Determining An Individual's 3-Dimensional Body Shape From Two 2-Dimensional Photographic Images

Problem presented by

David Evans

Poikos Ltd.

Executive Summary

Poikos are interested in the process of determining an individual's 3-D body shape from two 2-D images taken with standard hardware such as a camera phone or web cam. The study group addressed two particular issues in the overall process that Poikos would like to improve on, a markerless correction for radial distortion and improved segmentation of the person's outline from the image. Based on a radial distortion function the study group deduced and implemented an optimization method for finding the function parameters given straight lines in the distorted image. For the segmentation problem, the study group applied Perona-Malik pre-processing to improve edge detection in the image. An open source version of the 'segmentation by weighted aggregation' method was applied to the images and shows considerable promise. Together with prior information of the content of the image, this algorithm could provide better results than the current Poikos segmentation method.

Version 1.1
July 23, 2012
iii+16 pages

Report author

Rushen Patel (Industrial Mathematics KTN)

Contributors

Charles Brett (University of Warwick)
Laura Gallimore (University of Oxford)
Byron Jacobs (University of Witwatersrand)
Siân Jenkins (University of Bath)
Aleksandra Niechcial (Polish Academy of Sciences)
Rushen Patel (Industrial Mathematics KTN)
Colin Please (University of Southampton)

ESGI85 was jointly organised by
University of East Anglia
The Knowledge Transfer Network for Industrial Mathematics

and was supported by
The Engineering and Physical Sciences Research Council

Contents

1 Introduction

1.1 Background and scope

(1.1.1) The aim of the Poikos online 'MeasureCam' software is to determine a person's 3D body shape from two digital photographs taken with 'everyday' equipment such as a smartphone or web camera. The individual is requested to wear close fitting clothing and to adopt particular poses for the two photographs, at a distance of about 3m from the camera. One image is taken face-on to the camera, and the other is side-on to the camera. The user is also asked to input their height. An example pair of images is shown in Figure 1.

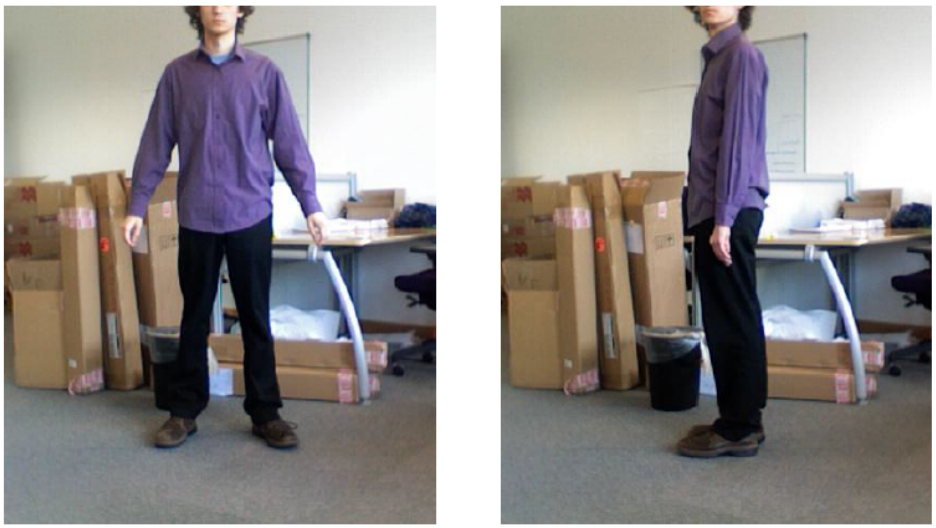

Figure 1: Example of the two poses captured of a person.

(1.1.2) There are a number of potential uses for 3D body shapes that can be obtained with relative ease in the home in this way, including online retail of clothing and applications in healthcare.

(1.1.3) The process of extracting a 3D body shape from the two 2D images is as follows

- The person in the two images is 'separated' from the background. This is currently achieved by comparing the image against a pre-calculated probability map of the pose, combined with colour information from the image and edge detection.
- Once the 2D silhouette for each input image is identified, it is compared against known 3D shapes using a 'nearest match' type system. Once a good fit is found, refinements are made to tune the shape to the subject.

