Simultaneous Human Segmentation, Depth and Pose Estimation via Dual Decomposition

Glenn Sheasby[1]
glenn.sheasby@brookes.ac.uk
Jonathan Warrell[1]
jwarrell@brookes.ac.uk
Yuhang Zhang[2]
yuhang.zhang@anu.edu.au
Nigel Crook[1]
ncrook@brookes.ac.uk
Philip H.S. Torr[1]
philiptorr@brookes.ac.uk

[1] Brookes Vision Group
Oxford Brookes University
Oxford, UK

[2] Research School of Computer Science
Australian National University
Canberra, ACT, Australia

Abstract

The tasks of stereo matching, segmentation, and human pose estimation have been popular in computer vision in recent years, but attempts to combine the three tasks have so far resulted in compromises: either using infra-red cameras, or a greatly simplified body model. We propose a framework for estimating a detailed human skeleton in 3D from a stereo pair of images. Within this framework, we define an energy function that incorporates the relationship between the segmentation results, the pose estimation results, and the disparity space image. Specifically, we codify the assertions that foreground pixels should relate to some body part, should correspond to a continuous surface in the disparity space image, and should be closer to the camera than the surrounding background pixels. Our energy function is NP-hard, however we show how to efficiently optimize a relaxation of it using dual decomposition. We show that applying this approach leads to improved results in all three tasks, and also introduce an extensive and challenging new dataset, which we use as a benchmark for evaluating 3D human pose estimation.

1 Introduction

Two tasks that have attracted a great deal of work from vision researchers over the years are the estimation of human pose in images, and segmentation of humans from a scene. Despite the large body of research focusing on 2D human pose estimation, relatively little work has been done to estimate pose in 3D.

The main objective of human pose estimation is often formalized by defining a skeleton model, which is to be fitted to the image. It is quite common to describe the human body as an articulated object, *i.e.* one formed of a connected set of rigid parts. On the other hand,

the goal of segmentation is simply to specify the set of pixels which contain the human. Human pose estimation and segmentation have a wide variety of applications, including video gaming [23], security [16], hazard detection in automobiles [8], and photo substitution [21].

In order to be useful for such applications, the results are required to be very accurate. However, various challenges arise, such as self-occlusion, where one body part obscures another; and for pose estimation, inter-part similarity, where different parts are very similar in appearance. This problem is chiefly noticeable when comparing opposite limbs.

Classical pose estimation algorithms run on a single RGB image. Several of these use simple models formed of 6 parts (upper body) or 10 parts (full body) [3, 9, 10]. Kumar *et al*. [13] use message passing to learn the relationship between adjacent parts, while Andriluka *et al*. [2] use a kinematic tree prior: part relations are specified by a directed acyclic graph, and the optimal skeleton is found jointly. Yang and Ramanan extend this approach by introducing a flexible mixture of parts model, allowing for greater intra-limb variation [25]. However, restricting oneself to a single RGB image means that the aforementioned issue of self-occlusion is very difficult to deal with.

Recently, the development of accurate, high resolution depth cameras such as the Microsoft Kinect [1] has improved performance [22], but due to infra-red interference, the Kinect fails in outdoor scenes. An alternative approach, which we follow in this work, is to use a stereo pair of cameras to build the depth image.

Stereo correspondence algorithms typically denote one image as the *reference image* and the other as the *target image*. A dense set of patches is extracted from the reference image, and for each of these patches, the best match is found in the target image. These matches are combined to form the *disparity map*. While there has been much research into the best way to create this disparity map [7, 11, 15, 18, 19], we find that an approach based upon a simple matching cost is sufficient to provide a reasonable disparity map.

The task of combining multiple vision algorithms to produce a rich understanding of a scene is one that has been extensively covered in the vision literature [3, 6, 14, 17]. The problem with putting the algorithms into a pipeline, where the result of one algorithm is used to drive the other, is that it is often impossible to recover from errors made in the early stages of the process. This problem can be ameliorated by joint inference, as proposed by Wang and Koller [24]. By constructing a multi-level inference framework, they are able to use dual decomposition [12] in order to simultaneously provide segmentation and pose estimation of humans.

