Real-time 3D Human Pose Estimation with a Single RGB Camera

BY:
SIDDHANT ARORA(2015CS50480)
MAKKUNDA SHARMA(2015CS50459)

Source:

- We have implemented the paper :”*VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera*” by Mehta et al. from mpi-inf presented at SIGGRAPH 2017

Problem Motivation :

- This approach is the first monocular RGB method usable in real-time applications such as 3D character control ,Virtual Reality ,Ubiquitous Motion Capture with Smartphones.
- Virtual Reality has many applications and this method enables them from a single consumer color camera.
- Real-time motion capture solutions provide a natural interface for game characters and virtual avatars, which go beyond classical mouse and gamepad control.

Problem Motivation :

- Motion capture with smartphones : By streaming the video to a machine with sufficient capabilities for the algorithm, one can turn any smart phone into a lightweight, fully-automatic, handheld motion capture sensor turning smartphones into motion capture devices for casual users without additional sensing devices.
- In short we will be getting a rgb-d camera (say kinect) like output with a single monocular rgb camera and the approach is more broadly applicable than RGB-D solutions, i.e., it works for outdoor scenes, community videos, and we only need atmost low quality commodity RGB cameras.

Project Pipeline

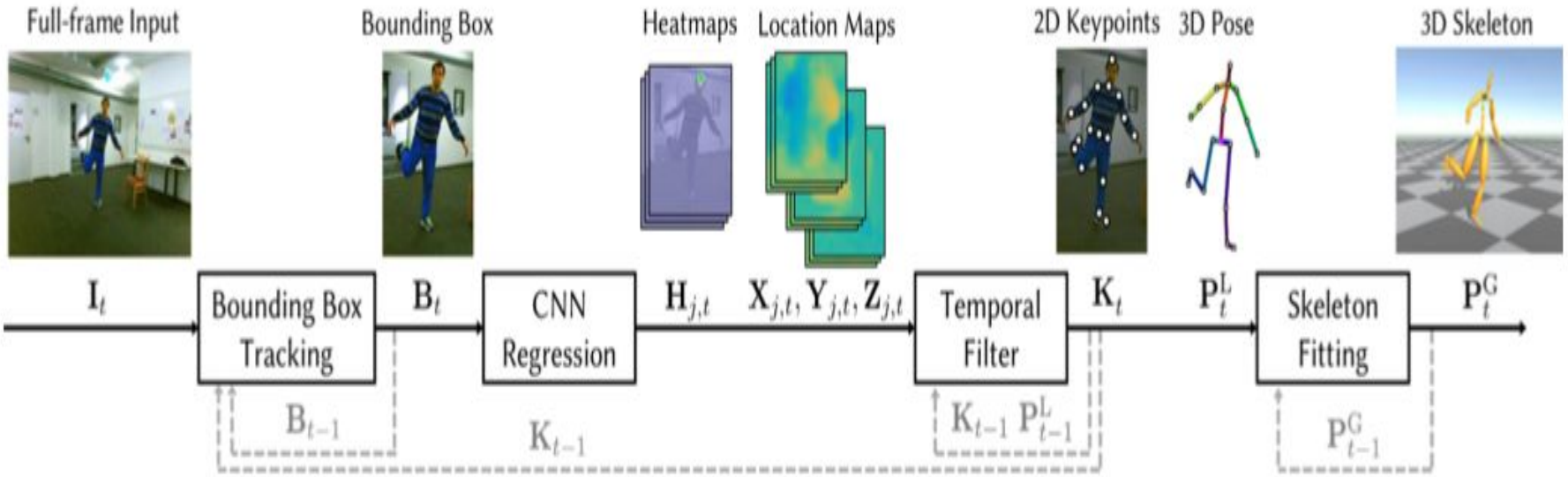

Image Source : The VNECT Paper

IMPLEMENTATION

- The Implementation was done in matlab , and we used Caffe for the CNN.
- Our implementation mainly comprised of 4 Stages-

1. Bounding Box Tracker
2. CNN Pose Regression
3. Temporal Filtering
4. Kinematic Skeleton Fitting

Bounding Box Tracker

WHY :

- By repeated experimentation, it was observed that runtime and accuracy of CNN was highly dependent on size of input image.
- Additionally, the CNN is trained for subject sizes in the range of 250–340 px in the frame, requiring averaging of predictions at multiple image scales per frame (scale space search) if processing the full frame at each time step.
- Guaranteeing real-time rates necessitates restricting the size of the input to the network and tracking the scale of the person in the image to avoid searching the scale space in each frame.