(1.1.4) There is prior information available to aid the process. The image is assumed to contain one person who is cooperating by standing roughly in the right pose, at approximately the right distance from the camera. From a database of manually checked images Poikos have a probability map which gives the probability that a pixel is part of the person for each pixel in the image. From this, it is possible to look at areas which are highly likely to be part of the person and deduce the colour of the clothes the person is wearing. In the current implementation that Poikos have, this information is then used for edge detection.

1.2 Study group problem

(1.2.1) In the overall process of taking the two 2D images and converting them to a 3D body shape there were two main areas that Poikos wanted to address during the study group.

- A reduction in the errors in the 2D images due to distortions caused by defects in the camera lens, in particular the barrel (or radial) distortion.
- An improved method for extracting the human outline from the image (segmentation).

(1.2.2) Both of these problems are well studied in the fields of camera calibration and image processing however there are a number of points that raise difficulties in this particular application:

- There is no control over the lighting and shadow conditions.
- In some instances people have similar tastes in clothing and furniture. Therefore colour can sometimes be a misleading indicator.
- The background and foreground can be cluttered i.e. there could be objects between the camera and person in the field of view as well as behind the person.
- Webcams and mobile phone cameras are cheap and often have wide-angle lenses making them more prone to suffer distortion effects.
- There are no known standard objects in the images and the cameras cannot be calibrated before use.
- There are only two images of the person from which the 3D body shape is constructed.
- The process needs to be automatic with no input from the subject (except to actually stand in the correct pose and take the photos).

(1.2.3) On the other hand, there is some known information that can be exploited to improve the results:

- The picture is of a person standing in a known pose.
- The images are colour.

- In general the images will contain straight edges in the background e.g. skirting boards or door frames.
- For the segmentation problem, it is not necessary to have fully defined hands and feet. Poikos' main concern is that the body, arms and legs are correctly extracted.

(1.2.4) This report will address the two problems of distortion correction and segmentation in turn.

2 Distortion Correction

2.1 Radial distortion

(2.1.1) For a camera with a perfect lens, straight lines in the real world would be rendered as straight in the image. In reality however this is not the case and the image will suffer from some distortion that needs to be corrected for. These distortion effects are more pronounced in the cheap wide-angle lenses provided on webcams and smart phones. The two prominent types of distortion that Poikos encounter are radial and perspective.

(2.1.2) Within radial distortion the effect which is observed when using wide-angle lenses is known as barrel distortion (Figure 2). Perspective distortion occurs when the bore-sight (axis) of the camera is not parallel to the ground plane. However, this type of distortion is difficult to correct for without knowledge of a fixed angle in the image, in other words, without calibrating the camera. The study group therefore only sought to address the radial distortion in the image, which is the more pronounced of the two in any case.

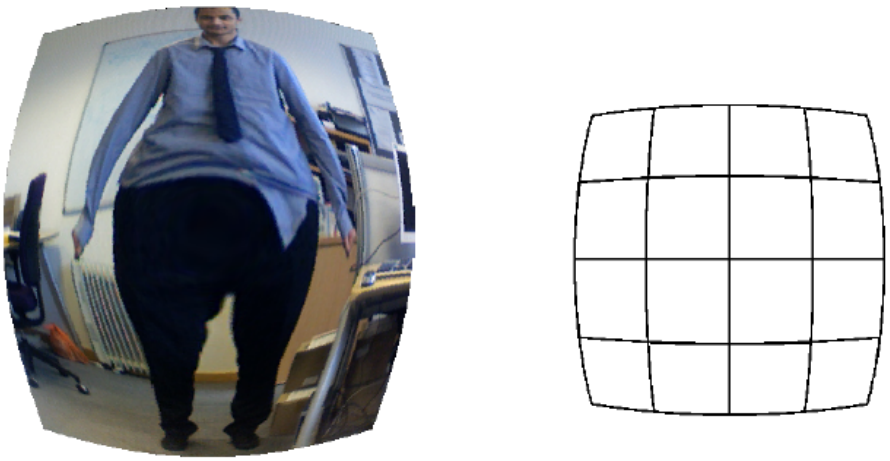

Figure 2: Exaggerated example of barrel distortion.