In this work, we extend the formulation of [24] to include stereo, so that we can provide segmentation and pose estimation in 3D, not just 2D. While existing stand-alone stereo correspondence algorithms are not sufficiently accurate to compensate for the lack of an infrared sensor, our multi-level inference framework aids us in segmenting objects despite errors in the disparity map. The contributions of this paper can be summarized as follows: we present a novel dual decomposition framework to combine stereo algorithms with pose estimation and segmentation. Our system is fully automated, and is applicable to more general tasks involving object segmentation, stereo and pose estimation. Drawing these together, we demonstrate a proof of concept that the achievements of Kinect can be matched using a stereo pair of images, instead of using infra-red depth data. We also provide an extensive new dataset of humans in stereo, featuring nearly 9,000 annotated images.

The remainder of the paper is organized as follows: our formulations of the three optimization problems are laid out in Section 2, while Section 3 explains how we unify the three methods. Results follow in Section 4, and conclusions are given in Section 5.

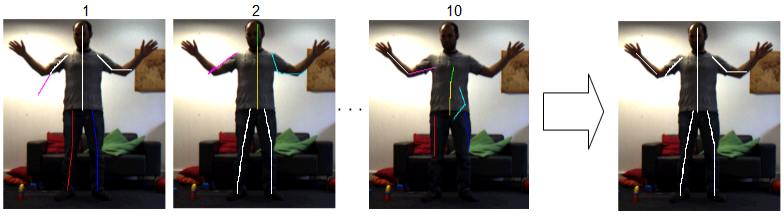


Figure 1: Several complete pose estimations are obtained from Yang and Ramanan's algorithm, which we split into ten body parts. We then select each part individually (one head, one torso, *etc.*). In this example, the parts highlighted in white are selected, enabling us to recover from errors such as the left forearm being misplaced in the first estimate.

2 Problem formulation

The energy function which we wish to optimize consists of three main parts: stereo, segmentation, and human pose estimation. Each of these are represented by one term in the energy function. We introduce two additional terms, hereafter referred to as *joining terms*, which combine information from two of the parts, encouraging them to be consistent with one another. Throughout this paper, we use the subscript $m = (x, y)$ to refer to a pixel, i for a part index, j for a proposal index, and k for a disparity value.

We take as input a stereo pair of images $\mathcal{L}$ and $\mathcal{R}$, and as a preprocessing step, we use the algorithm of Yang and Ramanan [25] to obtain a number N_E of proposals for N_P different body parts. In this paper, we use $N_P = 10$, with two parts for each of the four limbs, plus one each for the head and torso. Each proposal j for each part i comes with a pair of endpoints corresponding to a line segment in the image, representing the limb (or skull, or spine).

Our approach is formulated as a conditional random field (CRF) with two sets of random variables: one set covering the image pixels $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_N\}$, and one covering the body parts $\mathbf{B} = \{B_1, B_2, \ldots, B_{10}\}$. Any possible assignment of labels to the random variables will be called a *labelling* and denoted by $\mathbf{z}$. In a particular labelling $\mathbf{z}$, each pixel variable Z_m takes a label $z_m = [d_m, s_m]$, from the product space of disparity and segmentation labels $\mathcal{D} \times \mathcal{S}$, and each part B_i takes a label $b_i \in \mathcal{B} = \{0, 1, \ldots, N_E - 1\}$, denoting which proposal for part i has been selected. In general, the energy of $\mathbf{z}$ can be written as:

$$E(\mathbf{z}) = f_D(\mathbf{z}) + f_S(\mathbf{z}) + f_P(\mathbf{z}) + f_{PS}(\mathbf{z}) + f_{SD}(\mathbf{z}), \tag{1}$$

where f_D gives the cost of the disparity label assignment $\{d_m\}_{i=m}^{N}$, f_S gives the cost of the segmentation label assignment $\{s_m\}_{i=m}^{N}$, f_P gives the cost of the part proposal selection $\{b_i\}_{i=1}^{N_P}$, and f_{PS} and f_{SD} are the joining terms. Each term contains weights $\gamma_* \in \mathbb{R}^+$, which are learned by gradient ascent. In the following sections, we describe in turn each of the five terms.

2.1 Segmentation term

In order to build unary potentials for the segmentation term, we create a foreground weight map based on the pose detections obtained from Yang and Ramanan's algorithm. For each pixel m, each part proposal (i, j) contributes a weight w_{ij}^m, where $w_{ij}^m = 1$ if m lies directly on

the line segment representing the limb, and decreases exponentially as we move away from it. We then have a foreground weight $W_F = \sum_{i,j} w_{ij}^m$ and a background weight $\sum_{i,j}(1-w_{ij}^m)$ for each pixel. These weights are then used to fit Gaussian Mixture Models for the foreground and background regions, which together give us a posterior probability p_m of each pixel m being foreground. From this, we obtain unary costs $\theta_F = -\log(p_m)$ and $\theta_B = -\log(1-p_m)$, which store the costs of assigning each pixel m to foreground and background respectively.

