# Articulated Human Pose Tracking Using Game Theory

Chenguang Liu et al.

## SC 631 Project - Prof. Ankur A Kulkarni

---


Nitish Tongia - 170010042
Mihir Kavishwar - 17D070004
Rishabh Dahale - 17D070008

IIT Bombay

# TABLE OF CONTENTS

# Introduction

- **Problem Statement**: This paper deals with the computer vision task of human pose estimation. Authors have developed a unique technique of pose estimation using Game Theory
- Human pose tracking have attracted interests in recent time due to its promising applications like virtual reality, intelligent surveillance, advanced user interfaces, motion analysis etc
- **State of the Art Model**: Current methods of Human pose estimation deploy a multi layer complex convolution neural networks which leads to high computational complexity and sometimes cannot be processed in real time due to large number of parameters
- The proposed method neither requires a learning process which needs large training set and confines tracker to limited poses, nor it needs to obtain foreground blobs of the subject by background subtraction

# Introduction

- The basic idea of the method is as follows: Each limb can be viewed as an articulated object of which the two parts are hinged at joint
- By viewing each part as playing normal form game, it's global optimal can be found by calculating the Nash Equilibrium

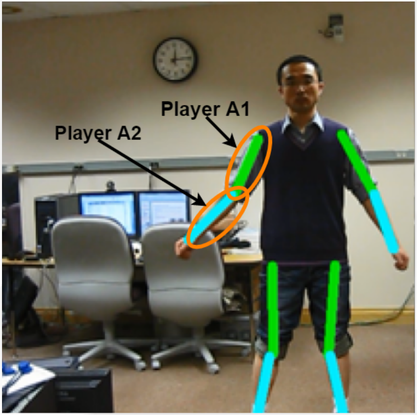


Figure: Players for tracking right arm

# Introduction

- Each agent has a number of strategies. By playing these strategies with the correlated agent, an agent assigns each strategy a payoff which is calculated through template matching and dependence measurement with the opponent strategy
- By computing the profit of each NE, four pairs of optimal strategies with maximum profit can be selected as the tracked human pose in the current frame
- The proposed method is decentralized as each agent only needs to optimize its own utility hence complicated joint motion estimation is avoided

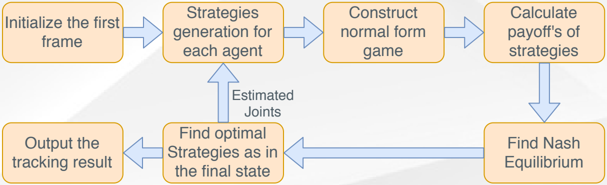


Figure: Flowchart of the Game Theory Based Human Pose tracking method

# Initializing the First Frame

Figure: Initializing the first frame

First frame needs to be annotated by human as shown above

# Strategy Generation

- The game is formulated as each player playing a game with it's adjacent player (Player A1 vs Player A2)
- The initialized strategies are generated based on the marked limbs in the initialization process, while the strategies in the subsequent frames are generated according to the limbs tracked in the previous frame.
- To generate strategies for each player following steps were followed: *M* random candidate joints were generated around each joint tracked in the previous frame using a 2D Gaussian distribution (M was set to 15)

Figure: Example of M points generated using Gaussian distribution for M=5

# Strategy Generation

- If the area of the gaussian is too small, real limb pose cannot be discovered and will lead to tracking issues
- Distribution area is determined by the variance $\sigma_r$ of the Gaussian which was set to $\sigma_r = \lambda \times W_0$. $W_0$ is the initialization width of the corresponding limb and $\lambda$ is the the scaling factor set to 0.5 through experiments
- Each two generated points to two correlated joints are connected using straight line segment which tells the orientation of a strategy i.e. all possible limb positions. All line segments are considered as strategies of an agent

# Normal Form Construction

- Represent the game in a matrix form which includes all strategies of each player and their corresponding payoffs
- Having specified M random candidate points for each joint, the total number of strategies for each agent is $N = M \times M$
- Represent the game as $N \times N$ matrix which provides the Payoff for each combination of strategies of the two agents
- The whole body movement is tracked using four independent normal form games, one for each limb between two agents who control the upper and lower part of that limb

