Click!

A study in automatic parameter selection for Computer Vision algorithms

Dragos Bozdog[1*], Ionut Florescu[1], Rustam Stolkin[2]

[1] *Department of Mathematical Sciences, Stevens Institute of Technology, USA*

[2] *School of Computer Science, University of Birmingham, UK*

[*]corresponding author: Tel: 201 216 5447; Fax: 201 216 8321 Email: dbozdog@stevens.edu

ABSTRACT: This work investigates the possibility of designing a self-calibrating computer vision algorithm for given tracking or segmentation methods. We concentrate on a (suitably modified) segmentation algorithm, the meanshift segmentation of Comaniciu and Meer (2002). We show that even for a basic algorithm such as the one under study, the same set of parameter values cannot be universally optimal. However, we determine that within the same image (visual conditions) it is possible to have the same set of parameter values behaving in an optimal way for the majority of objects depicted. Furthermore, we show that it may be possible to classify the images by situations and use the same parameter values within a class. The conclusion indicates that it is possible to gather preliminary information about the image itself (e.g., degree of clutter, range of the color space, dispersion of colors, scene dynamics, etc.) which will indicate the best parameter values to use in the algorithm.

Keywords: optimal parameter choice; computer vision algorithms; object segmentation;

1. INTRODUCTION

Many vision papers are concerned with development of the best and the fastest computer vision algorithm. However, in many cases when independently implemented these algorithms fail to perform as advertized. In many situations the vision algorithms are very sensitive to the choice of parameters values used. Choose the parameters detailed in such papers, but apply the algorithm to a different image sequence than the one presented in the article and very likely one has an underperforming algorithm.

This is the premise of the current article. We wish to study if it is even possible to determine the best parameter value as a function of the current video sequence under study. In optimization terms we wish to study the structure of the function to be optimized around the position of the best parameter values. If the function is relatively flat in the minimum region then a large range of parameter values would be performing similarly. On the other hand if the function is changing sharply around its minimum then determining the right parameter values is crucial.

The study presented herein is applied but the principled and the empirical results may be replicated and ported to any choice of computer vision algorithm. We consider one of the most popular segmentation algorithms *the meanshift procedure* (Comaniciu and Meer, 2002). We are applying the afore mentioned algorithm to a nonstandard segmentation technique and thus the published algorithm had to be modified in a substantial way. After modifications, only the basic meanshifting procedure as detailed in Comaniciu and Meer (2002) remains the same in the current article.

Objectives of the current work:

- For a given image start with a clicked pixel position contained within an object of interest. Adapt a segmentation algorithm to output the position and color distribution of the object.
- Study the shape of the output as a function of parameter values.
- Investigate the possibility of adapting the algorithm's parameters in response to the specific characteristics of the image.

The article is structures as follows. Section 2 presents the modified segmentation algorithm. Section 3 presents details of the statistical analysis and conclusions drawn when looking at the optimal parameter choice of the algorithm. Section 4 concludes.

1.1 Problem Motivation

Our prior work Stolkin et al (2007) and Stolkin et al (2008) detail the construction of a computer vision tracking algorithm (the ABCshift tracker). Assume that we are given a sequence of images (a video sequence) which contain an object which needs to be tracked. Tracking here means outputting at a minimum the (in frame) position of the object in question. To apply this work to real life situations we were faced with the problem of initializing the algorithm. In other words there must exist some way of starting the tracking process by recognizing the object to be tracked.

There is a wealth of papers available which detail various types of automatic initialization of the tracker. In general, any automatic initialization involves motion segmentation and object classification. We refer to Wang et al (2003) for a detailed description in the context of human motion tracking. In general, to avoid misclassification the tracker needs to recognize highly specialized objects such as boats on a river Kamberov et al (2005), people entering rooms and showing their face to the camera Numiato et al (2002) to cite some examples.

All of these automatic object detection methods rely on parameters which makes the respective methodology susceptible to fail given varying conditions of the input video. The more specialized the object to be detected the less prone is the method to fail given varying input conditions, but at the same time the generality of the method is lost.

All of these considerations lead us to consider the primary segmentation problem in computer vision. In the simplest form this problem can be expressed in this way: given an image containing one object, devise an algorithm that will categorize the pixels in the image into object pixels and other pixels. In a more general form segmenting an image implies dividing the image into “segments" all representing distinct objects and/or background present in the image. In this article we are concerned with devising a simple segmentation strategy to recognize an object present in an image.

