# Stereo Human Keypoint Estimation

Kyle Brown
Stanford University
Stanford Intelligent Systems Laboratory
kjbrown7@stanford.edu

## Abstract

*The goal of this project is to accurately estimate human keypoint coordinates in 3-dimensional space without markers. A deep convolutional neural network (CNN) is trained on annotated data to estimate keypoint coordinates in 2D. Twin instances of the network are deployed in a stereo configuration. At each time step, each instance of the CNN receives a 2D projection of a scene containing a human, and outputs a set of 2D keypoint estimations corresponding to its vantage point. The two sets of predictions are combined to produce a (perhaps very) rough estimate of the target individual's 3D articulated pose. Observations over multiple time steps are incorporated into an iterative optimization procedure that continuously refines an estimate of target individual's skeletal structure to yield increasingly accurate estimations of instantaneous pose.*

## 1. Introduction

Body language is an important mode of human-to-human communication. The way we move says a great deal about our intentions. An artificial agent that can accurately estimate 3D human pose (especially for an arbitrary number of humans simultaneously) in real time is well on its way to effective, safe, and complex interaction with humans. This is a key "skill" for a wide variety of autonomous agents. Consider, for example, the case of an autonomous vehicle. At a bare minimum, the vehicle must be able to detect and roughly localize pedestrians. Obviously this is prerequisite to avoiding fatal accidents. However, detection and rough localization don't always cut it. What if a police officer standing at an intersection uses hand signals to direct traffic? Will the autonomous vehicle be able to recognize and interpret the officer's commands? Or will the car freeze, unable to comprehend anything more about the situation than the fact that a pedestrian is standing in the road?

This paper approaches the problem of human pose estimation within the context of autonomous driving. Specifically, we consider a front-facing stereo camera configuration with cameras placed at the front left and right corners of the windshield. Twin instances of a deep convolutional neural network are deployed on each camera feed, and the 2D key-point predictions from each network are combined via an iterative optimization procedure that continuously refines an estimate of target individual's skeletal structure to yield increasingly accurate estimations of instantaneous pose.

One may question the value of deploying twin instances of a CNN to estimate key-points in 3D. After all, why not use standard stereo reconstruction to obtain a dense depth map of a scene, including the 3D geometry of any humans in the scene? We acknowledge that an ideal approach would be to learn 3D articulated pose directly from 3D data (such as a stereo depth map, lidar data, or any other depth sensor). However, as far as the authors are aware, there exists no dataset of 3D depth data densely annotated with 3D key-points for supervised learning. We therefore take the approach outlines above. It is worth noting, however, that our approach may make it possible to *create* densely annotated 3D datasets for supervised training at low cost.

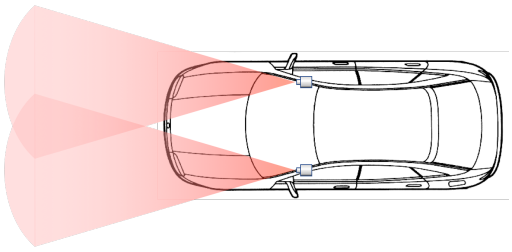

Figure 1. Diagram of the stereo camera configuration

## 2. Related Work

As with virtually every domain in computer vision in recent years, deep learning has created a revolution in the field of human key-point estimation. Many different deep architectures have achieved impressive accuracy on the various publicly available human key-point datasets. *He et. al* report state-of-the-art results on the Microsoft COCO

dataset with the versatile Mask R-CNN architecture. They approach the key-point estimation task in a straightforward manner, by training the network on one-hot binary masks (each representing a key-point) without incorporating any domain knowledge about human anatomy. Other approaches to pose estimation involve iterative refinement of a location estimate for a given body part based on the estimated positions of other body parts. *Ramakrishna et. al* take such an approach in Pose Machines [2], as do *Wei et. al* in Convolution Pose Machines [4]. *Newell et. al* use a stacked Hourglass architecture to synthesize spatial data at all across the entire image [1]. Tompson et al approach the task via joint training of a CNN and graphical model [3].