(2.1.3) The removal of radial distortion is commonly performed by first applying a parametric radial distortion model, estimating the distortion coefficients in the model and then correcting the distortion. The most commonly used radial distortion function has the following form:

$$r_d = rf(r, \mathbf{k}) = r(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...), \tag{1}$$

which is equivalent to

$$x_d = xf(r, \mathbf{k}), \qquad y_d = yf(r, \mathbf{k}), \tag{2}$$

where $\mathbf{k} = [k_1, k_2, ...]$ is the vector of distortion coefficients, r_d, x_d and y_d are the radial, x-directional and y-directional points in the distorted image and r,x and y are points in the undistorted image. The image centre, $r = 0$ is taken to be defined as $(x = 0, y = 0)$ and the relationship $r^2 = x^2 + y^2$ holds. The form of this distortion model can be traced back to an early study in photogrammetry [1]. The model assumes that distortion is radially symmetric around the centre of the image, which is a fair assumption in most cases [2]. The radial distortion function stated in [2] includes odd powers of r in the expression for f, we drop these terms to ensure the function is smooth at $r = 0$.

(2.1.4) Conventionally, when calibrating cameras, a known calibration object (such as a square grid of straight lines) is imaged. Since the true position of the points is known, a minimisation is performed to determine the distortion coefficients that minimise the error between the known points and the image points, after application of the distortion function. The process of correcting for radial distortion in the absence of a calibration object (i.e. known markers) is explained in the following section.

2.2 Markerless correction

(2.2.1) Through edge detection techniques, Poikos are able to identify features in the images, which they believe represent straight lines in the real world. For example the image of a door frame or section of skirting board which may be slightly curved in the image due to barrel distortion. The key idea behind our proposed method of markerless correction is to find the parameters $\mathbf{k}$ that best produce straight lines when applying an inversion of the distortion mapping described in (1) to the image. We will describe two methods, both explained with the simplifying assumption that the distortion is described by

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = (1 + k_1 r^2) \begin{pmatrix} x \\ y \end{pmatrix}, \tag{3}$$

although it is possible to generalise the methods if more terms in the barrel distortion function are required.

(2.2.2) First we consider a formal mathematical least squares approach. Suppose there are L distorted lines in our image and line j has N_j pixels on it, detected by Poikos' method for finding candidate lines in an image. The co–ordinates of the ith pixel of line j are $(x_{d_{i,j}}, y_{d_{i,j}})$, for $1 \leqslant i \leqslant N_j$ and $1 \leqslant j \leqslant L$. The true 'real world' position of the corresponding point is denoted $(x_{i,j}, y_{i,j})$, with $r_{i,j}^2 = x_{i,j}^2 + y_{i,j}^2$. We know that these points lie on a straight line, so $y_{i,j} = m_j x_{i,j} + c_j$. The indexing on the gradient m and the intercept c of the line is required because each of the L lines will in general have a different formulation. In order to find the k_1 that best undistorts these points to general straight lines we minimise

$$\sum_{j=1}^{L} \sum_{i=1}^{N_j} \left| \begin{pmatrix} x_{d_{i,j}} \\ y_{d_{i,j}} \end{pmatrix} - (1 + k_1 r_{i,j}^2) \begin{pmatrix} x_{i,j} \\ y_{i,j} \end{pmatrix} \right|^2 . \tag{4}$$

This objective function needs to be minimised over k_1 and the start and end points of each line. Given the start and end points of any line, the gradient m_j and intercept c_j can be found. Using the $x_{i,j}$ values, the relative $y_{i,j}$ values can be found. In the course of the study group this minimisation problem was implemented in Matlab using the in–built 'fminsearch' routine, which minimises unconstrained multivariable functions. In order to minimise the objective function, initial conditions are required for each minimisation variable, so that fminsearch has a starting point from which to begin. These initial conditions are usually found through a priori information on the variables.