Specifically, we are considering the following situation. The user of the system suddenly sees an object of interest while watching a real time video feed. He then clicks or identifies to the best of his/her abilities a pixel on the screen within the object to be tracked. We wish to design a method that will be able to start with this information and segment the object and the background. The method has to be quick (since the process is in real time) and *robust*. The robustness is the key notion studied here.

1.2 Differences from Traditional Segmentation

We note that we have a bit of extra information versus the traditional automatic motion detectors or segmentation methods. We know that an object is present in the image and we know that some of its characteristic features may be found by analyzing the location at (or close to) the pixel clicked. However, unlike the traditional motion detectors the object may be completely stationary, the camera may be moving and the location clicked may not define the most distinguishing feature of the object from its surroundings. Furthermore, the object may be of arbitrary shape or form.

Unlike the traditional segmentation algorithms we do not wish to segment the entire image, only the object(s) close to the original location clicked. However, the particular object segment has to be precise since this information is to be passed to the object tracker and the performance of the tracker depends directly on the characteristics of the original segment. We mention Georgescu et al (2003) for an application of the meanshift algorithm to clustering the pixels of an entire image into objects.

The tracker we designed (Stolkin et al, 2007, Stolkin et al, 2008) is capable of running in real time while automatically pan-tilt-zoom the camera toward the object of interest as long as it knows its characteristics. However, it cannot do so once the object of interest leaves the field of view. For this reason, the goal is to design a fast algorithm (hereafter referred to as *Click!*) to output the position and color distribution of the object in milliseconds.

2. DESCRIPTION OF THE SEGMENTATION PROCEDURE

We will use a variant of the meanshift idea (Cheng, 1995; Comaneciu and Meer, 2002). We start with the initial frame defined as the frame that the user chooses to click on. The initial frame is imported for analysis and each pixel is characterized by its location and color. We choose to work with the CIE-LUV (henceforth LUV) color space since it was designed so that two colors that are equally distant in the LUV color space (in the sense of regular Euclidian distance) are also equally distant perceptually (in the sense of human eye perception of color on a monitor screen).

2.1 Initialization

The first ingredient we are using to insure a robust system is to choose a small square box around the initial pixel clicked and treat all the pixels in this original box as being characteristic of the object. The size of this initial box is given by a first parameter henceforth denoted *Initbox*. Then every pixel in this initial box is mean-shifted according to the algorithm detailed in Comaneciu and Meer (2002).

The meanshift algorithm requires the choice of two parameters h_r and h_c. The first parameter h_r determines the range of pixels centered in the meanshifting point which are included in the calculation (it describes the typical range of pixels in the physical image that affects the neighboring pixels –one can think about it as describing the average size of the segments in the image). The second parameter h_c defines a similar range but in the color space. These two crucial parameters are entirely subjective and an automatic way of deciding them needs to be devised if the system is to be robust. We postpone the discussion about choosing them for later in the paper. For a visual description of the importance of these two parameters we direct the reader to the famous Baboon image (Figure 5) in Comaneciu and Meer (2002).

2.2 The Meanshift Procedure. Finding the "meanshifted" color values for each pixel in the initial box.

For a given point with coordinates x_0 and y_0 the points evaluated in this step are found within the range $(x_0 - h_r, x_0 + h_r) \times (y_0 - h_r, y_0 + h_r)$. Two coefficients are calculated for each such point with coordinates (x_i, y_i):

$$K_r(x_i, y_i) = k_r\left(\frac{x_i - x_0}{h_r}, \frac{y_i - y_0}{h_r}\right) \qquad (1)$$

$$K_c(L_i,U_i,V_i) = k_c\left(\frac{L_i - L_0}{h_c},\frac{U_i - U_0}{h_c},\frac{V_i - V_0}{h_c}\right) \tag{2}$$

where

$$k_r(x,y) = \left(1 - \sqrt{x^2 + y^2}\right) \cdot 1_{\{x^2+y^2 \leq 1\}} \tag{3}$$

$$k_c(L,U,V) = \left(1 - \sqrt{L^2 + U^2 + V^2}\right) \cdot 1_{\{L^2+U^2+V^2 \leq 1\}} \tag{4}$$

We use the notation 1_A for the indicator of a set *A* (i.e., the function equal to 1 if the point (x, y) is in the set *A* and equal to 0 otherwise).