We also have pairwise costs ϕ_S which store the cost of assigning adjacent pixels to different labels. Defining the set of neighboring pixels C, we follow equation (11) in Rother *et al.* [20], and write the pairwise energy as in (4) below. The energy we have to minimize is the following, with weighting terms γ_1 and γ_2:

$$f_S(\mathbf{z}) = \gamma_1 \cdot \theta_S(\mathbf{z}) + \gamma_2 \cdot \phi_S(\mathbf{z}), \tag{2}$$

where:
$$\theta_S(\mathbf{z}) = \sum_{Z_m \in \mathbf{Z}} s_m \cdot \theta_F(m) + (1-s_m)\cdot \theta_B(m); \tag{3}$$

$$\phi_S(\mathbf{z}) = \sum_{(m_1,m_2)\in C} \mathbf{1}(s_{m_1} \neq s_{m_2}) \exp(-\beta \left\| \mathcal{L}(m_1) - \mathcal{L}(m_2) \right\|^2). \tag{4}$$

2.2 Pose estimation term

Recall that Yang and Ramanan's algorithm provides us with a discrete set of part proposals. Each proposal j for each part i has a unary cost $\theta_P(i,j)$ associated with it, whose value is based on the weights w_{ij} defined in the previous section. The cost is also weighted according the estimate index j, since lower-ranked estimates are less likely to be correct.

A pairwise term ϕ_{i_1,i_2} is introduced to penalize the case where, for two parts that should be connected (e.g. upper and lower left leg), two proposals are selected that are distant from one another in image space. We define a tree-structured set of edges $\mathcal{T}$ over the set of parts, where $(i_1,i_2) \in \mathcal{T}$ if and only if parts i_1 and i_2 are connected. For each connected pair of parts $(i_1,i_2) \in \mathcal{T}$, we model the joint by a three dimensional Gaussian distribution over the relative position and angle between the two parts, using the training set to compute the mean and variance for each part and dimension. We minimize the following cost function:

$$f_P(\mathbf{z}) = \gamma_3 \cdot \sum_{i=1}^{10} \theta_P(i,b_i) + \gamma_4 \cdot \sum_{(i_1,i_2)\in\mathcal{T}} \phi_{i_1,i_2}(b_{i_1},b_{i_2}), \tag{5}$$

where:
$$\theta_P(i,b_i) = \exp\left(\frac{b_i}{2}\right) \sum_{Z_m \in \mathbf{Z}} w_{(i,b_i)}^m (1-p_m). \tag{6}$$

2.3 Stereo term

A particular disparity label d corresponds to matching the pixel (x,y) in $\mathcal{L}$ to the pixel $(x-d,y)$ in $\mathcal{R}$. We define a cost volume θ_D, which for each pixel $m=(x,y)$, specifies the cost of assigning a disparity label d_m. These costs incorporate the gradient in the x-direction (in $\Delta\mathcal{L}$ and $\Delta\mathcal{R}$), which means that we don't need to adopt a pairwise cost. The following energy function then needs to be minimized over labellings $\mathbf{z}$:

$$f_D(\mathbf{z}) = \gamma_5 \cdot \sum_m \theta_D(m,d_m), \tag{7}$$

$$\text{where:} \quad \theta_D(m,d_m) = \sum_{\delta x=-4}^{4}\sum_{\delta y=-4}^{4}\big(|\mathcal{L}(x+\delta x,y+\delta y)-\mathcal{R}(x+\delta x-d_m,y+\delta y)| + |\Delta\mathcal{L}(x+\delta x,y+\delta y)-\Delta\mathcal{R}(x+\delta x-d_m,y+\delta y)|\big). \quad (8)$$