Our method for keypoint estimation bears the most resemblance to Mask R-CNN (without the region-proposal part) and Stacked Hourglasses Networks.

## 3. Methods

As of this writing, satisfactory results have not been achieved in the first stage (keypoint estimation via deep convolutional network) of the processing pipeline. We thus approach the 2nd stage (iterative refinement of 3D articulated pose) with simulated data intended to mimic the behavior that might be expected of a properly functioning deep network. We hope to be able to plug the network into the pipeline once it is performing well enough to be useful.

We want to estimate the following 17 human keypoints: *Head, Throat, Shoulders, Elbows, Hands, Low-back, Hips, Knees, Ankles, and Feet*. At each time step we simulate the output of our twin network instances by a set of heatmaps, one per key-point per network instance (for a total of $2 \times 17 = 34$). Each heatmap is generated by randomly shifting the ground truth 2D key-point projection, and scattering random "votes" centered at the perturbed location, This yields a shifted, blurry "hot spot" that represents the network instance's pixel-level prediction map for the corresponding keypoint.

The blurry heatmaps are thresholded via non-maximum suppression–all but the top $N$ pixels are set to zero. Thus, only the high-confidence pixels get to "vote" in the 3D estimation phase.

Figures 2 through 5 demonstrate that a naive weighted least squares estimate of 3D articulated pose from the 2D heat maps falls short of the desired accuracy.

### 3.1. Cost Function

In order to gain a more robust and accurate estimate of 3D articulated pose over time, we define a cost function that incorporates observations across multiple video frames and imposes anatomical constraints on the 3D pose estimation. Specifically, we maintain a running average of the observed limb lengths between connected key-points (i.e. distance from ankle to knee, from knee to hip, etc.), and impose

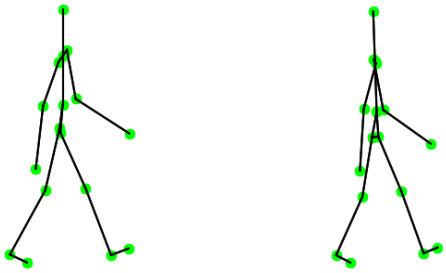

Figure 2. 2D projections from each camera's view of the ground truth 3D articulated pose for the simulated human.

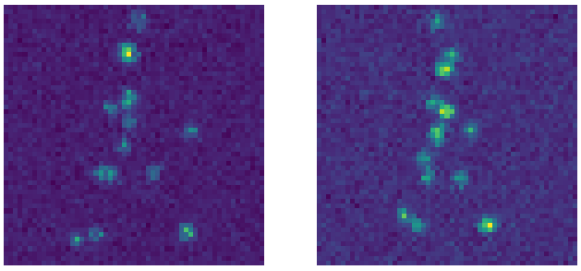

Figure 3. Flattened representation (for convenient display) of the simulated keypoint heatmaps for each camera view. Recall that there are 17 heatmaps per camera–one for each keypoint.

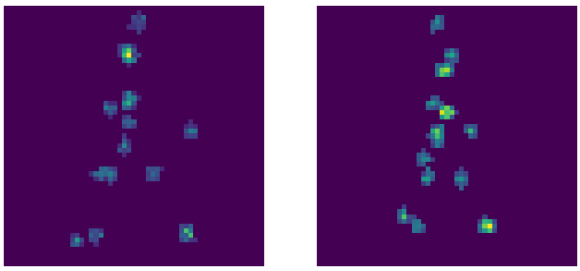

Figure 4. Flattened representation of the thresholded keypoint heatmaps for each camera view.