(2.2.3) The second approach can be thought of as a quick and dirty approximation to the first. We note that

$$r_d = r(1 + k_1 r^2), \tag{5}$$

and we assume k_1 to be small. A typical value of k_1 for a camera phone is 9×10^{-8} [8]. Rearranging we see that

$$r = r_d - k_1 r_d^3, \tag{6}$$

so substituting back we find that

$$r_d = r(1 + k_1 r_d^2 + O(k_1^2)). \tag{7}$$

Thus we claim a reasonable approximation to the inverse of the distortion map is given by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{(1 + k_1 r_d^2)} \begin{pmatrix} x_d \\ y_d \end{pmatrix} . \tag{8}$$

We take the end points of the distorted line in the image, guess a k_1 and 'undistort' them using (8). These two points now define a straight line and the objective function is defined as the sum of the squared distances of

all the other 'undistorted' points to this straight line, summed over each of the lines. We then numerically minimise the objective function over possible choices for k_1. During the course of the study group this method was implemented in Matlab using the in–built routine 'lsqnonlin' which is designed for nonlinear least squares problems such as this.

(2.2.4) During the study group we trialled both methods on created test data. We took points on lines, selected a k_1 value, applied the barrel distortion described in (3) and added a small amount of noise to the positions, to model imperfect imaging and image processing. The points were then used as inputs and both correction algorithms were run to try and obtain the original k_1. The first approach attempts optimisation in a high dimensional setting in stark contrast to the second approach which optimises over one variable. Unsurprisingly the first was found to be sensitive to the initial guess for k_1 and could take a long time to find solutions. The second appeared much faster and more robust in the few tests we had time to conduct. The first method can be greatly improved by using the output from the second method as an initial guess for k_1. Adding regularisation terms with respect to these initial conditions, also improved the first method:

$$\begin{aligned} &\sum_{j=1}^{L}\sum_{i=1}^{N_j}\left\|\begin{pmatrix} x_{d_{i,j}} \\ y_{d_{i,j}} \end{pmatrix} - (1+k_1 r_{i,j}^2)\begin{pmatrix} x_{i,j} \\ y_{i,j} \end{pmatrix}\right\|_2^2 + \\ &|k_1 - kg|^2 + \sum_{j=1}^{L}\left(|x_{1,j} - xg_{1,j}|^2 + |y_{1,j} - yg_{1,j}|^2 + \right. \\ &\qquad \left. |x_{N,j} - xg_{N,j}|^2 + |y_{N,j} - yg_{N,j}|^2\right), \end{aligned} \tag{9}$$

where each variable containing g is the initial guess (held constant) determined by the second method.

(2.2.5) As a concrete example we considered two lines and distorted them with $k_1 = -0.01$. The test was performed on a remote computer with an Intel Dual-Core 2.5GHz processor. The second method took 0.6 seconds to return a k_1 value with 1.7% error when compared to the true value for k_1. The regularised first method was implemented using this value as an initial guess for k_1 together with initial data on the start and end points of the lines from the lines created by the second method. This implementation also included running the second method to generate the required initial values. The computation took 1.8 seconds and returned a k_1 value with 8% error when compared to the true value for k_1. The code actually produces a worse answer than the initial guess it is given, suggesting it has found a local minimum of the objective function. Based on the few tests that were completed during the study group, we recommend the second correction method. Although the first maybe useful in extreme cases where k_1 cannot be safely regarded as small. Finally we note that both methods

will struggle if they are given segments of distorted lines that are close to radial. This problem can be avoided by careful selection of the input lines. If that is not possible the importance of the errors associated with each line can be weighted by the shortest distance between the extended infinite line and its perpendicular distance to the origin (centre of the image).

3 Improved Segmentation

3.1 Colour splitting

(3.1.1) The capability of the current Poikos segmentation algorithm can be seen in Appendix A.1. The current algorithm operates by taking an area within the image with a high probability of being part of the person and analysing the colours that are present through clustering. This information combined with edge detection, followed by smoothing, is used to achieve the segmentation. The probability plot for the frontal pose is shown in Figure 3.

Figure 3: Probability man. Pixels with high probability of being part of the man show up as whiter in the image.

(3.1.2) One issue with the current method is that clustering via colours can lead to poor segmentation when objects in the image are the same colour as the person's clothes. The algorithm will struggle to distinguish between the two. A quick fix to this problem that the study group proposed was

to separate colours based on their probability of being part of the person via thresholding. For example, currently all pixels which are close to a colour (e.g. blue) would be clustered together in the RGB space leading to poor segmentation. However, if these pixels belong to one or more objects excluding the person, they can be clustered into separate regions by analysing the probability that they are part of a person. Pixels below a threshold probability can be separated as not part of a person as shown in Figure 4.

