A Robust Stereo Prior for Human Segmentation

Glenn Sheasby, Julien Valentin, Nigel Crook, and Philip Torr

Oxford Brookes University

Abstract. The emergence of affordable depth cameras has enabled significant advances in human segmentation and pose estimation in recent years. While it leads to impressive results in many tasks, the use of infra-red cameras have their drawbacks, in particular the fact that they don't work in direct sunlight. One alternative is to use a stereo pair of cameras to produce a disparity space image. In this work, we propose a robust method of using a disparity space image to create a prior for human segmentation. This new prior leads to greatly improved segmentation results; it can be applied to any task where a stereo pair of cameras is available, and segmentation results are desired. As an application, we show how the prior can be inserted into a dual decomposition formulation for stereo, segmentation and human pose estimation.

1 Introduction

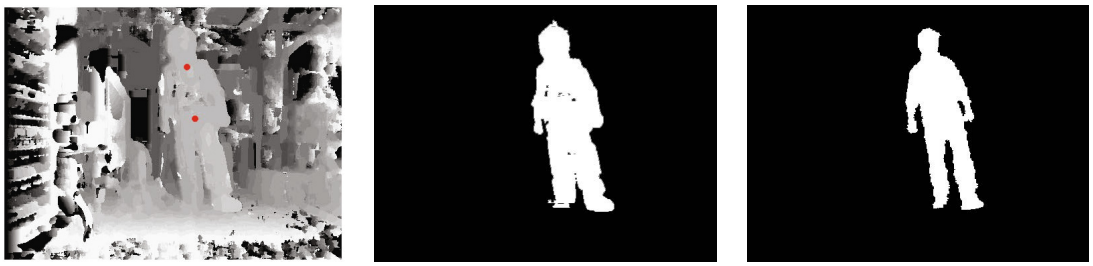
Over the past few years, depth cameras have emerged as a valuable tool to aid segmentation and pose estimation, especially of humans. The Microsoft Kinect [13] is one such camera, which uses an infra-red sensor to produce a 16-bit depth image. Given this depth image, the tasks of segmenting and estimating the pose of the human body become easier. Shotton et al. [18] use segmentation as an initial step towards pose estimation. The depth values within the segmented body shape are used to build a set of simple features, which are then used as weak classifiers. The skeleton classification is performed using a randomized forest of decision trees [4].

One drawback of using infra-red sensors is that they fail in outdoor scenes due to IR interference from sunlight, and therefore can not be used for pedestrian detection. An alternative approach, which we follow in this work, is to use a stereo pair of cameras to build the depth image.

In this paper, we show how a high-quality segmentation result can be obtained directly from the disparity map. In object segmentation, the goal is to obtain a binary labelling of an image, showing which pixels belong to the object of interest. This object will generally occupy a continuous region of the scene, and the depth of the object will be smooth within this region. In contrast, the background region tends to have a different depth.

This property motivates our method of generating a segmentation prior: after finding a small set of seed pixels which have high probability of belonging to the foreground object, we use a flood fill algorithm to find a continuous region of the image which has a smooth disparity.

K.M. Lee et al. (Eds.): ACCV 2012, Part II, LNCS 7725, pp. 94–107, 2013.



(a) (b) (c)

Fig. 1. The seeds (in red) on the disparity map in **(a)** are used to generate the flood fill prior in **(b)**, which is a good approximation of the ground truth segmentation **(c)**.

As shown in Fig. 1, a flood fill prior can generate promising human segmentations by itself. This result can be improved further by unifying the tasks of stereo, segmentation and human pose estimation. For this, we use dual decomposition, building on the framework of [17] to include range moves for stereo [9], and the flood fill prior for segmentation.

The main contributions of this paper are twofold: first, we demonstrate how a disparity map can be used to form a robust prior for segmentation. Secondly, we show how this prior can be applied to greatly improve the segmentation results of our previous work [17]. The paper is structured as follows: we summarize related work in the next section, describing the range expansion formulation. Then, the flood fill prior is introduced in Section 3. The framework of [17], and our modifications to it, are shown in Section 4. Experimental results follow in Section 5, and concluding remarks are given in Section 6.

2 Related Work

2.1 Segmentation Priors

The use of priors for segmentation has made a significant difference for a variety of computer vision problems in recent years. Many of these approaches can be traced back to the introduction of GrabCut [15], an interactive method in which the user specifies a rectangular region of interest, and then the foreground region is refined based on that region using graph cuts. Larlus and Jurie [12] used a similar approach, using object detection to supply bounding boxes and thus automate the process. More sophisticated priors have been used in works such as `ObjCut` and `PoseCut`, which use the output of object detection and pose estimation algorithms respectively to provide a shape prior term [3,8].

Lower-level priors are used by Gulshan et al. [6], who formulate the task of segmenting humans as a learning problem, using linear classifiers to predict segmentation masks from HOG descriptors. Ladický et al. [11] showed how multiple such features and priors can be combined, avoiding the need to make an *a priori* decision of which is most appropriate. Their Associative Hierarchical Random

Field (AHRF) model has demonstrated state of the art results for several semantic segmentation problems.