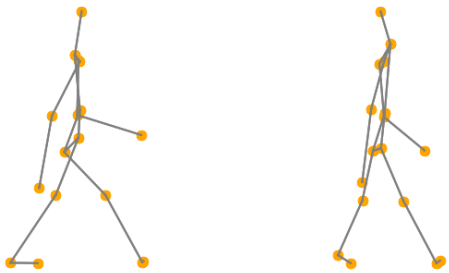

Figure 5. 2D projections of the (very poor) 3D pose estimate derived from a naive weighted least squares estimation.

a quadratic penalty for deviation of the reconstructed 3D skeleton from the estimated limb lengths. This penalty is combined with a reprojection error cost, which applies a penalty to deviation of each reprojected key point from the region proposed by the corresponding heat map. To restate succinctly, the skeleton is penalized is its limbs are too long or too short, and if its joints show up far from the positions estimated by the heat maps. Our optimization objective thus takes the following form:

$$
\begin{aligned}
J = \sum_{k}^{K} \Bigg( & \\
& \sum_{j}^{N} \Big( b_{1,k,j} ||\mathbf{M_1} \hat{P}_k - p_{1,k,j}||^2 \\
& + b_{2,k,j} ||\mathbf{M_2} \hat{P}_k - p_{2,k,j}||^2 \Big) \\
& + \sum_{i \neq k} \mathbf{W}_{i,k} \Big|\Big| ||\hat{P}_i - \hat{P}_k|| - \mathbf{L}_{i,k} \Big|\Big|^2 \Bigg)
\end{aligned}
\tag{1}
$$

where $K$ is the number of key-points being estimated, $N$ is the number of "voting" pixels per heat map, $b_{1,k,j}$ is the intensity or "voting" magnitude of the $j^{th}$ pixel in the heat map for key point $k$ in camera frame 1, $\mathbf{M}_1$ is the camera matrix for camera frame 1 (i.e. $\mathbf{M}_1 \hat{P}_k$ is a projective transform mapping 3D point $\hat{P}_k$ to 2D camera coordinate frame 1), $\hat{P}_k$ is the estimated 3D location of the $k^{th}$ key point, $\mathbf{W}_{i,k}$ is the weight or confidence assigned to the running average value $\mathbf{L}_{i,k}$, which represents the expected distance between *connected* key points $i$ and $k$. $\mathbf{W}_{i,k} = 0$ between key points that are not directly connected at a kinematic distance of 1. For example, $\mathbf{W}_{head,rightankle} = 0$ because the head is several joint connections away ankle.

Thus, the first inner summation term in 1 is the reprojection error cost, and the second inner summation term is the anatomical constraint cost

### 3.2. Optimization

At each time step we employ an iterative optimization scheme, which is set out in Algorithm 1. The first phase uses weighted linear least squares regression to produce an initial estimate $\hat{P}$ of the 3D key point coordinates from the voting points extracted from the heat maps. This estimate is used to update the matrices $\mathbf{L}$ and $\mathbf{W}$ containing the running average limb lengths and associated confidence weights.

The second phase of the iterative optimization algorithm traverses the point estimates and uses Newton's method to update each estimate $\hat{P}_k$ based on the associated reprojective and anatomical cost functions. The Hessian matrix is trivial to compute, as its dimensionality is determined by the degree of connectivity of $\hat{P}_k$ and therefore never exceeds $4 \times 4$.

**input :** $\mathcal{H}_1, \mathcal{H}_2, \mathbf{L}, \mathbf{W}, \mathbf{M_1}, \mathbf{M_2}, \alpha$
**output:** Estimated 3D key points $\hat{P}$