Bounding Box Tracker

IMPLEMENTATION:

- The 2D pose predictions from the CNN at each frame are used to determine the BB for the next frame
- We make a slightly larger box around the predictions.
- The smallest rectangle containing the key points.K is computed and augmented with a buffer area to obtain bounding box for next frame.

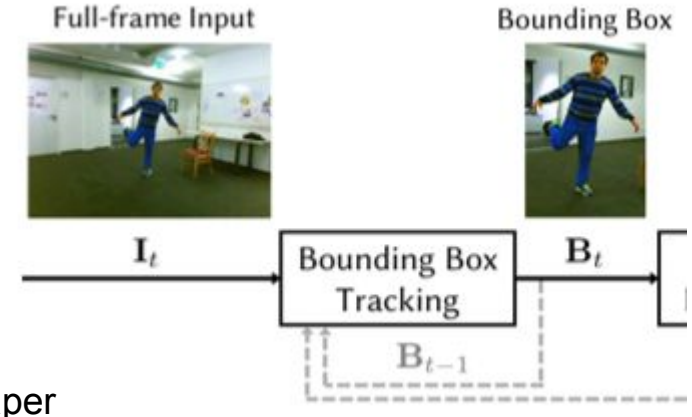

Image Source : The VNECT Paper

CNN Pose Regression

- The goal of CNN pose regression is to obtain an estimate joint positions, both,in 2D image space and 3D.
- The heatmap based formulation naturally ties image evidence to pose estimation by predicting a confidence heatmap $H_{j,t}$ over the image plane for each joint $j \in \{1..J\}$.
- In addition to this, it extends 2D Heatmap formulation to 3D by providing 3 additional location maps for each joint which capturing the root-relative locations x_j , y_j and z_j respectively.
- We will be using the CNN kindly provided to us by original paper formulators.