2.4 Jointly estimating pose and segmentation

Here, we encode the concept that foreground pixels should be *explained* by some body part; conversely, each selected body part should explain some part of the foreground. We use the same weights w_{ij}^m as defined in Section 2.1, and calculate two terms: cost J_1 is added if the part candidate (i,j) is selected and the pixel m is labelled as background; secondly, cost J_2 is added if a pixel m is assigned to foreground, but not explained by any body part. We set a threshold $\tau = 0.1$ (value determined empirically), and a cost is accrued for m when for all parts (i,j), $w_{ij}^m < \tau$. The overall cost f_{PS} for a particular labelling $\mathbf{z}$ can be written as:

$$f_{PS}(\mathbf{z}) = \gamma_6 \cdot J_1(\mathbf{z}) + \gamma_7 \cdot J_2(\mathbf{z}), \quad (9)$$

$$\text{where: } J_1(\mathbf{z}) = \sum_{i,j}\sum_{m}\left(\mathbf{1}(b_i = j)\cdot(1-s_m)\cdot w_{ij}^m\right); \quad (10)$$

$$J_2(\mathbf{z}) = \sum_{m}\mathbf{1}(\max_{i,j} w_{ij}^m < \tau)\cdot s_m. \quad (11)$$

2.5 Jointly estimating segmentation and stereo

Here, we encode the idea that, assuming that the objects closest to the camera are body parts, foreground pixels should have a higher disparity than background pixels. To do this, we use the foreground weights W_F obtained in Section 2.1 to obtain an expected value E_F for the foreground disparity:

$$E_F = \frac{\sum_m w_m \cdot d_m}{\sum_m w_m} \quad (12)$$

Using a hinge loss with a non-negative slack variable $\xi = 2$ to allow small deviations to occur, we then have the following cost measure to penalize pixels with high disparity being assigned to the background:

$$f_{SD}(\mathbf{z}) = \gamma_8 \cdot \sum_m (1-s_m)\cdot\max(d_m - E_F - \xi, 0). \quad (13)$$

3 Dual decomposition

3.1 Binarizing variables

Many of the minimization problems defined in Section 2 are multiclass problems, and are therefore NP-hard to solve in their current forms [5]. However, we can binarize the multiclass label sets $\mathcal{D}$ and $\mathcal{B}$. For pixels, we extend the labelling space so that each pixel takes a vector of binary labels $z_m = [d_{(m,0)}, d_{(m,1)}, \cdots, d_{(m,K-1)}, s_m]$, with each $d_{(m,k)}$ equal to 1 if and only if disparity value k is selected for pixel m. For parts, we extend the labelling space so that each part takes a vector of binary labels $b_i = [b_{(i,0)}, b_{(i,1)}, \cdots, b_{(i,N_E-1)}]$, where each $b_{(i,j)}$ is equal to 1 if and only if the j^{th} proposal for part i is selected.

A particular solution to this binary labelling problem is denoted by $\bar{\mathbf{z}}$. Only a subset of the possible binary labellings will correspond directly to multiclass labellings $\mathbf{z}$; these are

those such that each pixel has exactly one disparity turned on, and each part has exactly one proposal selected. The set of solutions for which all pixels satisfy this constraint is called the *feasible set* F. We can write:

$$F = \left\{ \bar{\mathbf{z}} : \sum_{k=0}^{K-1} d_{(m,k)} = 1 \ \forall Z_m \in \mathbf{Z}; \sum_{j=0}^{N_E-1} b_{(i,j)} = 1 \ \forall B_i \in \mathbf{B} \right\}. \tag{14}$$

We rewrite the cost functions from (5) and (7) as follows:

$$f'_P(\bar{\mathbf{z}}) = \gamma_3 \cdot \sum_{(i,j)} b_{(i,j)} \cdot \theta_P(i,j) + \gamma_4 \cdot \sum_{(i_1,i_2)\in\mathcal{T}} \sum_{j_1,j_2} b_{(i_1,j_1)} \cdot b_{(i_2,j_2)} \cdot \phi_{i_1,i_2}(j_1,j_2); \tag{15}$$

$$f'_D(\bar{\mathbf{z}}) = \gamma_5 \cdot \sum_m \sum_k d_{(m,k)} \cdot \theta_D(m,k). \tag{16}$$

The joining functions given in Sections 2.4 and 2.5 can be binarized in a similar fashion (the details are omitted due to space constraints). The energy minimization problem in (1) can be restated in terms of these binary functions, giving us:

$$\begin{aligned} E(\bar{\mathbf{z}}) = f'_{\mathcal{D}}(\bar{\mathbf{z}}) + f_S(\bar{\mathbf{z}}) + f'_{\mathcal{P}}(\bar{\mathbf{z}}) + f'_{PS}(\bar{\mathbf{z}}) + f'_{SD}(\bar{\mathbf{z}}) \\ \text{subject to: } \bar{\mathbf{z}} \in F. \end{aligned} \tag{17}$$

3.2 Optimization

Minimizing this energy function across all labellings $\bar{\mathbf{z}}$ simultaneously is NP-hard [5], so in order to simplify the problem, we use dual decomposition. A brief explanation is given here; the interested reader is directed to [4] for an excellent tutorial.