In this paper, we use the intuition that foreground objects will occupy a continuous region of 3D space, and will therefore exist within a smooth range of depth values. Given the bijective relation between depth and disparity, we show how disparity can be used as a discriminative tool to aid segmentation, and propose a novel segmentation prior based on the disparity map produced by a stereo correspondence algorithm.

2.2 Stereo Correspondence Algorithms

A plethora of stereo correspondence algorithms have been developed over the years; an excellent review of earlier methods was given in Scharstein and Szeliski [16]. Kolmogorov et al. [7] propose adaptations of dynamic programming and graph cuts in order to solve a simplified form of the stereo correspondence problem. They are interested in real-time foreground-background segmentation of video sequences, and achieve decent results without any explicit notion of temporal consistency, so their algorithm is not susceptible to drift-related problems.

In a similar vein, Criminisi et al. [5] use a four-state dynamic programming approach to eliminate some of the more visible errors in new view synthesis, which can suffer from problems like "halo" artifacts, where a narrow area surrounding an object is mistakenly classified as foreground. However, their approach relies on background models that assume a quasi-static background, which we can not count on in the general case.

In this paper, we follow the approach of Kumar et al. [9], who proposed an efficient way to solve the stereo correspondence problem using graph cuts. They note that the disparity labels form a discrete range of possible values, and that therefore the stereo correspondence problem is suitable for the application of range moves. A *range move* is a move making approach where rather than only considering one or two labels at each iteration, a range of several possible values are considered. Kumar et al. define two different range move formulations; the one we use in this paper is *range expansion*. In a single range expansion move, each pixel can either keep its old label, or choose from a range of consecutive labels – in this case, disparity values.

2.3 Range Moves

Given a stereo pair of images $\mathcal{L}$ and $\mathcal{R}$, our aim is to solve the stereo correspondence problem by finding a disparity image. In order to solve this problem via graph cuts, we construct a set of pixels $P = \{p_1, p_2, \ldots, p_N\}$, and a set of disparity labels $\mathcal{D} = \{0, 1, \ldots, M-1\}$. A solution obtained via graph cuts is referred to as a labelling $\mathbf{z}$, where each pixel variable p_i is assigned a label $d(p_i) \in \mathcal{D}$. The set of possible labellings is referred to as the *label space*.

Move-making algorithms find a local minimum solution to a graph cut problem by making a series of "moves" in the label space, where each move is governed by a set of specific rules. One of the more popular moves is α-expansion [2], where each

(a) (b) (c)

Fig. 2. The results of three successive range expansion iterations. At first, there are very few disparities to choose from, but with further iterations, the person becomes more visible against the background.

variable can either retain its current label, or change its label to α. When the set of labels is ordered (as $\mathcal{D}$ is), it is possible to efficiently obtain a local minimum by considering a range of labels rather than just one. This can be done by using the range expansion algorithm, as proposed by Kumar et al. [9].

At each iteration m of the range expansion algorithm, let $\mathbf{z}_m$ be the current labelling of pixels; each pixel p_i has a disparity label $d_m(p_i)$. We consider an interval I_m of labels, where $I_m = \{\alpha_m, \alpha_m + 1, \ldots, \beta_m\}$ The option is provided for each pixel to either keep its current label $d_{m+1}(p_i) = d_m(p_i)$, or choose a new label $d_{m+1}(p_i) \in I_m$. Formally, we find a labelling $\mathbf{z}_{m+1}$ satisfying:

$$\mathbf{z}_{m+1} = \arg\min_{\mathbf{z}} f_D(\mathbf{z}) \tag{1}$$
$$\text{such that } \forall\ p_i \in P : d_{m+1}(p_i) = d_m(p_i) \text{ OR } d_{m+1}(p_i) \in I_m$$

where f_D is the energy function of the stereo correspondence problem, with unary potentials θ_D, pairwise potentials ϕ_D, and weight parameters γ_1 and γ_2:

$$f_D(\mathbf{z}) = \gamma_1 \sum_{p_i} \theta_D(p_i, d_m(p_i)) + \gamma_2 \sum_{(p_i,p_j)\in C} \phi_{(p_i,p_j)}(d_m(p_i), d_m(p_j)) \tag{2}$$

where C is the set of neighboring pairs of pixels in P, and ϕ is the L_2 norm on the depth difference.

To minimize this energy function, we follow the formulation of [9], using $\gamma_1 \cdot \theta_D$ for the unary potentials, and $\gamma_2 \cdot \phi_{(p_i,p_j)}$ for the pairwise potentials (full details are not included here due to space constraints, but can be found in the supplementary material). An example of a sequence of range move iterations is shown in Fig. 2.