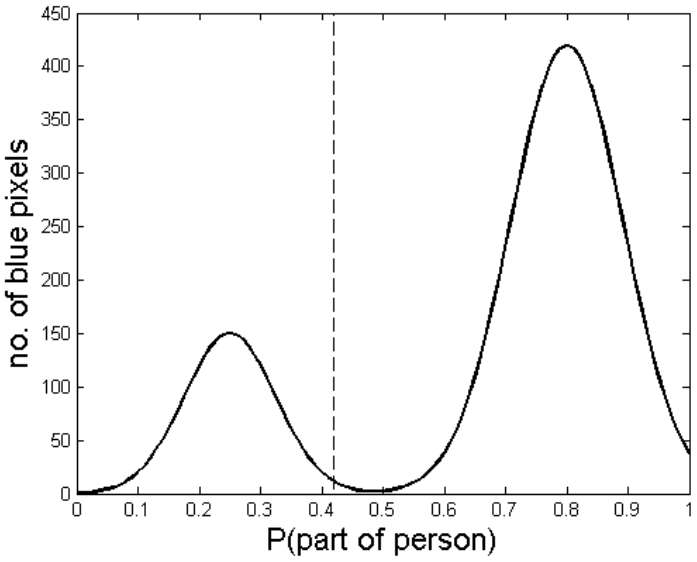

Figure 4: Regions of colour similar to the person can be separated via thresholding.

3.2 Smoothing

(3.2.1) The study group suggested that edge detection in the current Poikos framework or other segmentation techniques would be enhanced with image pre-processing. A technique for doing so is know as Perona-Malik diffusion [3], given as

$$\frac{\partial I}{\partial t} = \nabla \cdot (c_k \nabla I) \tag{10}$$

$$c_k\left(|\nabla I|\right) = e^{-|\nabla I|^2/k^2}, \tag{11}$$

where I is the image intensity, t is time, c_k is the diffusivity, ∇ is the divergence operator and k is a constant. The diffusivity c_k is not directional but does preferentially smooth areas of low intensity gradient, thereby leaving edges in the image sharp. The constant k controls the sensitivity to edges and is either chosen experimentally or as a function of the image

noise. An example of the application of Perona-Malik diffusion is shown in Figure 5.

Figure 5: Left - original image. Right - image after application of Perona-Malik ($k = 0.05$, end time $T = 20$)

3.3 Segmentation by weighted aggregation

(3.3.1) There are a number of segmentation methods that were discussed in the study group including active contour methods and graph cut algorithms.

(3.3.2) Segmentation by weighted aggregation is an algorithm that builds on the normalised cuts approach to image segmentation. In the general normalised cut method the image is modeled as a graph partitioning problem and a global criterion, the *normalised cut* is used for segmenting the graph. The normalised cut measures both the total dissimilarity between the different segments as well as the total similarity within the segments. There is an efficient computational technique based on the generalized eigenvalue problem that can be used to optimise the criterion.

(3.3.3) The pixels in the image form fully connected (undirected graph) $G = (V, E)$, where the nodes $V = (v_1, v_2, ..., v_n)$ in the graph are the pixels in the image and the edges E are formed between each pair of nodes. The weight on each edge, $w(i, j)$, is a function of the similarity between nodes i and j. For the simple case of partitioning the graph into two disjoint sets A, B, the *cut* can be defined as

$$cut(A, B) = \sum_{u \in A, p \in B} w(u, p) \tag{12}$$

and we can see that an optimal segmentation will minimise the cut value. When partitioning into multiple subgraphs this criterion favours cutting

small sets of isolated nodes on the graph since the cut as defined in (12) increases with the number of edges going across the partitioned parts. Therefore a normalisation is used which computes the cost as a function of the total edge connections to all the nodes in the graph. This *normalized cut* is defined as

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}, \tag{13}$$

where $assoc(A,V) = \sum_{u\in A, t\in V} w(u,t)$ is the total connection from nodes in A to all nodes in the graph. The exact minimisation of this criterion is NP-complete.