*Extract the "voting" points*;
$\mathbf{b}_1, \mathbf{p_1} \leftarrow$ `GetVotes` $(H_1)$
$\mathbf{b}_2, \mathbf{p_2} \leftarrow$ `GetVotes` $(H_2)$
*Compute an initial estimate for* $\hat{P}$
**for** $k \leftarrow 1$ **to** $K$ **do**
  $\hat{P}_k \Leftarrow$ `NaiveLSQ` $(\mathbf{b}_{1,k}, \mathbf{p_{1,k}}, \mathbf{b}_{2,k}, \mathbf{p_{2,k}})$
**end**
*Update running averages and confidence weights*
$\mathbf{L}, \mathbf{W} \leftarrow$ `UpdateL` $(\mathbf{L}, \mathbf{W}, \mathbf{b}_1, \mathbf{p}_1, \mathbf{b}_2, \mathbf{p}_2,)$
*Iterate over the full set of points* $T$ *times*
**for** $t \leftarrow 1$ **to** $T$ **do**
  **for** $k \leftarrow 1$ **to** $K$ **do**
    **for** $i \leftarrow 1$ **to** *MAX_ITERS* **do**
      ***Compute Reprojection cost:***
      $J_R \leftarrow$
       $J_R(\mathbf{M_1}, \mathbf{M_2}, \hat{P}_k, \mathbf{b}_{1,k}, \mathbf{p}_{1,k}, \mathbf{b}_{2,k}, \mathbf{p}_{2,k})$
      ***Compute Anatomic Cost:***
      $J_A \leftarrow J_A(\hat{P}_k \mathbf{L}, \mathbf{W})$
      $J \leftarrow J_R + J_A$
      ***Newton's method update:***
      $\mathbf{g} \leftarrow \nabla J$
      $\mathbf{H} \Leftarrow \nabla^2 J$
      $\hat{P}_k \leftarrow \hat{P}_k - \alpha \mathbf{H}^{-1} \mathbf{g}$
    **end**
  **end**
**end**

**Algorithm 1:** Iterative Keypoint Estimation for a single time step

## 4. Experiments and Results

The optimization algorithm was tested on simulated data from a human walking trajectory. The noisy heat map predictions were obtained from ground truth data as described in the previous section. Results are evaluated using two metrics: *absolute length prediction error* compares the ground truth values of $\mathbf{L}$ to the values obtained from the $\hat{P}$ estimation. *squared keypoint error* compares the locations of the ground truth keypoint locations vs. the $\hat{P}$ estimation.

Visual results of a few examples comparing naive vs. final predictions can be seen in figures 6, 7, 8, and 9. Note that the final prediction does not really yield a substantial gain over the initial naive estimation. In fact, the opposite is true! The optimization algorithm yields results that are *worse* than the initial estimate. This can be seen in figure 10. We puzzled over this, and determined that the blame lies squarely on the shoulders of the running average limb-length estimator.

Whereas one might intuitively expect a running average

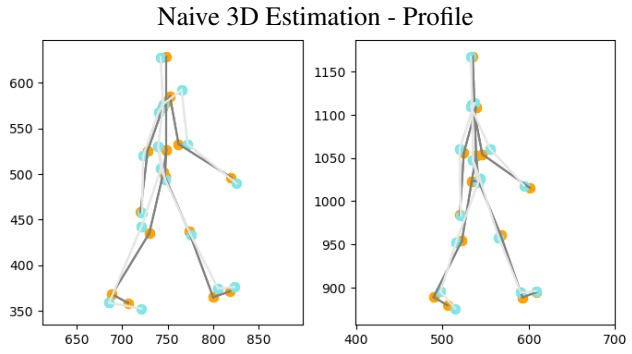


Figure 6. 2D projections of the naive least squares 3D key point estimate (blue) compared with the ground truth (orange).

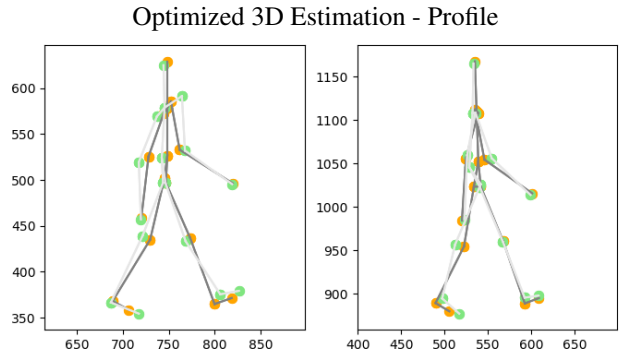


Figure 7. 2D projections of the final 3D key point estimate (blue) compared with the ground truth (orange).