The weight is computed for each point within the window limits using the following relation:

$$W_i = K_r(x_i,y_i) \cdot K_c(L_i,U_i,V_i) \tag{5}$$

Finally, we find the location of the meanshifted point corresponding to the centroid of all the points in the window centered at (x_0,y_0) using:

$$x_{CG} = \frac{\sum x_i W_i}{\sum W_i}$$
$$y_{CG} = \frac{\sum y_i W_i}{\sum W_i} \tag{6}$$

If the distance in color space between the initial and the meanshifted point is greater than an initial defined parameter d_c, then the process is repeated by replacing (x_0,y_0) with (x_{CG},y_{CG}) and repeating the procedure. If the following condition is satisfied, then the final pixel LUV color is associated to the initial pixel.

$$d[(x_{CG},y_{CG}),(x_0,y_0)] < d_c \tag{7}$$

Using this procedure, any pixel in the original image is associated with a specific color in a virtual color space which is referred to, from now on, as the meanshifted color space. The pixel classification based on color affinity proceeds using this new space.

2.3 Characterization of Initial Object

The first step in the object classification procedure is the evaluation of all the pixels in the initial box whose size is defined by *Initbox*. The evaluation of all these pixels is needed to provide sufficient information for the preliminary characterization of the meanshifted color space. Furthermore, it solves imprecision issues related to the accurate selection of the initial point, as well as the existence of various points with uncharacteristic color space with respect to the object of interest. All the points within the boundaries of the initial box are classified as object and the meanshift procedure above is applied to each point. The resulting LUV set in the meanshifted color space is then evaluated, after which we estimate the mean and the covariance matrix for the points. The classification process implements an ellipsoid as an object-background classification criterion. Specifically given a set of n points $\{(L_i, U_i, V_i)\}_{i\in\{1,2,\dots,n\}}$ we calculate their mean $(\bar{L}, \bar{U}, \bar{V})$ and their covariance matrix S.

Under the assumption that the color within the object in the meanshifted space vary according to a multivariate normal distribution with mean vector μ we have that:

$$\left((\bar{L}, \bar{U}, \bar{V}) - \mu\right)^T \mathrm{S}^{-1}\left((\bar{L}, \bar{U}, \bar{V}) - \mu\right) \sim \frac{(n-1)p}{n(n-p)} F_{p,n-p} \qquad (8)$$

where $F_{p,n-p}$ denotes a Snedecor F distribution with numerator p and denominator n-p degrees of freedom. Thus we can find the level sets of the ellipsoid determined by the points using this distribution. However, in our case $p=3$ and n is much larger than p so we use the following result:

$$\frac{(n-1)p}{n(n-p)} F_{p,n-p}(\alpha) \xrightarrow{n-p\to\infty} \chi_p^2(\alpha) \qquad (9)$$

for every percentile α. This simplifies the code in the sense that we do not need to recalculate quantiles of the F distribution for every step. Using this result we define the ellipsoid with center at $(\bar{L}, \bar{U}, \bar{V})$ and axes given by the correlation matrix S as the set of coordinate points (L, U, V) in the meanshifted space with the property

$$\left((L, U, V) - (\bar{L}, \bar{U}, \bar{V})\right)^T \mathrm{S}^{-1}\left((L, U, V) - (\bar{L}, \bar{U}, \bar{V})\right) \leq \chi_3^2(\alpha) \qquad (10)$$

The α level is a new parameter introduced in our definition of the algorithm and we shall study its influence on the output in the statistical analysis which follows. The typical values of alpha are 0.95, 0.97, 0.99. $\chi_3^2(\alpha)$ denotes the α-quantile of a chi-squared distribution with 3 degrees of freedom.

2.4 Dynamic Evaluation and Classification System

The objective of the main algorithm is to evaluate the neighboring points of the previously classified object and extend the limits of the object by evaluating new pixels in a dynamic classification environment. We start with the ellipsoid in the meanshifted space defined by the pixels in the *Initbox* according to equation (10). All the pixels already in the ellipse are assigned a code 2 = object pixel. All the other pixels are assigned a code 0 = not classified.
At each iteration we evaluate a point in the image which neighbors a previously classified object pixel. We first calculate its (L, U, V) coordinates in the meanshifted space. Then we check the condition (10). If it is not satisfied, the pixel is assigned a code 1 = not an object or criteria not large enough. If the condition is satisfied then the pixel is assigned a code 2 = object and the ellipsoid center $(\bar{L}, \bar{U}, \bar{V})$ and the covariance matrix *S* are updated to include the information contained in the new object pixel. The updating process can be done very fast by just updating two running sums (see for example West (1975)). If at any given iteration step, one object pixel is completely surrounded by other object pixels, then the pixel is marked accordingly and skipped in the next iterations.