(3.3.4) In [6] the authors show that an approximate discrete solution can be found efficiently with the basic approach given as follows. Construct a symmetric cost matrix $W(i,j) = w(i,j)$ and define the diagonal matrix D as the sum of costs from node i, $D(i,i) = \sum_j W(i,j)$, $D(i,j) = 0$. The optimal cut is then given by

$$\min Ncut(A,B) = \min_y \frac{y^T(D-W)y}{y^T D y}, \tag{14}$$

with

$$y(i) \in \{1, -b\}, \quad y^T D\mathbf{1} = 0 \tag{15}$$

and

$$b = \frac{k}{1-k}, \tag{16}$$

where k is defined as the proportion of the sum of weights inside the set A to the total sum of the weights. The solution to the problem is given by the solution to a 'generalised' eigenvalue problem,

$$(D-W)y = \lambda D y, \tag{17}$$

which is solved by converting to a standard eigenvalue problem

$$D^{1/2}(D-W)D^{-1/2}z = \lambda z, \quad z = D^{1/2}y. \tag{18}$$

The second smallest eignevector of the generalised eigensystem is the approximate solution to the normalised cut problem. In [6] the authors show that the sign of the real valued $y(i)$ determines the set to which the node belongs. The algorithm can be extended to multiple segment partitioning straightforwardly. Once the graph is bipartioned, the algorithm can be run on each subgraph separately to give a 4-way partition and so on to give higher order partitions.

(3.3.5) The segmentation by weighted aggregation algorithm (SWA) takes a multiscale approach for recursively reducing the normalised-cut minimisation

problem [5]. Again the image is regarded as a weighted graph $G = (V, E)$ and the edge weight is given by

$$w_{ij} = e^{-\alpha|I_i - I_j|}, \tag{19}$$

where I_i, I_j are the intensities of pixels i,j, respectively, and α is a positive constant. Weights are only assigned to nearest neighbours. In SWA a pyramid structure of graphs (from the initial fine level to a graph with a single node i.e. a single segment) is created and the minimisation takes place recursively over the whole structure. At each level in the structure a single node (and the pixels associated with it) is considered a segment. The first fine level graph is constructed using half the pixels in the image as seed points and assigning the coupling weights. An adaptive coarsening procedure is then performed where representative nodes are selected as seeds for the coarser (higher level) graph. The coupling weights between the higher level nodes are computed by aggregating the statistics of surrounding nodes in the lower level graph and an interpolation matrix is computed between each level of the graph. A full description of the algorithm can be found in [7]. Figure 6 shows the pyramid structure for the image after the algorithm has been run.

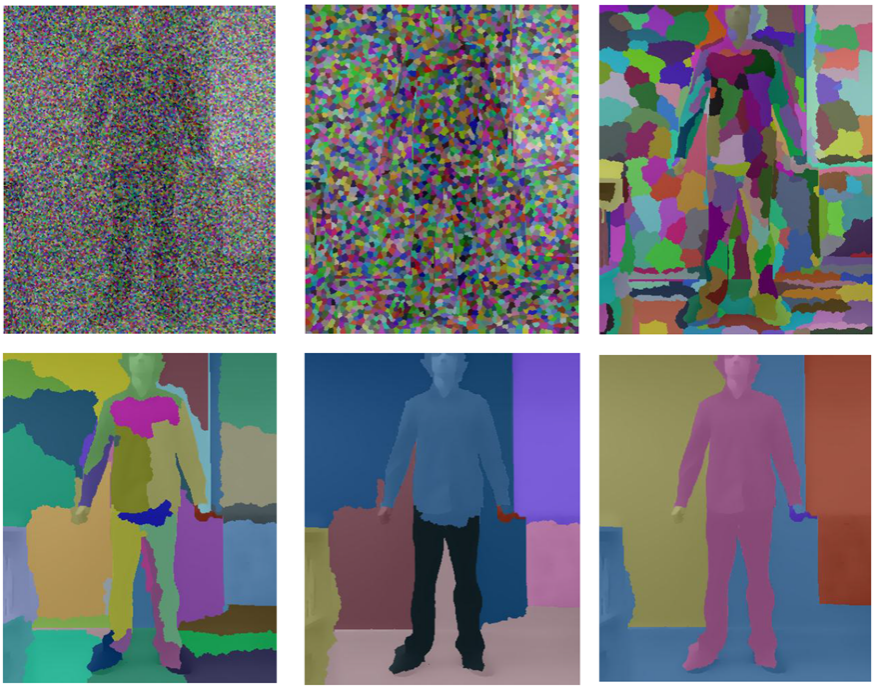

Figure 6: The pyramid structure of the SWA images, from fine level to coarse level.