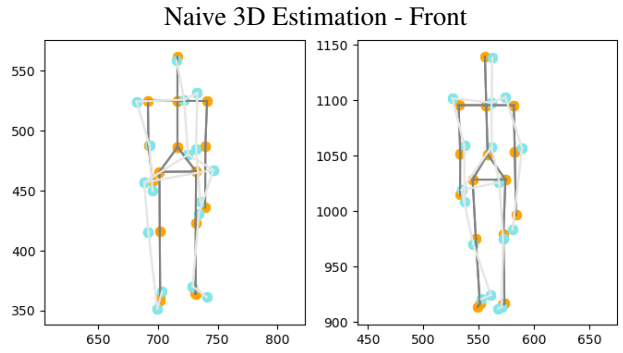


Figure 8. 2D projections of the naive least squares 3D key point estimate (blue) compared with the ground truth (orange).

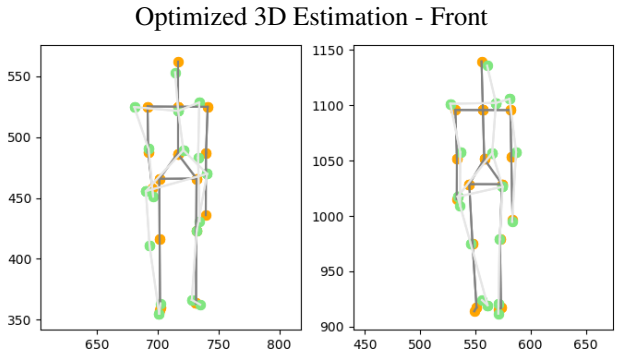


Figure 9. 2D projections of the final 3D key point estimate (blue) compared with the ground truth (orange).

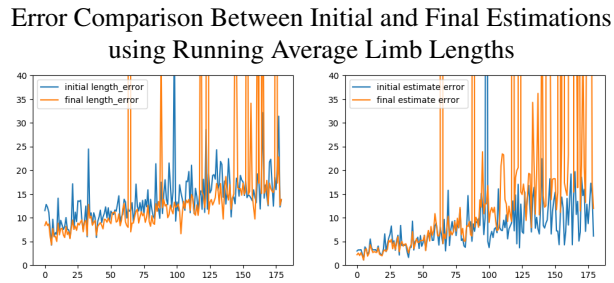


Figure 10. On the left, absolute error in limb length prediction. Note that the final error (orange) is worse than the initial! On the right, squared 3D localization error for key point estimation. Note that, once again, the final results (orange) is worse than the initial guess.

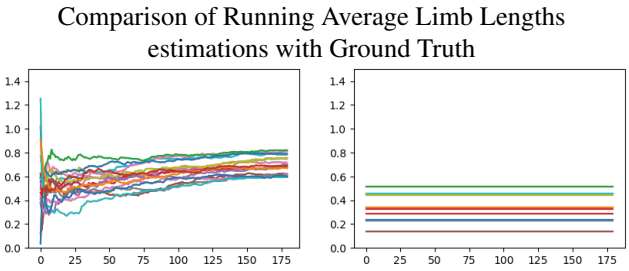


Figure 11. The above plot shows a systematic positive bias that accrues in the running averages for limb length over time. On the left are the running averages, on the right are the corresponding ground truth values, corresponding in color. This effect is not mitigated by applying a decay operation to the running average.

to converge to the true limb length value, this assumption is flawed when the noisy limb length observations are independent. Because we are adding noise at independently at each key point to simulate the heat map estimates, our average limb length accrues a positive bias over as observations accumulate. As the perturbations all share the same standard deviation, this artificial positive bias is even more pronounced for key points that should naturally fall very close together (like ankles and toes, for example). This effect is clearly visible in figure 11.