3 Flood Fill Prior

After using range expansions to find the disparity map, we apply a flood fill to generate a prior for human segmentation. Flood fill is an algorithm that, given

Algorithm 1. Generic flood fill algorithm for an image I of size $W \times H$.

Input: non-empty list S of seed pixels
initialize a zero-valued matrix F of size $W \times H$; empty queue of new ranges R
for all seeds $s = (s_x, s_y) \in S$ **do**
  $(F, x_{\min}, x_{\max}) \leftarrow$ `doLinearFill`(F, s_x, s_y)
  if $x_{\min} < x_{\max}$ **then**
    add $(x_{\min}, x_{\max}, y)$ to back of queue R
  end if
  while $R \neq \emptyset$ **do**
    remove $(x_{\min}, x_{\max}, y)$ from front of R
    if $y > 0$ **then**
      $(R, F) \leftarrow$ `multiLinearFill`$(F, x_{\min}, x_{\max}, y-1)$
    end if
    if $y < H - 1$ **then**
      $(R, F) \leftarrow$ `multiLinearFill`$(F, x_{\min}, x_{\max}, y+1)$
    end if
  end while
end for
return F

a list of seeds, fills connected parts of a multidimensional array, e.g. an image, with a given label. Starting with two pixels which have a high probability of belonging to the human, we find a continuous region of the image which has a smooth disparity. This region is specified by a binary image F.

The flood fill algorithm is a standard method that has many applications: for example, it can be used for bucket filling in graphics editors such as GIMP or Microsoft Paint; a memory-efficient version is given in general form in Algorithm 1. The function `doLinearFill`(F, s_x, s_y) finds the largest range $(x_{\min}, x_{\max})$, where $x_{\min} \leq s_x \leq x_{\max}$, such that for all x, $x+1$ within the range, $|I(x, s_y) - I(x+1, s_y)| < \epsilon$. In other words, it finds the largest smooth horizontal region of the image containing (s_x, s_y).

The function `multiLinearFill`$(F, x_{\min}, x_{\max}, y)$ executes `doLinearFill` starting from each pixel (x, y) where $x_{\min} \leq x \leq x_{\max}$, provided $F(x, y) = 0$, i.e. (x, y) has not already been filled by the algorithm. Each execution of `doLinearFill` generates a new range $(x'_{\min}, x'_{\max})$; if $x'_{\min} < x'_{\max}$, then $(x'_{\min}, x'_{\max}, y)$ is added to R. Pseudocode for the algorithm is given in the supplementary material.

We use the flood fill algorithm to generate a segmentation of the human from the disparity map. We run the pose estimation algorithm of Yang and Ramanan [19] on the original RGB image, and then use the endpoints of the top-ranked torso estimate as seeds.

To determine whether a value $I(x, y)$ matches its neighbor $I(x + \delta x, y + \delta y)$ (where $|\delta x| + |\delta y| = 1$), we consider the difference between the two values; the pair of values match if the difference is below a preset threshold.

If the person is fully visible, then we will also be able to see the point where their feet touch the floor. At this point, the disparity value of the floor is often very similar to the disparity value of the foot, and so it is possible for the

segmentation prior to 'leak' on to the floor. In order to prevent this, we estimate the position of the floor plane.

The disparity d of a pixel p_i has an inverse relation to its depth z,

$$z(p_i) = \frac{b(f_x + f_y)}{2d(p_i)} \tag{3}$$

where b is the baseline (the distance between the camera centres) of the stereo camera, and f_x and f_y are the focal lengths in x and y respectively (diagonal elements of the camera matrix).

Once we have the depth z, the real-world height can be obtained via a projective transformation. We also know that the camera is parallel to the floor plane, and we know its height above the ground at capture time, so by thresholding the height values, we can obtain an estimate for the floor plane. A typical flood fill prior is shown in Fig. 1.

4 Application: Human Segmentation

Here, we discuss the framework of our previous work [17]. The objective is to minimize an energy function which consists of five terms: three representing the distinct problems that the function unifies (i.e. stereo correspondence, segmentation, and pose estimation), and two *joining terms*, which encourage consistency between the solutions of these problems. As input, we have a stereo pair of images $\mathcal{L}$ and $\mathcal{R}$. The parts-based pose estimation algorithm of Yang and Ramanan [19] is used as a preprocessing step to obtain a number N_E of proposals for $K = 10$ body parts: head, torso, and two for each of the four limbs. For each part i, each proposal j consists of a pair of image co-ordinates, representing the endpoints of the limb (or skull, or spine).