Using this procedure, the number of new pixels evaluated is kept within reasonable range and repetitions are eliminated. However, the pixels not classified as object in the previous steps, but connected with object pixels, will continue to be evaluated as the criteria of acceptance changes with each new pixel accepted. This is needed as the theory tells us that the means $(\bar{L}, \bar{U}, \bar{V})$ vector and the covariance matrix *S* will stabilize to their true values, but at the start of the algorithm there may be pixels erroneously classified as not belonging to the object. Furthermore, there may exist pixels erroneously classified as object pixels in the beginning, but they are generally few due to the conservative way in which we construct the ellipsoid. Recall that the ellipsoid level sets are given by quantiles corresponding to 95%, 97% and 99% confidence intervals meaning that we are in fact eliminating some points that may belong to the distribution.

2.5 Object, Boundary, and Stopping Criteria

The iterative algorithm is evaluating the pixels located at the physical boundary of the object pixels (code 2) and the search region expands as new classified pixels are included in the object. The algorithm stops when no more new pixels can be classified as object. Any pixel code 2 which has a neighbor code 1 is categorized as a boundary pixel. The algorithm paints all these boundary pixels with a different color for visualization of the segmentation result.

This concludes the first objective declared in the paper. The algorithm outputs the position and color components of the object under study. Next we address the second objective – parameter adaptation to the local conditions of the image.

3. OPTIMAL PARAMETER CHOICE

We are studying the influence of four parameters, two originally present in the meanshift algorithm:

h_r = The range of pixels in the image included in the meanshift calculation

h_c = The range of pixels in the color space included in the meanshift calculation,

and two new ones introduced by our specific algorithm

Initbox= Size of the initial selection containing object pixels

α = Confidence level defining the ellipse level set

We test the possibility of choosing the best parameters in the algorithm described above using the Berkley image dataset (Martin et al, 2001). This database contains 1000 images hand segmented by various users. Since we perform a detailed statistical study we had to choose a limited sample from these images. We select five images of different structural types. The images chosen are presented in Figures 1-5. We pick a set of four to five objects in each image and we perform the statistical analysis for each of the objects. The LUV color space for each image is very diverse which is why we chose these particular images in our study.

[Figures 1 – 5 about here]

The objects chosen in each image are described in the figure captions. With one exception (image 2) the objects we choose are potentially moving human or animal. Image 2 is special since it depicts large homogeneous objects composing the same individual. The objects under study have to be relatively homogeneous in the color space since we use the meanshifted LUV space as the primary tool for segmentation. This is not a drawback, once the segments are determined properly they could be combined into an overall description of the object. Similar work combining the output of two segmentation methods had been developed recently by Millet et. al. (2010). The images presenting the corresponding LUV space are obtained using the 3D Color Inspector program of Barthel (2007).

The data to be analyzed is obtained by choosing reasonable parameter levels. Specifically, we consider the following levels for each parameter:

- *Initbox*: {3,5,7,9,11}
- h_r: {3,5,7,9,11}
- h_c: {1,2,3,4,5,10,20,50}
- α: {95%, 97%, 99%}

The number of possible parameter combinations is 600. We repeat the segmentation for 10 different randomly chosen starting points within each object, which gives a total of 6,000 data points for each object. We analyze a total of 23 objects in 5 images and thus the total number of observations in our study is 138,000. For each object chosen, the Berkley data set records the true segment, as determined by human operators. Thus, for each of our data points we run the segmentation algorithm and we record two types of errors:

Error I = number of object pixels erroneously classified by the algorithm as background

Error II = number of background pixels erroneously classified by the algorithm as object

Then we calculate a response variable Y as the total error expressed as a proportion of total object size:

$$Y = \frac{Error\ I + Error\ II}{ObjectSize} \tag{11}$$

Clearly, the two types of error are fundamentally different and could have penalized more one type or the other, however for the current analysis we decided to penalize them equally.

The objectives of the statistical analysis:

1. Determine if there exist parameters that may be studied independently of the other parameters.
2. Determine if there exist a set of parameters which are optimal for all the objects *within an image*
3. Determine if there exist a set of parameters optimal for all the objects *in all the images*.
4. Determine relations between the input conditions and the optimal parameter choice

Objective 1: Can we simplify the analysis?