To test the hypothesis that the running average estimates were indeed the source of failure, we performed the optimization using ground truth limb lengths in place of the averages. We even increased the noise added to the heatmaps. Sure enough, the enormous gap between initial and final es-

Error Comparison Between Initial and Final Estimations using Ground Truth Limb Lengths

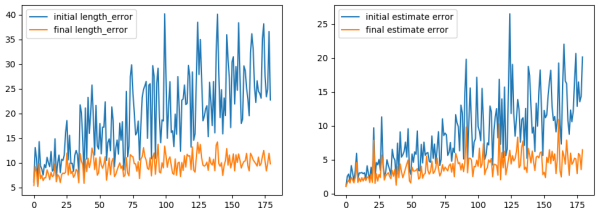


Figure 12. On the left, absolute error in limb length prediction. Note that the final error (green) is nearly identical to the initial. On the right, squared 3D localization error for key point estimation. Note that, once again, the final results (orange) are identical to those for the initial estimation.

timation errors reversed itself. Figure 12 shows that, when using ground truth limb lengths in place of running average estimates, the optimization algorithm does indeed *optimize* the 3D estimation. Examples of the 3D key point estimation for this scenario can be seen in figures 13 and 14.

Naive 3D Estimation at 45 degrees

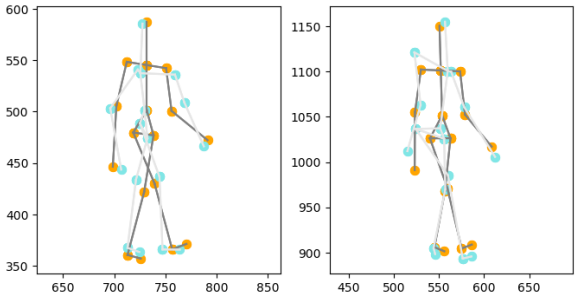

Figure 13. 2D projections of the Naive 3D key point estimate (blue) compared with the ground truth (orange).

Optimized 3D Estimation at 45 degrees using Ground Truth Limb Lengths

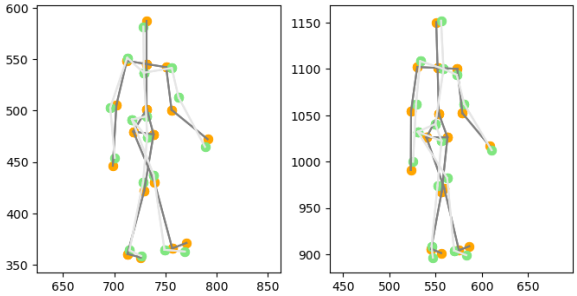

Figure 14. 2D projections of the final 3D key point estimate (green) compared with the ground truth (orange). Note that this result was achieved with the ground truth limb lengths in place of running average estimates.

We thus conclude that the flaw in our initial results is entirely attributable to the running average estimator. The optimization algorithm itself functions as expected. We note that a rather straightforward mapping function could be introduced to correct for the systematic bias, but we consider that this is not particularly useful unless coupled with an analysis of the neural network prediction distribution. Hence, we conclude our analysis here.

## 5. Conclusion

We have shown that the optimization approach presented herein is effective *if* measures are taken to assure that the limb length estimations do not accrue positive bias. An augmented approach, perhaps incorporating Bayesian Filtering, may have potential to significantly enhance performance. We look forward to implementing the full pipeline with a deep neural network in place of our simulationg. We reaffirm that 3D estimation of articulated human pose is an important step toward enabling autonomous systems to understand human body language.

## 6. Appendices

## References

[1] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. 3 2016. 2

[2] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose Machines: Articulated Pose Estimation via Inference Machines. 2

[3] J. Tompson, A. Jain, Y. Lecun, and C. Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. 2

[4] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. 1 2016. 2