We introduce duplicate variables $\bar{\mathbf{z}}_1$ and $\bar{\mathbf{z}}_2$, and only enforce the feasibility constraints on these duplicates. Our energy function thus becomes:

$$\begin{aligned} E(\bar{\mathbf{z}},\bar{\mathbf{z}}_1,\bar{\mathbf{z}}_2) = & f'_{\mathcal{D}}(\bar{\mathbf{z}}_1) + f_S(\bar{\mathbf{z}}) + f'_{\mathcal{P}}(\bar{\mathbf{z}}_2) + f'_{PS}(\bar{\mathbf{z}}) + f'_{SD}(\bar{\mathbf{z}}) \\ \text{subject to: } & \bar{\mathbf{z}}_1,\bar{\mathbf{z}}_2 \in F,\ \bar{\mathbf{z}}_1 = \bar{\mathbf{z}},\ \bar{\mathbf{z}}_2 = \bar{\mathbf{z}}. \end{aligned} \tag{18}$$

We remove the equality constraints via adding Lagrangian multipliers, and decompose this dual problem into three subproblems L_1, L_2 and L_3, as follows:

$$\begin{aligned} L(\bar{\mathbf{z}},\bar{\mathbf{z}}_1,\bar{\mathbf{z}}_2) &= f'_{\mathcal{D}}(\bar{\mathbf{z}}_1) + f_S(\bar{\mathbf{z}}) + f'_{\mathcal{P}}(\bar{\mathbf{z}}_2) + f'_{PS}(\bar{\mathbf{z}}) && (19) \\ &\quad + f'_{SD}(\bar{\mathbf{z}}) + \lambda_D(\bar{\mathbf{z}} - \bar{\mathbf{z}}_1) + \lambda_P(\bar{\mathbf{z}} - \bar{\mathbf{z}}_2) \\ &= L_1(\bar{\mathbf{z}}_1,\lambda_D) + L_2(\bar{\mathbf{z}}_2,\lambda_P) + L_3(\bar{\mathbf{z}},\lambda_D,\lambda_P), && (20) \\ \text{where:} \quad L_1(\bar{\mathbf{z}}_1,\lambda_D) &= f'_D(\bar{\mathbf{z}}_1) - \lambda_D\bar{\mathbf{z}}_1; && (21) \\ L_2(\bar{\mathbf{z}}_2,\lambda_P) &= f'_P(\bar{\mathbf{z}}_2) - \lambda_P\bar{\mathbf{z}}_2; && (22) \\ L_3(\bar{\mathbf{z}},\lambda_D,\lambda_P) &= f_S(\bar{\mathbf{z}}) + f'_{SD}(\bar{\mathbf{z}}) + f'_{PS}(\bar{\mathbf{z}}) + \lambda_D\bar{\mathbf{z}} + \lambda_P\bar{\mathbf{z}}. && (23) \end{aligned}$$

are the three slave problems, which can be optimized independently and efficiently, while treating the dual variables λ_D and λ_P as constant. This process is shown graphically in Figure 2. Intuitively, the role of the dual variables is to encourage the labellings $\bar{\mathbf{z}}$, $\bar{\mathbf{z}}_1$, and $\bar{\mathbf{z}}_2$ to agree with each other.

Given the current values of λ_D and λ_P, we solve the slave problems L_1, L_2 and L_3, denoting the solutions by $\bar{L}_1(\lambda_D)$, $\bar{L}_2(\lambda_P)$ and $\bar{L}_3(\lambda_D,\lambda_P)$ respectively. We concatenate $\bar{L}_1(\lambda_D)$

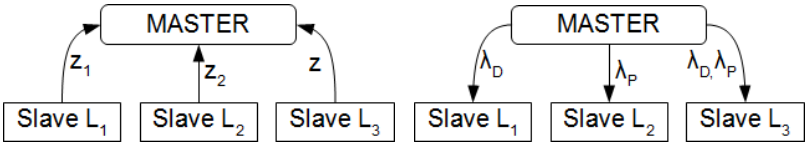


Figure 2: Diagram showing the two-stage update process. **(a):** the slaves find labellings $\mathbf{z}$, $\mathbf{z}_1$, $\mathbf{z}_2$ and pass them to the master; **(b):** the master updates the dual variables λ_D and λ_P and passes them to the slaves.