# Normal Form Game

Column Player $A_2$

| Row Player $A_1$ | 1 | 2 | ··· | $N$ |
|---|---|---|---|---|
| 1 | $P_1^1(1), P_2^1(1)$ | $P_1^1(2), P_2^2(1)$ | ··· | $P_1^1(N), P_2^N(1)$ |
| 2 | $P_1^2(1), P_2^1(2)$ | $P_1^2(2), P_2^2(2)$ | ··· | $P_1^2(N), P_2^N(2)$ |
| ··· | ··· | ··· | ··· | ··· |
| $N$ | $P_1^N(1), P_2^1(N)$ | $P_1^N(2), P_2^2(N)$ | ··· | $P_1^N(N), P_2^N(N)$ |

Figure: Normal Form Construction

- Here $P_k^i(j)$ represents the payoff of the $i^{th}$ strategy of agent k against the $j^{th}$ strategy of the opponent
- The payoff function comprises of two parts

$$P_k^i(j) = P_{color}^i \times P_{dist}^i(j)$$

# Payoff Function

## Color Payoff Function

- The color payoff of each strategy is measured by comparing the observation of the strategy with its corresponding template initialized in the first frame
- The likelihood between the observation and the limb template is measured by comparing the 2D Hue-Saturation (HS) color histograms
- The metric used for this comparison is the Bhattacharya distance between the two histograms (s and t) with m bins given by

$$B(s,t) = \sqrt{1 - \sum_{u=1}^{m} \sqrt{s(u)t(u)}}$$

- The color payoff function is then defined by

$$P^{i}_{color} = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(\frac{-B_i(s_i, t_i)}{2\sigma_c^2}\right)$$

# Payoff Function

### Distance Payoff Function

- The distance payoff measures the dependency between a strategy and its opponent strategy
- This models the distance between the correlated joints of the strategies of the two agents
- $d_{ij}$ is the distance of the correlated joints of the $i^{th}$ strategy of an agent against the $j^{th}$ strategy of the other agent
- The smaller this distance is, stronger is the dependence
- The distance payoff function is then defined by

$$P^{i}_{dist}(j) = \frac{1}{\sqrt{2\pi}\sigma_d}\exp\left(\frac{-d_{ij}^2}{2\sigma_d^2}\right)$$

# Nash Equilibrium

- By definition, **Nash Equilibrium** is a solution concept of a game involving two or more players, where each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his or her own

- We can arrive at the global optimal solution of limb poses by finding the Nash Equilibria points for the Normal form game described previously

- The process of finding NEs is performed in three major steps:
    1. Find maximum payoffs of row player in every column, and set all the other payoffs of row player to zero
    2. Find maximum payoffs of column player in every row, and set all the other payoffs of column player to zero
    3. In revised normal form, select the entries (pair of strategies) with nonzero payoffs as the NEs

## Why does the algorithm work?

Assuming $P_k^i(j) > 0$ represents the payoff of the $i^{th}$ strategy of agent k against the $j^{th}$ strategy of opponent, if (i, j) is a Nash Equilibrium then

$$P_1^i(j) \geq P_1^{i'}(j) \ \forall \ i' \neq i \quad and \quad P_2^j(i) \geq P_2^{j'}(i) \ \forall \ j' \neq j$$

$\Longrightarrow$ $P_1^i(j)$ is the largest payoff for row player in column j and $P_2^j(i)$ is the largest payoff for column player in row i

$\Longrightarrow$ Setting the payoffs which are not maximum (in every column for row players and every row for column players) to zero, does not affect the Nash Equilibria points

$\Longrightarrow$ If both payoffs remain non-zero at the end of step 2, they are maximum values in their respective columns and rows; therefore no player benefits from unilateral deviation

$\Longrightarrow$ The pair of strategies with nonzero payoffs in the revised normal form are Nash Equilibrium strategies