(3.3.6) Using open source code from [4], the study group processed the Poikos test images, using SWA only and Perona-Malik preprocessing together

with SWA. The results are shown in Figures 7 - 8 and are encouraging. They show an improved performance when compared to the current Poikos implementation and this is achieved without optimising parameters in the algorithm or the use of prior information. The algorithm is also fast, with a computation time of the order of a few seconds on a standard laptop (Pentium 2.1GHz).

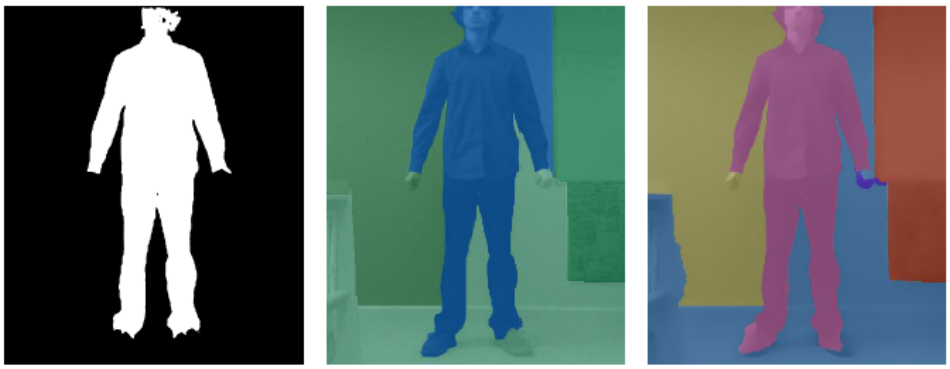

Figure 7: Left, current segmentation. Centre, SWA only. Right, SWA with Perona-Malik. SWA achieves a similar quality to the current Poikos segmentation

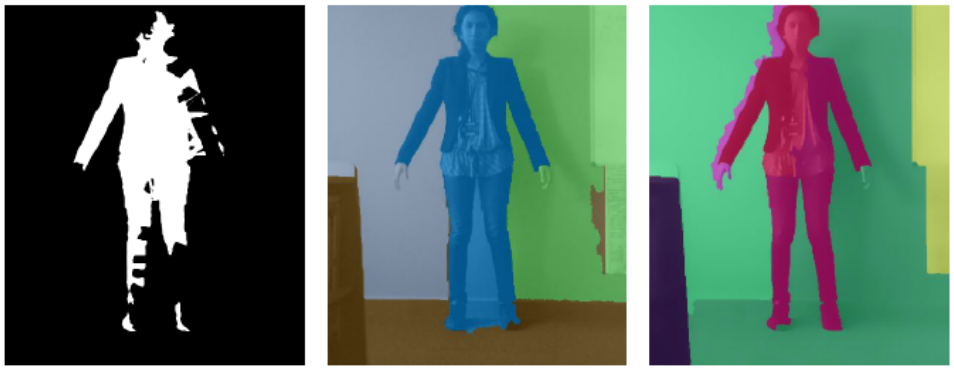

Figure 8: Left, current segmentation. Centre, SWA only. Right, SWA with Perona-Malik. SWA outperforms the current Poikos segmentation (Note that in the right picture the woman's right hand and the 'outer shadow' are a segmented separately to the body).

(3.3.7) The study group considered ways in which the SWA algorithm could be modified to include the prior information that Poikos have about the image as described in Section 1.2. The current edge weighting mainly considers intensity information with colour information only used in a crude way. It could be better incorporated by adding an additional term to the