and $\bar{L}_2(\lambda_P)$ to form a vector of the same dimensionality as $\bar{L}_3(\lambda_D,\lambda_P)$. The master then calculates the subgradient of the relaxed dual function at (λ_D,λ_P), given by:

$$\nabla L(\lambda_D,\lambda_P) = \bar{L}_3(\lambda_D,\lambda_P) - [\bar{L}_1(\lambda_D),\bar{L}_2(\lambda_P)]. \tag{24}$$

The master problem can then update the dual variables using the subgradient method, similar to that of [24], and then update the λ by adding $\alpha_t \nabla L(\lambda_D,\lambda_P)$ to them, and then passing the resulting λ vectors back to the slaves. Here, α_t is the step size indexed by iteration t, which we adaptively set as detailed in [17]. The α_t form a decreasing sequence, so we make progressively finer refinements with each iteration. The costs of variables for which the slave problems disagree are altered in order to encourage the solutions to match.

3.3 Solving sub-problem L_1

Since problem L_1 contains terms that only depend on the disparity variables, we can relax the feasibility constraint in (14) to only depend on these variables. The feasible set is the set of $\bar{\mathbf{z}}$ such that for all pixels Z_m, $\sum_{k=0}^{K-1} d_{(m,k)} = 1$. We call this expanded feasible set F_D. Then, we can write L_1 in terms of the binary function f'_D as in (25) below. Since f'_D includes only unary terms, this equation can be solved independently for each pixel.

$$\begin{aligned} L_1(\bar{\mathbf{z}}_1,\lambda_D) =& f'_D(\bar{\mathbf{z}}_1) - \lambda_D \bar{\mathbf{z}}_1 \\ \text{subject to: } & \bar{\mathbf{z}}_1 \in F_D. \end{aligned} \tag{25}$$

3.4 Solving sub-problem L_2

L_2 contains functions that depend only on the pose variables b_{ij}, so we can again relax the feasibility constraint. This expanded feasible set is the set of $\bar{\mathbf{z}}$ such that for all parts B_i, $\sum_{j=0}^{N_E-1} b_{(i,j)} = 1$. Denoting this set F_P, L_2 can be written in terms of f'_P as follows:

$$\begin{aligned} L_2(\bar{\mathbf{z}}_2,\lambda_P) =& f'_P(\bar{\mathbf{z}}_2) - \lambda_P \bar{\mathbf{z}}_2 \\ \text{subject to: } & \bar{\mathbf{z}}_2 \in F_P, \end{aligned} \tag{26}$$

with f'_P as in (15). Ordering the parts B_i such that $(i_1,i_2) \in \mathcal{T}$ only if $i_1 < i_2$, we find the optimal solution via a bottom-up process based on the Viterbi algorithm [23]. The score of each leaf node is the following, calculated for each estimate j:

$$\text{score}_i(j) = \theta_P(i,j) - \lambda_P(i,j). \tag{27}$$

For a node i with children, we can compute the following:

$$\text{score}_i(j) = \theta_P(i,j) - \lambda_P(i,j) + \sum_{(i_1,i)\in\mathcal{T}} \min_{j_1} \left(\phi_{i_1,i}(j_1,j) + \text{score}_{i_1}(j_1)\right), \tag{28}$$

and the globally optimal solution is found by keeping track of the arg min indices, and then selecting the root (torso) estimate with minimal score.

3.5 Solving sub-problem L_3

Sub-problem L_3 is significantly more complex, as it includes the joining terms f'_{PS} and f'_{SD}. Since we have rewritten L_1 and L_2 in terms of the binary variables $\bar{\mathbf{z}}_1$ and $\bar{\mathbf{z}}_2$, we need to do the same to the joining terms. $J_1(\mathbf{z})$ penalized background pixels being assigned a body part, while $J_2(\mathbf{z})$ penalized foreground pixels not being explained by any body part. Together, these form $f_{PS}(\mathbf{z})$, as in (9). For parts i, estimates j, and pixels m, this becomes:

$$f'_{PS}(\bar{\mathbf{z}}) = \gamma_6 \cdot J'_1(\bar{\mathbf{z}}) + \gamma_7 \cdot J'_2(\bar{\mathbf{z}}), \tag{29}$$

where:

$$J'_1(\bar{\mathbf{z}}) = \sum_{i,j}\sum_{m} \left(b_{(i,j)} \cdot (1-s_m) \cdot w_{ij}^m\right); \tag{30}$$

$$J'_2(\bar{\mathbf{z}}) = \sum_{m} \mathbf{1}\left(\tau - \max_{i,j} w_{ij}^m > 0\right) \cdot s_m. \tag{31}$$