To answer this question we run a ANOVA study with 4 factors (one for each parameter) and the response variable *Y*. We included all the interaction terms to answer this question. The analysis was performed using SAS statistical language and a summary of all of the 23 analyses of variance performed is presented in Table 1.

[Table 1 about here]

If there is one parameter whose interaction with others is non-significant then the analysis becomes much simpler since we can study that parameter separately from all the others and chose its best level regardless of the values of the other parameters. Unfortunately, as we see from the table in the vast majority of cases the interaction between terms is significant. Although, there exist situations where one parameter does not interact with the others (e.g., in image 2 object 3, the α-level does not interact with the other parameters), in general the parameter interaction is present. Therefore, the answer to this question is no, we cannot consider the factor levels separately; instead we have to consider the combination of all four factors.

Objectives 2 and 3: Is there a set of parameters optimal for all the objects in a given image? Is there a set universally valid for all images?

We grouped these two objectives together since their study is very similar. To answer the question we run the analysis for each object at all the 600 total combinations of parameter levels. The objective 1 told us that we have to proceed in this way; no reduction of this set is possible. For each object we chose the combination that gives the best (smallest) error percentage. Then we calculate the standard error of the response and we construct a 99.7% confidence interval. We took a conservative approach which is why we fixed a relatively high confidence level (high confidence means large interval which means we keep much more combinations ergo conservative). Any combination of parameter values from the possible 600 which produces an error rate within this interval is retained as an optimal choice. In statistical terms: any combination kept did not produce a statically different output from the best possible combination and this statement is produced with confidence 99.7%.

Proceeding in this way we find that a large number of combinations are optimal for each object considered. Table 2 present the number of optimal choices for each object studied. We do not present the actual optimal sets due to lack of space. This large number of choices represents good news for the algorithm, since it means that we can vary the parameters quite significantly and still find a good segmentation object/background. We should also mention that for a particular instance: the soldier uniform in image 3 it appears from the table that all the combinations are optimal. In fact this is not the case, the best combination gave a high error rate (70%). The confidence interval was very large and it contained all the parameter combinations. Indeed, in this particular case all the combinations were subpar and there was no combination that produced good visual results.

[Table 2 about here]

To answer objective 2 we calculated the intersection of all the optimal sets for all the objects within each image. To our surprise, even though each object has a large set of optimal parameter values when we look at common values the number of optimal combinations of parameters is very small for each image. The set of best parameter values is presented in Table 3. In one instance (image 3) the intersection is empty.

[Figures 6 and 7 about here]

Furthermore, it is very clear that the common parameter set used is not the best possible for the particular objects. If we chose our confidence interval less conservatively (lower than 99.7% confidence will produce narrower intervals) we do not find a set of common parameter values anymore.

Objective 3 is clearly answered with a resounding NO. It is not possible to have the same set of parameters working well for all situations. Thus clearly the parameter values have to be adapted to the image conditions.
Answering objective 2 which asks if we can make the parameters dependent on image conditions is not so simple. It appears that in most situations there is a very small subset of parameters that behave in an optimal way for all the objects in an image. We present the segmentation of all the objects in an image for these parameters in Figures 6 and 7. In one instance (image 3) one object was not segmented successfully but that seems to be related to the size of the object (much larger than the other objects in the image) and the color distribution of the respective object (not an ellipse in the LUV space). Regardless, the fact that there are so few optimal choices indicate that the relation between parameters and the object itself is more important than the relationship parameters-image.

Objective 4: Inferring relationships between object characteristics and the optimal parameter values.

Analyzing Table 3, we see very clearly that there are commonalities between these values. This indicates that the error rate is a smooth function and it should be possible to minimize it - of course if ground truth is present. However, in general ground truth information is absent and the best we can do is to try and deduce relationships between parameters and images.

Image 1 contains drastic changes in color and the objects are many and fragmented. Clutter is a very important parameter to describe the image. The optimal parameter choice is obtained when the search range in space $(h_r, Initbox)$ is large and the range in color h_c is small. Furthermore, the ellipse is growing with the smallest radius $\alpha = 95\%$.

Image 2 is in contrast with image 1. Objects are big and the color does not present much variation. Clutter once again would describe well this image (low value). We can also see that the parameters are taking opposite values from the previous situation (space range small, color range large and largest ellipse radius).