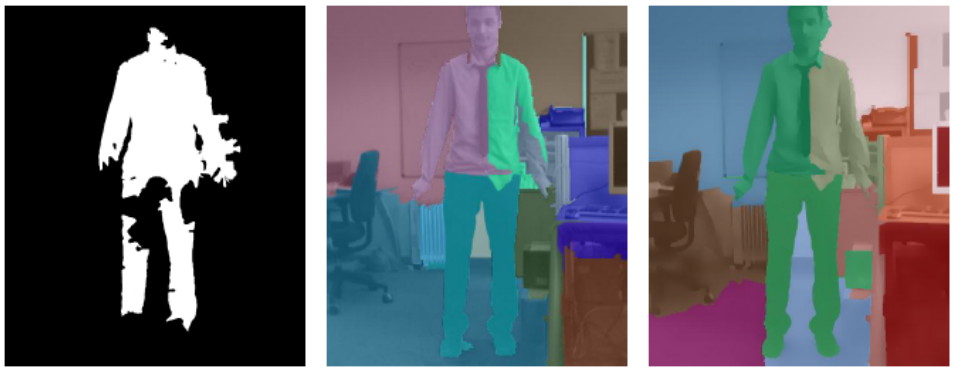

Figure 9: Left, current segmentation. Centre, SWA only. Right, SWA with Perona-Malik. SWA performs poorly as does the current Poikos segmentation. However, the inclusion of prior information would improve the SWA performance.

exponential,

$$w_{ij} = e^{-\alpha|I_i - I_j| - \beta\|c_i - c_j\|_2}, \tag{20}$$

where $c_i = (r, g, b)_i$ are the RGB values at the pixel, and β is a positive constant. With this modification stronger weight is applied to those connections that are similar both in intensity and colour. Information from the probability plot can also be included by incorporating another term in the weights to give

$$w_{ij} = e^{-\alpha|I_i - I_j| - \beta\|c_i - c_j\|_2} + \gamma(p_i + p_j - 1)^2, \tag{21}$$

where $p_i \in [0, 1]$ is the probability the pixel is part of the person, and γ is a positive constant. The additional term is large when both pixel probabilities are small (and are likely to be in the background / clutter) and large when both probabilities are high (and are likely to be part of the person). Another approach discussed by the study group is to use the knowledge that the extracted silhouette must be approximately symmetric to improve the results, specifically at which iteration to stop the aggregation.

4 Conclusions

4.1 Remarks

(4.1.1) Markerless radial distortion correction can be achieved through minimisation of distorted points to a straight line.

(4.1.2) Perona-Malik diffusion is a promising pre-segmentation smoother, that can be applied whatever segmentation algorithm is chosen.

(4.1.3) Segmentation by weighted aggregation (SWA) is a fast and robust way of obtaining the human silhouette from a 2D image. It gives good results (better than Poikos currently achieve), particularly when Perona-Malik pre-processing is applied.

(4.1.4) SWA can be tuned for the Poikos application by including colour and probability information into the weights. Using the prior knowledge of human symmetry might also be beneficial.

4.2 Suggested further work

(4.2.1) A systematic analysis of the error and run times for correcting radial distortion.

(4.2.2) Implementation, parameter optimisation and testing of modifications to the SWA algorithm.

(4.2.3) Microsoft Office tools have in-built segmentation functions which operate very well (albeit with some user input). These algorithms use trained Bayesian networks to perform the segmentation and the study group recommends a review of these methods. It might be also be possible to combine Bayesian networks with SWA.

A Appendix

A.1 Poikos current segmentation quality

(A.1.1) The quality of segmentation currently achieved by Poikos is shown in Figures 10 - 12.

Figure 10: An example of good segmentation.

Figure 11: The lighting and shadow causes the current algorithm problems resulting in artifacts in the segmentation.

Figure 12: The presence of background and foreground clutter causes problems with the segmentation.

Bibliography

[1] C. Salma: *Manual of Photogrammetry*, American Society of Photogrammetry, fourth edition, 1980

[2] R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edition, 2004

[3] P. Perona and J. Malik: *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12:7, 1990

[4] *http://www.wisdom.weizmann.ac.il/ swa/index.html*

[5] E. Sharon, M. Galun, D. Sharon, R. Basri and A. Brandt: *Heirachy and adaptivity in segmenting visual scenes* Nature, Vol. 442(7104): 719-846, 2006

[6] J. Shi and J. Malik: *Normalized Cuts and Image Segmentation* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 22:8, 2000

[7] E. Sharon, A. Brandt and R. Basri: *Segmentation and Boundary Detection Using Multiscale Intensity Measurements* Conference on Computer Vision and Pattern Recognition, 2001

[8] *http://scien.stanford.edu/pages/labsite/2007/psych221/projects/07/geometric_distortion/project.htm*