# Finding Optimal Strategies

- In order to find the optimal solution of tracking, we first calculate the utilities of all the NEs, and then choose from them the NE with the highest utility.
- Given the payoff $P_k^i(j)$ of the i$^{th}$ of the k$^{th}$ player and the payoff $P_q^j(i)$ of the j$^{th}$ strategy of the q$^{th}$ player which construct a NE, the total utility of the NE is calculated according to the following equations

$$U(i,j) = P_k^i(j) + P_q^j(i)$$

- The pair of strategies with highest total utility is selected as the optimal pair

$$(i_{opt}, j_{opt}) = \arg\max_{(i,j)\in NE} U(i,j)$$

# Experimental Results

- The test videos were captured by using a Canon IXUS 8015 digital camera. The frame size was 640x480 pixels, and the frame rate was 30 fps. The proposed algorithm was developed by using C++ and implemented on a desktop with Intel(R) Pentium(R) D 3.00GHz CPU and 3.00GB RAM.
- Game Theory (GT) based method results were compared with then state of the art Particle Filter (PF) based method by calculating average error of all joints.
- In the first 400 frames, the average error of all the joints of PF tracker is **14.08 pixels**, while that of GT tracker is **13.71 pixels**. Even though, the average errors of the two methods are similar, the proposed GT method spends only **1.5 seconds per frame** while the PF tracker takes about **13.5 seconds per frame**.
- The average tracking error for difficult parts (right hand tracking in this test) of GT tracker is **17.68 pixels**, while that of the PF tracker is **29.10 pixels**.

# Experimental Results

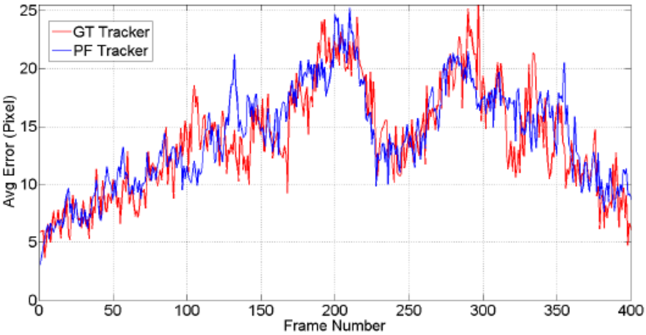


**Fig. 3**. Comparison of average error in pixel between the proposed method and the particle filter based method.

# Experimental Results

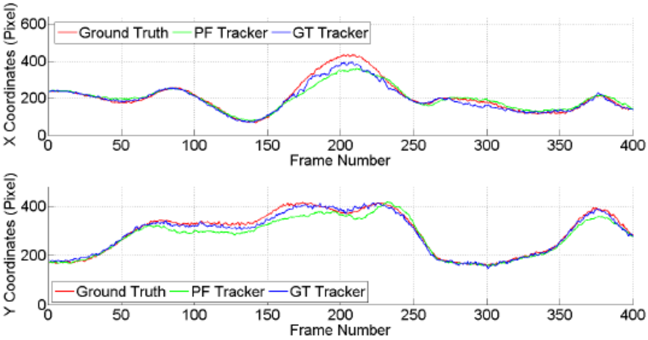


**Fig. 4.** Comparison of right hand tracking between the proposed method and the particle filter based method.

# Experimental Results

**Fig. 5.** Human pose tracking result of the proposed GT tracker. The video contains 1127 frames.

# Conclusion

- A game theory based method for tracking articulated poses was proposed which does not require a learning process to learn prescribed motions
- The global optimum of each two articulated limbs is located by finding the NEs of two agents playing normal form game. Each player only needs to optimize its own utility, the computational complexity is reduced greatly.
- The experiments prove the effectiveness and efficiency of the proposed method. In future work, they aim to research on finding Nash equilibria for multi-player games and apply it to finding global optima for all the limbs.

# References


1. Articulated human pose tracking based on game theory
2. Game-Theoretic Multiple Target Tracking
3. Deep Learning based Human Pose Tracking

Thank You!