Image 3 does not have an optimal set. However we can investigate the reason why this is so by looking at the optimal values for each of the three small objects (the large one is irrelevant). Objects 1 and 2 in this image (child head and soldier's head) have optimal values similar with the objects in image 1. However object 3 (child's suit) is different, the optimal values are large in both space and color. This seems to be related with the fact that the suit has a particular two color suit distribution so the dispersion in the object color is very relevant for this case.

Image 4 optimal values resemble the ones obtained for Image 2. However the space range h_r has smaller values than in that case reflecting that the color is homogeneous but that the objects are not that large.

Finally, for image 5 the defining characteristic of parameter values seems to be that space range and color range are in inverse relationship.

Based on this analysis we see that the following quantities calculated in a region close to the pixel originally selected definitely are related to the optimal values of the parameters:

- A local measure of clutter
- A local measure of variation of color histogram
- A degree of texture change

This preliminary study indicates that the following parameter values may provide better results:

Situation	$Initbox, h_r$	h_c	α
Small objects, homogeneous color, cluttered background	large	small	small
Large objects, homogeneous color, clear background	small	large	large

Small objects, non-homogeneous color, clear background	large	large	any
Small objects, homogeneous color, clear background	small	small	Small

4. CONCLUSION AND DIRECTIONS FOR THE FUTURE

In this paper we have shown that the proper choice of parameter values makes a world of difference for the results produced by a computer vision algorithm. We have investigated the possibility of choosing the parameter values automatically.

First, we have shown that this is an absolutely crucial issue. It is not possible to use the same parameter values in different image situations and continue to produce good results. Furthermore, even for the objects within the same image the set of optimal values is very small and therefore very hard to actually reuse the best parameter values for other objects.

However, hope is not entirely lost. By looking at the actual optimal values and how they relate to situations in specific images we see that consistent relations can be seen that may be associated with the local image conditions.

Clearly, more specific situations would provide better insight. Furthermore, a quantitative study relating the local measures proposed with the optimal parameter choice can be beneficial for the art.

We should mention that if we assume that the objects segmented do not have holes an improved segmentation algorithm may be created. Once the algorithm stops, we could add to the LUV object distribution the meanshifted values for any pixels within the boundary of the object. Then we restart the process using the new centroid and the new ellipse recalculated using these additional values. However, all these additions are subject of future work.

REFERENCES

Barthel K.U. (2007) *3D Color Inspector/Color Histogram*, available at http://rsb.info.nih.gov/ij/plugins/color-inspector.html

Cheng, Y. (1995, Aug.). Mean Shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*(8), 790-799.

Comaneciu D. and P. Meer (2002, May). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24(*5), 603-619.

Georgescu B., I. Shimshoni, P. Meer (2003), Mean Shift Based Clustering in High Dimensions: A Texture Classification Example, *Ninth IEEE International Conference on Computer Vision (ICCV'03)* – vol. 1, pp.456

Kamberov, G., G. Kamberova and A. Jain (2005), 3D shape from unorganized 3D point clouds, *Intl. Symposium on Visual Computing, Lake Tahoe, NV,* Springer Lecture Notes in Computer Science, vol. 3804, pp.621-629.

Martin D., C. Fowlkes, D. Tal, and J. Malik (2001), A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, *Proc. 8th Int'l Conf. Computer Vision*, July, vol. 2, pp 416-423.

Millet C., I Bloch, P. Hède, and P-A. Moëllic (2010), Automatic cleaning and segmentation of web images based on colors to build learning databases, *Image and Vision Computing*, vol. 28(3), pp. 317-328

Nummiaro, K., Koller-Meier, E., and Gool, L. V. (2002) Object tracking with an adaptive color-based particle filter. *Pattern Recognition: 24th DAGM Symposium, Preceedings*, Volume 2449 of *Lecture Notes in Computer Science*, Zurich, Switzerland, pp. 353-601.

Stolkin R., I. Florescu, M. Baron, C. Harrier and B. Kocharov (2008) Efficient visual servoing with the ABCshift tracking algorithm, *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, 19-23 May 2008, pp. 3219-3224

Stolkin R., I. Florescu, G. Kamberov (2007) An adaptive background model for CAMSHIFT tracking with a moving camera, Advances In Pattern Recognition, *Proceedings of the Sixth International Conference Indian Statistical Institute*, Kolkata, India 2-4 January 2007, World Scientific Publishing, pp. 147-151.

West D. H. (1975) Updating mean and variance estimates: an improved method, *Commun. ACM*, vol. 22, nr. 9, Sep. 1979, pp. 532-535.