4.1 Original Formulation

The approach is formulated as a conditional random field (`CRF`), with two sets of random variables. The set $P = \{p_1, p_2, \ldots, p_N\}$ represents the image pixels; in addition to a disparity label from the multi-class label set $\mathcal{D}$ defined in Section 2.3, each pixel p_i is given a binary segmentation label $s(p_i)$ from $\mathcal{S} = \{0, 1\}$. The set $B = \{B_1, B_2, \ldots, B_K\}$ represents the body parts; labels are assigned to each part from the multi-class label set $\mathcal{B}$. Any possible assignment of labels to these variables is called a *labelling*, and denoted by $\mathbf{z}$. The energy of a labelling is written as:

$$E(\mathbf{z}) = f_D(\mathbf{z}) + f_S(\mathbf{z}) + f_P(\mathbf{z}) + f_{PS}(\mathbf{z}) + f_{SD}(\mathbf{z}) \tag{4}$$

with each term containing weights $\gamma_i \in \mathbb{R}^+$.

In order to utilize range moves and apply the new flood fill prior, we need to make some amendments to the formulation given in [17]. We modify three of the terms: f_D, which gives the cost of the disparity label assignment $\{d(p_i)\}_{i=1}^N$; f_S,

which gives the cost of the segmentation label assignment $\{s(p_i)\}_{i=1}^{N}$; and f_{SD}, which unifies the two. We now describe in turn each of the terms in Eq. (4), along with our modifications to these terms. For clarity, the terms as they appeared in [17] will be denoted f_*, while our new functions will be denoted g_*. Weights carried over from the original formulation will be denoted γ_* (with the same index), while new weights will be denoted w_*.

4.2 Stereo Term f_D

$f_D(\mathbf{z})$ represented the energy of the disparity map. In [17], the energy was simply given in terms of a cost volume θ_D, which for each pixel p_i, specified the cost of assigning a disparity label $d(p_i)$. This cost incorporated gradient in the x-direction, so a pairwise term was not added:

$$f_D(\mathbf{z}) = \gamma_1 \sum_{p_i} \theta_D(p_i, d(p_i)) \tag{5}$$

In order to apply range moves, we introduce a truncated pairwise term to formalize the notion that adjacent pixels should have similar disparities. After the k^{th} iteration, denote by $d_k(p_i)$ the labelling of pixel p_i. We define C to be the set of pairs of neighboring pixels (p_i, p_j). Given a labelling $\mathbf{z}$, the pairwise cost associated with a pair of pixels $(p_i, p_j) \in C$ is:

$$\phi_{(p_i,p_j)}(d_k(p_i), d_k(p_j)) = |d_k(p_i) - d_k(p_j)| \tag{6}$$

The overall energy of the labelling is:

$$g_D(\mathbf{z}) = \gamma_1 \sum_{p_i} \theta_D(p_i, d_k(p_i)) + w_1 \sum_{(p_i,p_j)\in C} \phi_{(p_i,p_j)}(d_k(p_i), d_k(p_j)) \tag{7}$$

4.3 Segmentation Terms f_S and f_{SD}

$f_S(\mathbf{z})$ gave the segmentation energy. The part estimates obtained from Yang and Ramanan's algorithm were used to create a foreground weight map W_F, from which Gaussian Mixture Models based on RGB values were fitted for the foreground and background respectively, built using the Orchard-Bouman clustering algorithm [14]. These were used to obtain unary costs θ_F and θ_B of assigning each pixel to foreground and background respectively. Additionally, a pairwise cost ϕ_S was included to penalize the case where adjacent pixels are assigned to different labels:

$$f_S(\mathbf{z}) = \gamma_2 \cdot \theta_S(\mathbf{z}) + \gamma_3 \cdot \phi_S(\mathbf{z}) \tag{8}$$

where:
$$\theta_S(\mathbf{z}) = \sum_{p_i \in P} s(p_i) \cdot \theta_F(p_i) + (1 - s(p_i)) \cdot \theta_B(p_i) \tag{9}$$

$$\phi_S(\mathbf{z}) = \sum_{(p_i,p_j)\in C} \mathbf{1}(s(p_i) \neq s(p_j)) \exp(-\beta \left\|\mathcal{L}(p_i) - \mathcal{L}(p_j)\right\|^2) \tag{10}$$

$f_{SD}(\mathbf{z})$ was a cost function that penalized the case where pixels with high disparity were assigned to foreground, contrary to the notion that the foreground object is in front of the background. Using the foreground weight map W_F defined above, a set $\mathcal{F}$ of pixels with a high probability of being foreground was obtained, from which we the expected foreground disparity E_F was calculated. Background pixels with disparity greater than $E_F + \xi$, where ξ is a non-negative slack variable, were then penalized:

$$f_{SD}(\mathbf{z}) = \gamma_4 \sum_{p_i \in P} (1 - s(p_i)) \cdot \max(d(p_i) - E_F - \xi, 0) \tag{11}$$