Function $f_{SD}(\bar{\mathbf{z}})$, which penalizes background pixels with a higher disparity than the foreground region, becomes the following for pixels m and disparities k:

$$f'_{SD}(\bar{\mathbf{z}}) = \gamma_8 \cdot \sum_{m}\sum_{k} \left((1-s_m) \cdot d_{(m,k)} \cdot \max(j - E_F - \xi, 0)\right). \tag{32}$$

Since all the terms in the energy function L_3 are submodular, the optimization problem is convex, and can be efficiently minimized via graph cuts.

4 Results

Experimental Setup: For each experiment, the training set is used for two tasks: to train the pose estimation algorithm we use to generate proposals, and to learn the weights γ_i that we attached to the energy terms. These weights are learned by coordinate ascent.

Datasets: While there are plenty of datasets in the vision community for evaluating pose estimation and stereo algorithms, we are not currently aware of any datasets for evaluating 3D pose estimation algorithms that run on stereo images. Therefore, we present a new dataset, which we call Humans in Two Views (H2view for short).

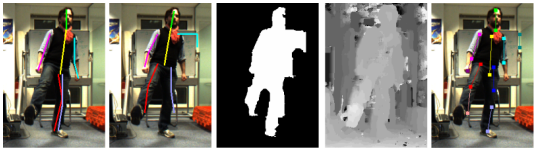
The dataset, which is publicly available[1], consists of 8,741 images of humans standing, walking, crouching or gesticulating in front of a stereo camera, divided up into 25 video sequences, with eight subjects and three locations. The dataset is fully annotated, with left and right RGB images available, plus ground-truth depth, segmentation and pose information obtained via a Microsoft Kinect, and corrected manually. The training set contains 7,143 images, while the test set features 1,598 images, with a different location and different subjects from the training set.

[1] http://cms.brookes.ac.uk/research/visiongroup/h2view/

Method	Torso	Head	Upper arm	Forearm	Upper leg	Lower leg	Total
Ours	**94.9**	**88.7**	**74.4**	**43.4**	88.4	78.8	**75.37**
Yang[25]	72.0	87.3	61.5	36.6	**88.5**	**83.0**	69.85
Andriluka [2]	80.5	69.2	60.2	35.2	83.9	76.0	66.03

Table 1: Results (given in % PCP) on the H2view test sequence.

(a) Pose, segmentation and stereo results together.



(b) Error correction: the first estimate (first image) misclassifies the left leg (red), while the second estimate (second image) gets it right; our segmentation (third image) and stereo (fourth image) cues enable us to recover (fifth image).

Figure 3: Some sample results from our new dataset.

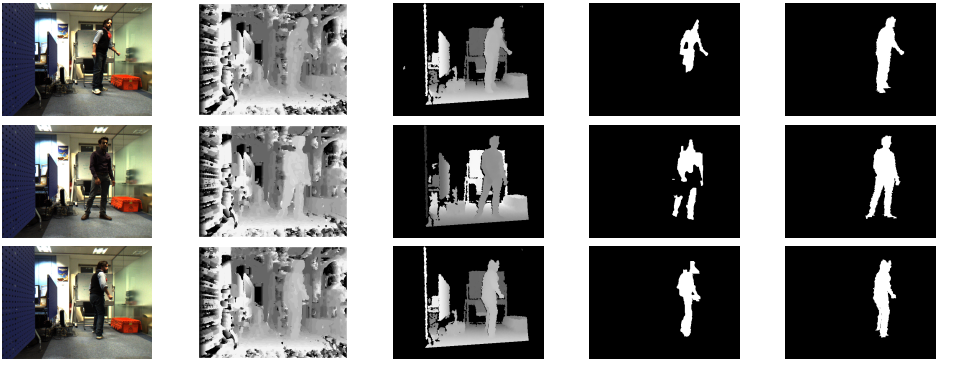
Performance: To evaluate pose estimation, we follow the standard criteria of probability of correct pose (PCP) [10] to measure the percentage of correctly localized body parts. Quantitative results are given in Table 1, while we include qualitative results in Figure 3.