Wang, L., Hu, W., and Tan, T., (2003, March). Recent developments in human motion analysis. *Pattern Recognition 36*(3), pp 585-601.

Table 1: ANOVA results for all the objects. The variables are: X_1=*Initbox*, $X_2=h_c$, $X_3=h_r$, $X_4=\alpha$. A checkmark in each respective column signifies that the respective term was significant at the 0.05 level for the *Y* variable (significant reduction in error rate). A dash in each entry signifies a non-significant term for *Y*.

Img	Obj	X1	X2	X3	X4	X1 * X2	X1 * X3	X1 * X4	X2 * X3	X2 * X4	X3 * X4	X1 * X2 * X3	X1 * X2 * X4	X1 * X3 * X4	X2 * X3 * X4	X1 * X2 * X3 * X4
1	1	X	X	X	X	X	X	X	-	X	X	-	-	-	-	-
1	2	X	X	X	X	X	-	X	X	X	X	-	-	-	-	-
1	3	X	X	X	-	X	X	X	-	X	-	-	-	-	-	-
1	4	X	X	X	X	X	X	X	-	X	-	-	-	-	-	-
1	5	X	X	X	X	X	X	X	X	X	X	-	X	-	-	-
2	1	X	X	X	X	-	X	X	X	X	X	-	-	-	-	-
2	2	X	X	X	X	-	X	-	-	X	X	-	-	-	-	-
2	3	X	X	X	X	-	X	-	X	-	-	-	-	-	-	-
2	4	X	X	X	X	X	-	-	X	X	-	-	-	-	-	-
2	5	X	X	X	X	X	X	X	-	X	X	-	-	-	-	-
3	1	X	X	X	X	X	X	-	X	X	X	-	-	X	X	-
3	2	X	X	X	X	-	X	X	X	X	X	-	-	-	-	-
3	3	X	X	X	X	X	X	X	X	X	X	-	X	-	X	-
3	4	X	X	X	X	X	-	X	X	X	X	-	-	-	-	-
4	1	X	-	X	X	-	-	X	-	-	-	-	-	-	-	-
4	2	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-
4	3	X	X	X	X	X	X	X	X	-	X	-	-	-	-	-
4	4	X	-	-	X	X	X	X	-	-	-	-	-	-	-	-
5	1	X	X	X	X	X	X	X	X	X	X	X	X	X	-	-
5	2	X	X	-	X	X	X	X	-	X	-	-	-	-	-	-
5	3	X	X	X	X	X	X	X	X	X	-	-	X	-	-	-
5	4	X	X	X	X	X	X	X	X	X	X	X	X	X	-	-
5	5	X	X	X	-	X	-	X	-	-	-	-	-	-	-	-

Table 2: The number of optimal parameter choices for each object studied.

Image 1	Optimal choices	Image 2	Optimal choices	Image 3	Optimal choices	Image 4	Optimal choices	Image 5	Optimal choices
Obj. 1	188	Obj. 1	465	Obj. 1	12	Obj. 1	100	Obj. 1	227
Obj. 2	107	Obj. 2	555	Obj. 2	165	Obj. 2	12	Obj. 2	319
Obj. 3	198	Obj. 3	228	Obj. 3	149	Obj. 3	372	Obj. 3	105
Obj. 4	335	Obj. 4	394	Obj. 4	600	Obj. 4	434	Obj. 4	139
Obj. 5	248	Obj. 5	36					Obj. 5	320

Table 3: Optimal parameter values for all the objects in an image.

	$(h_r, Initbox, h_c, \alpha)$	$(h_r, Initbox, h_c, \alpha)$	$(h_r, Initbox, h_c, \alpha)$	$(h_r, Initbox, h_c, \alpha)$
Image 1	(9, 11, 3, 95%)			
Image 2	(3, 7, 10, 99%) (3, 7, 20, 99%) (3, 7, 50, 95%) (3, 7, 10, 97%)	(5, 7, 1, 99%) (5, 7, 10, 99%) (5, 7, 20, 99%) (5, 7, 50, 97%)	(9, 7, 50, 97%)	(11, 9, 20, 95%) (11, 9, 20, 97%)
Image 3	-			
Image 4	(3, 3, 50, 99%)	(3, 5, 20, 99%) (3, 5, 50, 97%)	(5, 3, 50, 99%)	
Image 5	(3, 3, 10, 97%)	(9, 3, 10, 99%)	(9, 5, 5, 95%) (9, 5, 5, 97%) (9, 5, 5, 99%)	(11, 3, 10, 99%) (11, 7, 4, 95%) (11, 7, 4, 97%)