We find that a flood fill prior proves to be much more reliable in discriminating between high-disparity pixels that belong to the object, and those far from the object. Therefore, we replace f_{SD} with a flood fill-based term. Define $F = \{\rho_k(p_i) : p_i \in P\}$ to be the flood fill prior at iteration k. We can penalize disagreements between F and the segmentation s using a Hamming distance between this prior and the current segmentation mask: $f_{SD}(\mathbf{z}) = \gamma_{SD} \sum_{p_i \in P} \mathbf{1}(s(p_i) \neq \rho(p_i))$, where γ_{SD} is again a preset weighting term. f_{SD} now depends only on the flood fill prior and the segmentation. Since the flood fill prior at each iteration is fixed, we can merge the two terms f_{SD} and f_S, with the former being subsumed by the unary segmentation potential. This becomes:

$$\begin{aligned}\theta_S(\mathbf{z}, \rho) = \sum_{p_i \in P} \big(&\gamma_2 \big[s(p_i) \cdot \theta_F(p_i) + (1 - s(p_i)) \cdot \theta_B(p_i)\big] \\ &+ w_2 \mathbf{1}\left[s(p_i) \neq \rho(p_i)\right]\big)\end{aligned} \tag{12}$$

where θ_F and θ_B are as defined in Eq. (9). The overall segmentation energy is:

$$g_S(\mathbf{z}, \rho) = \theta_S(\mathbf{z}, \rho) + \gamma_3 \cdot \phi_S(\mathbf{z}) \tag{13}$$

4.4 Pose Estimation Terms f_P and f_{PS}

The terms relating to pose estimation are unchanged from the formulation of [17]. $f_P(\mathbf{z})$ denotes the cost of a particular selection of parts $\{b_k\}_{k=1}^{10}$. Each proposal j for each part i has an associated unary cost $\theta_P(i, j)$; connected parts i_1 and i_2 have associated pairwise costs, to penalize the case where parts that should be connected are distant from one another in image space. Defining $\mathcal{T}$ as the set of connected parts (i_1, i_2), the cost function is:

$$g_P(\mathbf{z}) = f_P(\mathbf{z}) = \gamma_5 \sum_{k=1}^{10} \theta_P(k, b_k) + \gamma_6 \sum_{(i_1, i_2) \in \mathcal{T}} \phi_{i_1, i_2}(b_{i_1}, b_{i_2}) \tag{14}$$

$f_{PS}(\mathbf{z})$ encodes the relation between segmentation and pose estimation: we expect foreground pixels to be close to some body part, and we expect pixels close to some body part to be foreground. The energy is written as:

$$g_{PS}(\mathbf{z}) = f_{PS}(\mathbf{z}) = \gamma_7 \sum_{j,k} \sum_{p_i} \left(\mathbf{1}(b_j = k) \cdot (1 - s(p_i)) \cdot w_{jk}^i \right) + \gamma_8 \sum_{p_i} \mathbf{1}(\max_{j,k} w_{jk}^i < \tau) \cdot s(p_i) \tag{15}$$

4.5 Energy Minimization

Our new energy function, with the terms defined in Sections 4.2 and 4.3, is as follows:

$$E(\mathbf{z}) = g_D(\mathbf{z}) + g_S(\mathbf{z}) + g_P(\mathbf{z}) + g_{PS}(\mathbf{z}) \tag{16}$$

In order to efficiently minimize this energy function, the label set $\mathcal{B}$ is binarized. Each body part B_i takes a vector of binary labels $b_i = [b_{(i,1)}, b_{(i,2)}, \cdots, b_{(i,N_E)}]$, where each $b_{(i,j)}$ is equal to 1 if and only if the j^{th} proposal for part i is selected. Note that since we solve g_D efficiently using range moves, and the term linking stereo and segmentation is now included in g_S, we do not need to binarize $\mathcal{D}$, as we can efficiently find a multi-class solution. This has an effect on the cost vector λ_D, which we discuss in Section 4.6. An assignment of binary labels to all pixels and parts is denoted by $\bar{\mathbf{z}}$. We introduce duplicate variables $\bar{\mathbf{z}}_1$ and $\bar{\mathbf{z}}_2$, and add Lagrangian multipliers to penalize disagreement between $\bar{\mathbf{z}}$ and its two copies, forming the Lagrangian dual of the energy function in Eq. (16). This is then divided into three slave problems using dual decomposition:

$$\begin{aligned} L(\bar{\mathbf{z}}, \bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \rho) &= g_{\mathcal{D}}(\bar{\mathbf{z}}_1) + g_S(\bar{\mathbf{z}}, \rho) + g_{\mathcal{P}}(\bar{\mathbf{z}}_2) + g_{PS}(\bar{\mathbf{z}}) \\ &\quad + \lambda_D(\bar{\mathbf{z}} - \bar{\mathbf{z}}_1) + \lambda_P(\bar{\mathbf{z}} - \bar{\mathbf{z}}_2) \qquad (17) \\ &= L_1(\bar{\mathbf{z}}_1, \lambda_D) + L_2(\bar{\mathbf{z}}_2, \lambda_P) + L_3(\bar{\mathbf{z}}, \lambda_D, \lambda_P, \rho) \qquad (18) \\ \text{where: } L_1(\bar{\mathbf{z}}_1, \lambda_D) &= g_D(\bar{\mathbf{z}}_1) - \lambda_D \bar{\mathbf{z}}_1 \qquad (19) \\ L_2(\bar{\mathbf{z}}_2, \lambda_P) &= g_P(\bar{\mathbf{z}}_2) - \lambda_P \bar{\mathbf{z}}_2 \qquad (20) \\ L_3(\bar{\mathbf{z}}, \lambda_D, \lambda_P, \rho) &= g_S(\bar{\mathbf{z}}, \rho) + g_{PS}(\bar{\mathbf{z}}) + \lambda_D \bar{\mathbf{z}} + \lambda_P \bar{\mathbf{z}} \qquad (21) \end{aligned}$$