Our model exhibits a significant performance increase for the upper body, where the segmentation cues are the strongest. However, there is a slight reduction in performance for the upper and lower legs. Our joint inference model improves on the performance of Yang and Ramanan's algorithm by 5.52%; an example where our formulation has corrected a mistake is shown in Figure 3(b). Some qualitative stereo and segmentation results are given in Figure 4. Due to the difference between the fields of view of the stereo and Kinect cameras, ground truth disparity values are only available for the foreground objects, some background objects, and surrounding floor space. Comparing our stereo results (Figure 4(b)) to the ground truth disparity values for these objects (Figure 4(c)) shows a good correspondence between the two. On the negative side, the segmentation frequently omits pixels from the legs (Figure 4(d)). This is perhaps because the foreground weights are weaker in that area, as all of the top ten detections returned by Yang and Ramanan's algorithm contribute to the foreground weight map, and the accuracy of leg detections drops off sharply after the top detection.

Runtime: Our algorithm requires around 15 seconds per frame, using a single 2.67GHz processor. This is similar to the observed runtime of Yang and Ramanan (about 10 seconds per frame), which only solves pose estimation, and is much quicker than the implementation provided in Andriluka *et al*. [2], which requires around 3 minutes per frame.

5 Conclusions

In this paper, we have described a novel formulation for solving the problems of human segmentation, pose estimation and depth estimation, using a single energy function. The algorithm we have presented is self-contained, and performs very well in the pose and depth estimation tasks; however, there is considerable room for improvement in the segmentation results. Additionally, we have introduced an extensive, fully annotated dataset for 3D human pose estimation.



(a) (b) (c) (d) (e)

Figure 4: Sample stereo and segmentation results. **(a):** RGB image; **(b):** disparity map; **(c):** ground truth depth; **(d):** segmentation result; **(e):** ground truth segmentation. The Kinect depth data used to generate the ground truth depth in (c) is only available for some pixels, due to the slightly different field of view of the camera.

The algorithm is modular in design, which means that it would be straightforward to substitute alternative approaches for each slave problem; a thorough survey of the efficacy of these combinations would be a promising direction for future research. There is also scope for improvement in our runtime by sharing operations across CPU cores, for instance by running the slave algorithms in parallel.

References

[1] Xbox kinect. full body game controller from microsoft. *http://www. xbox. com/kinect*.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021. IEEE, 2009.

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1365–1372, 2009.

[4] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods. *Notes for EE364B, Stanford University*, 2007.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[6] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. *European Conference on Computer Vision (ECCV)*, pages 642–655, 2006.

[7] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P.H.S. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110, 2007.

[8] M. Enzweiler, A. Eigenstetter, B. Schiele, and D.M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 990–997, 2010.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[10] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[11] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 407–414, 2005.

[12] N. Komodakis, N. Paragios, and G. Tziritas. Mrf energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (99):1–1, 2011.

[13] M.P. Kumar, A. Zisserman, and P.H.S. Torr. Efficient discriminative learning of parts-based models. In *IEEE International Conference on Computer Vision (ICCV)*, pages 552–559, 2009.

[14] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):530–545, 2010.

[15] Y. Matsumoto and A. Zelinsky. An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In *Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 499–504, 2000.

[16] W. Niu, J. Long, D. Han, and Y.F. Wang. Human activity detection and recognition for video surveillance. In *IEEE International Conference on Multimedia and Exp (ICME)*, volume 1, pages 719–722, 2004.

[17] B. Packer, S. Gould, and D. Koller. A unified contour-pixel model for figure-ground segmentation. *European Conference on Computer Vision (ECCV)*, pages 338–351, 2010.

[18] S. Pellegrini and L. Iocchi. Human posture tracking and classification through stereo vision and 3d model matching. *Journal on Image and Video Processing*, 2008:1–12, 2008.

[19] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314, 2004.

[21] C. Rother, V. Kolmogorov, Y. Boykov, and A. Blake. Interactive foreground extraction using graph cut. *Advances in Markov Random Fields for Vision and Image Processing*, 2011.

[22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

[23] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[24] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2433–2440, 2011.

[25] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392, 2011.