Kinematic Skeleton Fitting

This goal is achieved mainly by following 2 processes-

1. Temporal Filtering by using Euro filter
2. Minimising total objective energy using Levenberg–Marquardt Algorithm

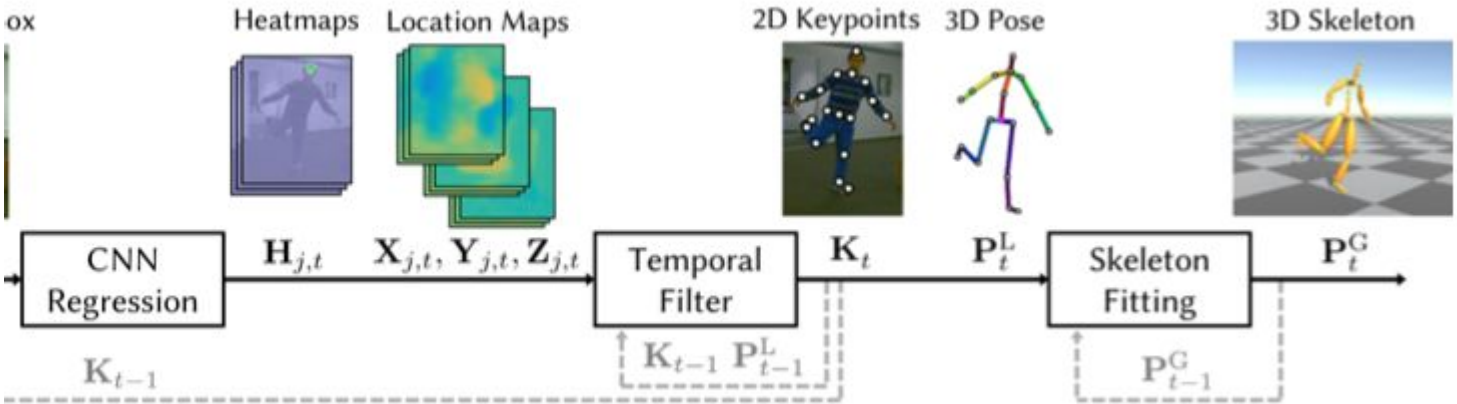

Image Source : The VNECT Paper

Euro Filter

- The temporal filter used was the 1 euro filter by Casiez et al. (2012) as it is a lightweight filter and it has been specifically designed to minimize jitter and lag when tracking human motion.
- It uses a first order low-pass filter with an adaptive cutoff frequency: at low speeds, a low cutoff stabilizes the signal by reducing jitter, but as speed increases, the cutoff is increased to reduce lag.
- It is lightweight and has very little lag which are beneficial when writing a real time implementation of the complete algorithm

Minimising Total Objective Energy

It was observed that to obtain temporal consistencies and to reduce pose inaccuracies we need to minimise 4 basic quantities-

1. Inverse kinematic term
2. Projective term
3. Temporal smoothening term
4. Depth smoothening term

$$E_{\text{total}}(\boldsymbol{\theta}, \mathbf{d}) = E_{\text{IK}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{proj}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{smooth}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{depth}}(\boldsymbol{\theta}, \mathbf{d}),$$

Inverse Kinematic Term

This term determines overall similarity of pose to 3D CNN output.It is implemented with L2 Loss.

$$E_{\mathrm{IK}} = \|(\mathbf{P}_t^{\mathrm{G}} - \mathbf{d}) - \mathbf{P}_t^{\mathrm{L}}\|_2.$$

- Where P_t^{G} is final 3D pose .
- d determines global position(will be corrected using projective term which takes pose from root based frame to 3D world frame fitted on camera)
- P_t^{L} is 3D CNN Output.

Projection Term

- The projection term E_{proj} determines global position d and corrects the 3D pose by re-projection onto the detected 2D keypoints.
- We assume the pinhole projection model. If the camera calibration is unknown a vertical field of view of 54 degrees is assumed to get a generic camera matrix.
- Here || is projection from 3D plane to image plane.

$$E_{\text{proj}} = \| \Pi(\mathbf{P}_t^G) - \mathbf{K}_t \|_2$$

Temporal Smoothening

- Temporal stability is enforced with smoothness prior
- We are penalizing the acceleration P_t^G to bring temporal stability.

$$E_{\text{smooth}} = \|\widehat{\mathbf{P}_t^G}\|_2$$

Depth Smoothening

- To counteract the strong depth uncertainty in monocular reconstruction, we penalize large variations in depth additionally with E_{depth}
- Here the term is the Z component of the 3d velocity $\widetilde{\mathbf{P}}_t^G$

$$E_{\text{depth}} = \|[\widetilde{\mathbf{P}}_t^G]_z\|_2$$

Least Squares Minimisation

- We finally do a weighted sum of these 4 quantities to obtain total objective function (Weights obtained empirically).
- Then we run the iterative Levenberg–Marquardt algorithm to get 3D pose which minimizes the objective function
- We pass this 3D pose through another temporal filter to get the final 3D output
- To get the final 2D pose in the image plane, we again project it using the camera matrix

$$E_{\text{total}}(\boldsymbol{\theta}, \mathbf{d}) = E_{\text{IK}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{proj}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{smooth}}(\boldsymbol{\theta}, \mathbf{d}) + E_{\text{depth}}(\boldsymbol{\theta}, \mathbf{d}),$$

References:

- Mehta et al. . VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera (SIGGRAPH 2017)
- Casiez, G., Roussel, N. and Vogel, D. (2012). 1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '12).