Figure 3 shows the decomposition structure used in [17], and how it relates to our new decomposition structure. The change in L_3 means that the messages passed between the slaves and the master also change. The new message passing structure sees the flood fill prior passed from the master to the slave, instead of λ_D.

4.6 Modifications to $\boldsymbol{\lambda_D}$ Vector

Removing the explicit statement of f_{SD} means that the loss function L_3 from [17] no longer depends on finding values for stereo. Recall that an expected foreground disparity E_F, and a non-negative slack variable ξ, were defined in Section 4.1. This gives us a range $[E_F - \xi, \cdots, E_F + \xi]$ of disparity values which we associate with the foreground region.

When a segmentation result has been obtained, the λ_D cost vector can be adjusted by comparing the flood fill prior ρ_k with the segmentation result s_{k+1},

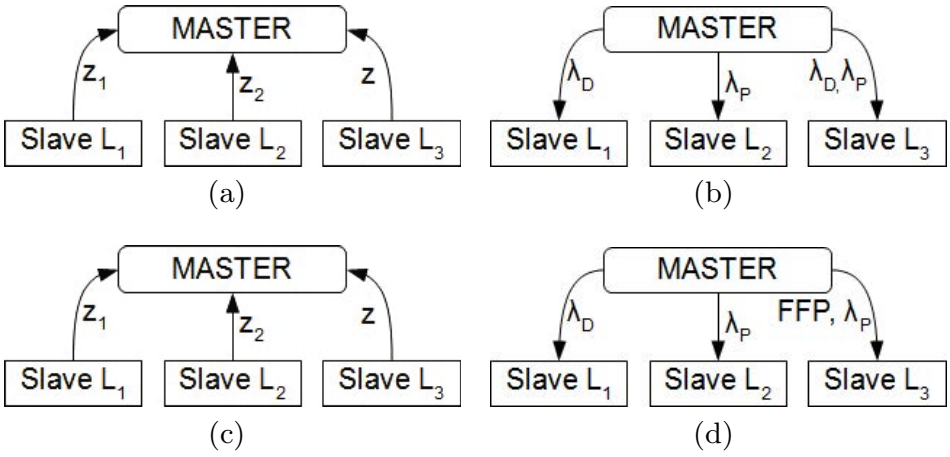


Fig. 3. Diagram showing the two-stage update process. **(a)** and **(c):** the slaves find labellings $\mathbf{z}$, $\mathbf{z}_1$, $\mathbf{z}_2$ and pass them to the master; **(b):** the master updates the cost vectors λ_D and λ_P and passes them to the slave. **(d)**: in the new formulation, the flood fill prior ρ is passed to the slave L_3 instead of the cost vector λ_D.

Table 1. Segmentation results on the H2view test sequence, compared with previous results. Using the RGB image to refine the result obtained by GrabCut on depth improves the result slightly, but our joint approach is superior.

Method	Precision	Recall	Overlap
Ours	**79·59**%	83·23%	**69·23**%
Two-stage GrabCut	60·00%	73·53%	49·34%
GrabCut on depth	69·96%	53·24%	45·14%
GrabCut on RGB	43·03%	**93·64**%	43·91%
ALE [10]	48·63%	41·07%	28·23%
Original formulation [17]	24·78%	30·01%	22·20%

and altering the cost for those pixels which for which the segmentation result disagrees with the flood fill prior. If a pixel is segmented as foreground despite the flood fill prior encouraging it to be segmented as background, this implies that the disparity costs within this range should be reduced, and the costs outside the range increased; conversely, if a pixel is background despite the flood fill insisting it is foreground, then the costs for disparities within the range should be increased. Therefore, we set the cost update vector λ_D as follows:

$$\lambda_D(x,y,d) = \begin{cases} \rho_k(x,y) - s_{k+1}(x,y), & \text{if } |d - E_F| < \xi \quad (22a) \\ s_{k+1}(x,y) - \rho_k(x,y), & \text{otherwise} \quad (22b) \end{cases}$$

