for better results, we need to use some modification of this algorithm which is less expensive in both contexts.

References:

[1] *An Adaptive Color Based Particle Filter.* Katja Nummiaro, Esther Koller Meier, Luc Van Gool [2002].

[2] *P. Viola and M. Jones: Robust real-time object detection*. In Proc. of IEEE Workshop on Statistical and Computational Theories of Vision [2001], pg. 4-8.

[3] *Particle Filtering and Object Tracking* by Rob Hess [2006].

[4] Jean-Yves Bouguet: *Pyramidal Implementation of the Lucas Kanade Feature Tracker, Description of the algorithm* [2001], pg.2-4.

[5] *Klas Nordberg and Gunnar Farneback: A Framework for Estimation of Orientation and Velocity,* Computer Vision Laboratory, Department of Electrical Engineering, Linkoping University[2001],pg. 3-4.

[6] *Efficient Visual Event Detection using Volumetric Features* by Yan Ke, Rahul Suthankar, Martial Herbert [ICCV'05], pg. 2-4.

[7] *Tennis stroke detection and classification based on boosted activity detectors and particle filtering* by Takehito Ogata, William Christmas, Joseph Kitler and Seiji Ishikawa [2006], pg. 2-4.