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

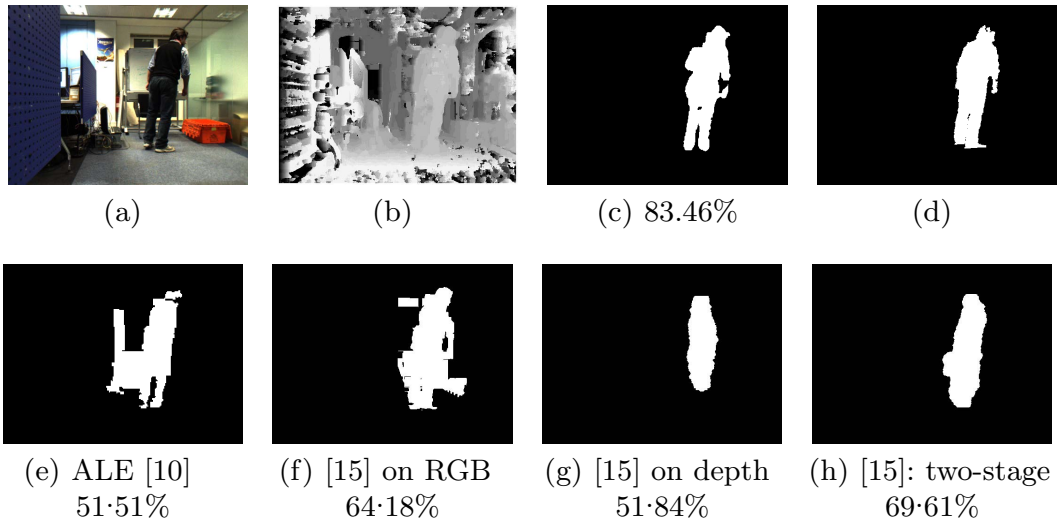
Fig. 4. Sample segmentation results. **(a):** RGB image; **(b):** result of [17]; **(c):** our result; **(d):** ground truth. Note that the flood fill prior can fill in gaps left by the original result. Additional results can be found in the supplementary material.

5 Experiments

To test the effects that our modifications to the formulation of [17] had on the results, we evaluate our algorithm on the H2view dataset introduced in [17].[1] This dataset comprises a sequence of images of a person standing, walking, crouching, and gesticulating in front of a stereo camera. To evaluate pose estimation, we use the standard probability of correct pose (PCP) criterion; in this section, we also present quantitative segmentation results, using the overlap (intersection over union) metric.

Segmentation: We used the H2view test sequence of 1598 images to evaluate our approach; for comparison, we also tested GrabCut [15], ALE [10], and our original algorithm [17]. We achieve a segmentation accuracy of 69·23%; some qualitative results are shown in Fig. 4. The accuracy is reasonably consistent over the sequence, with most images achieving accuracy close to the average, but that there is room for improvement in temporal consistency via video-based cues. It should be noted that the segmentation results are quite sensitive to initialisation; if the torso is classified incorrectly, the flood fill prior is likely to be

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



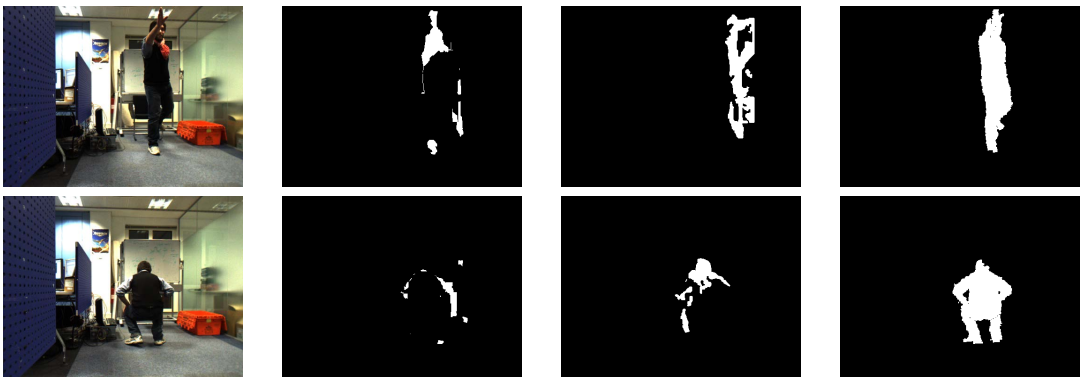
(a) (b) (c) 83.46% (d)

(e) ALE [10] 51·51% (f) [15] on RGB 64·18% (g) [15] on depth 51·84% (h) [15]: two-stage 69·61%

Fig. 5. Sample segmentation results, with overlap scores for this image. **(a):** RGB image; **(b):** Disparity map; **(c):** our result; **(d):** ground truth; **(e)-(h):** results of [15] and [10]. Note that ALE and GrabCut on RGB both classify parts of the window and other background areas as foreground, while GrabCut based on the disparity map fails to capture thin structures such as the arm and head.

Table 2. Results (given in % PCP) on the H2view test sequence

Method	Torso	Head	Upper arm	Forearm	Upper leg	Lower leg	Total
Ours	**96·3**	**92·4**	**77·3**	42·0	**89·3**	81·4	**76·86**
Sheasby [17]	94·9	88·7	74·4	**43·4**	88·4	78·8	75·37
Yang [19]	72·0	87·3	61·5	36·6	88·5	**83·0**	69·85
Andriluka [1]	80·5	69·2	60·2	35·2	83·9	76·0	66·03



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

Fig. 6. Failure cases of segmentation. **(a):** the original image; **(b):** result of [17]; **(c):** our result; **(d):** ground truth. In (c), incorrect torso detection means that the flood fill prior misleads the segmentation algorithm, resulting in the wrong region being segmented.

incorrect, leading to a bad segmentation result (Fig. 6). Developing a resistance to this pitfall is a promising direction for future research.

These results compare very favorably to those obtained by the other approaches we evaluated (a qualitative comparison is shown in Fig. 5). Our original algorithm [17] only achieved 22·2% accuracy, while ALE failed to generalize to the new location featured in the test set, achieving just 28·23% accuracy. In order to test GrabCut, we obtained initial bounding boxes by thresholding the foreground weight map W_F defined in Section 4.3. We ran the GrabCut algorithm on three passes through our test set: first, using solely the RGB images; second, on disparity values; and finally, running GrabCut the on disparity map and then refining the result with the RGB information. The results, given in Tab. 1, show that combining information from depth and RGB (as we do in "two-stage GrabCut") can improve the segmentation results, but the accuracy can be improved by a large margin by using a flood fill prior.

Pose Estimation: While we did not directly alter the pose estimation algorithm used in [17], our improved segmentation results also lead to a slight improvement in pose estimation results. Full quantitative results are given in Tab. 5.

6 Conclusion

This work has demonstrated the applicability of the flood fill algorithm for generating segmentation priors. Disparity maps generated by stereo correspondence algorithms can be used as input, and given a set of seeds with high probability of belonging to the foreground object, accurate segmentation results can be obtained. We have also incorporated our prior into the dual decomposition framework of [17], greatly improving upon the segmentation results. Our segmentation results show that combining information from depth and RGB can improve the segmentation results, but the accuracy can be improved by a large margin by using a flood fill prior.

We believe that the results presented in this paper, both for segmentation and for human pose estimation, can be further improved by adding video-based cues, such as motion tracks. This should improve temporal consistency, in particular giving the algorithm a chance to recover in the rare cases where the seeds used in flood fill were incorrect (shown in Fig. 6). Indeed, the centroid of the segmented object could be used as a seed for the next frame.

References

1. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. Computer Vision and Pattern Recognition (CVPR), 1014–1021 (2009)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 23(11), 1222–1239 (2001)

3. Bray, M., Kohli, P., Torr, P.: PoseCut: Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph-Cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 642–655. Springer, Heidelberg (2006)
4. Breiman, L.: Random forests. J. of Machine learning 45(1), 5–32 (2001)
5. Criminisi, A., Blake, A., Rother, C., Shotton, J., Torr, P.H.S.: Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. International Journal of Computer Vision (IJCV) 71(1), 89–110 (2007)
6. Gulshan, V., Lempitsky, V., Zisserman, A.: Humanising GrabCut: Learning to segment humans using the Kinect. In: International Conference on Computer Vision (ICCV) Workshops, pp. 1127–1133 (2011)
7. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. Computer Vision and Pattern Recognition (CVPR) 2, 407–414 (2005)
8. Kumar, M.P., Torr, P.H.S., Zisserman, A.: Objcut: Efficient segmentation using top-down and bottom-up cues. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32(3), 530–545 (2010)
9. Kumar, M.P., Veksler, O., Torr, P.H.S.: Improved moves for truncated convex models. J. of Machine Learning Research (JMLR) 12, 31–67 (2011)
10. Ladický, L., Torr, P.H.S.: The Automatic Labelling Environment, `http://cms.brookes.ac.uk/staff/PhilipTorr/ale.htm`
11. Ladický, L., Russell, C., Kohli, P., Torr, P.H.S.: Associative hierarchical crfs for object class image segmentation. In: International Conference on Computer Vision (ICCV), pp. 739–746 (2009)
12. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. Computer Vision and Pattern Recognition (CVPR), 1–7 (2008)
13. Microsoft: Xbox Kinect. Full body game controller from Microsoft (2010). `http://www.xbox.com/kinect`
14. Orchard, M.T., Bouman, C.A.: Color quantization of images. IEEE Transactions on Signal Processing 39(12), 2677–2690 (1991)
15. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics (TOG) 23(3), 309–314 (2004)
16. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision (IJCV) 47(1), 7–42 (2002)
17. Sheasby, G., Warrell, J., Zhang, Y., Crook, N., Torr, P.H.S.: Simultaneous Human Segmentation, Depth and Pose Estimation via Dual Decomposition. In: British Machine Vision Conference, Student Workshop, BMVW (2012)
18. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. Computer Vision and Pattern Recognition, CVPR (2011)
19. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. Computer Vision and Pattern Recognition (CVPR), 1385–1392 (2011)