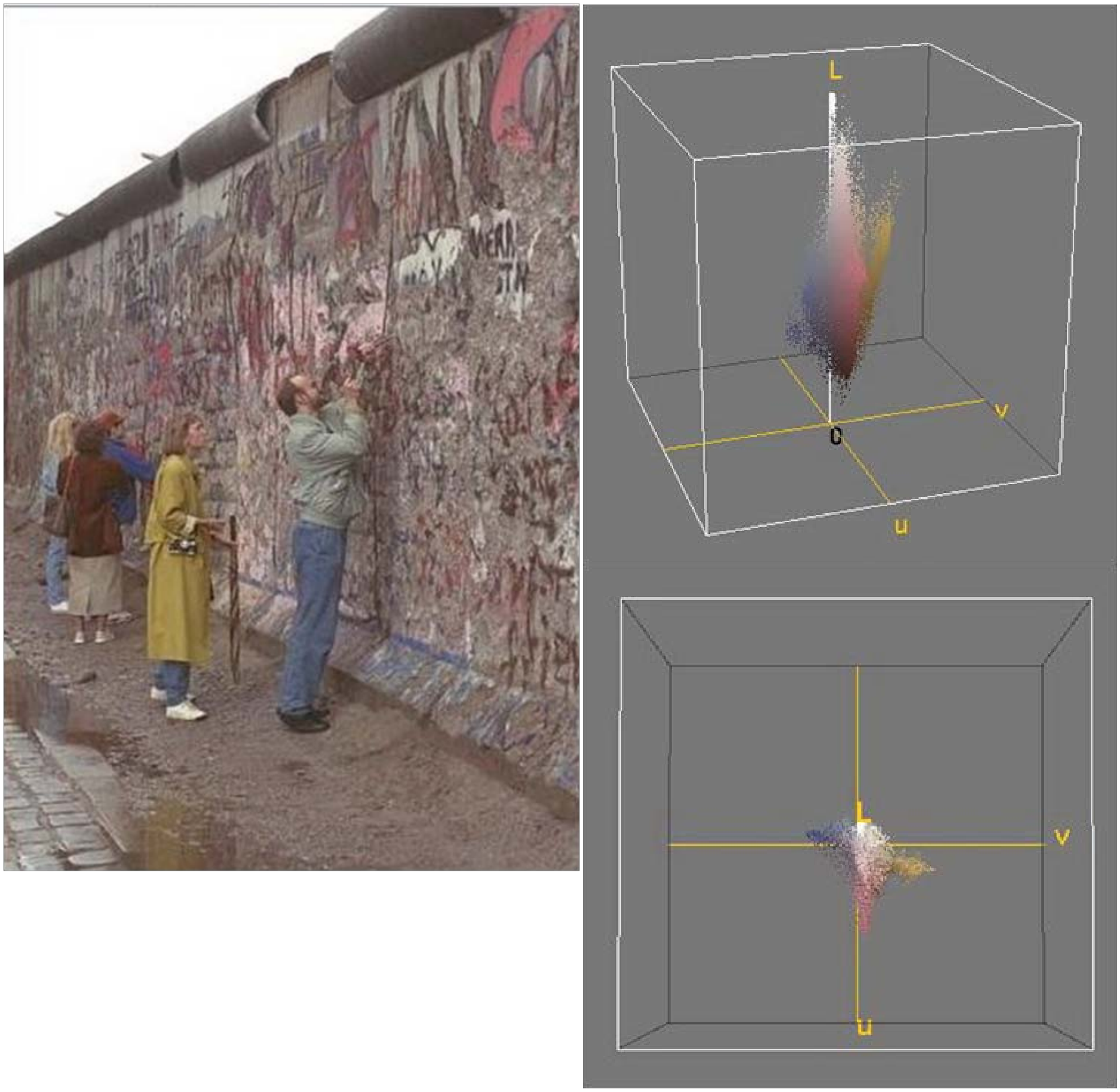

Figure 1. Image 1 and the original LUV space representation. Situation: small objects in cluttered background. The five selected objects are: 1- blue pants for man, 2-blouse for man, 3 – yellow trench for older woman, 4- brown coat and 5 - skirt for faraway woman. The LUV

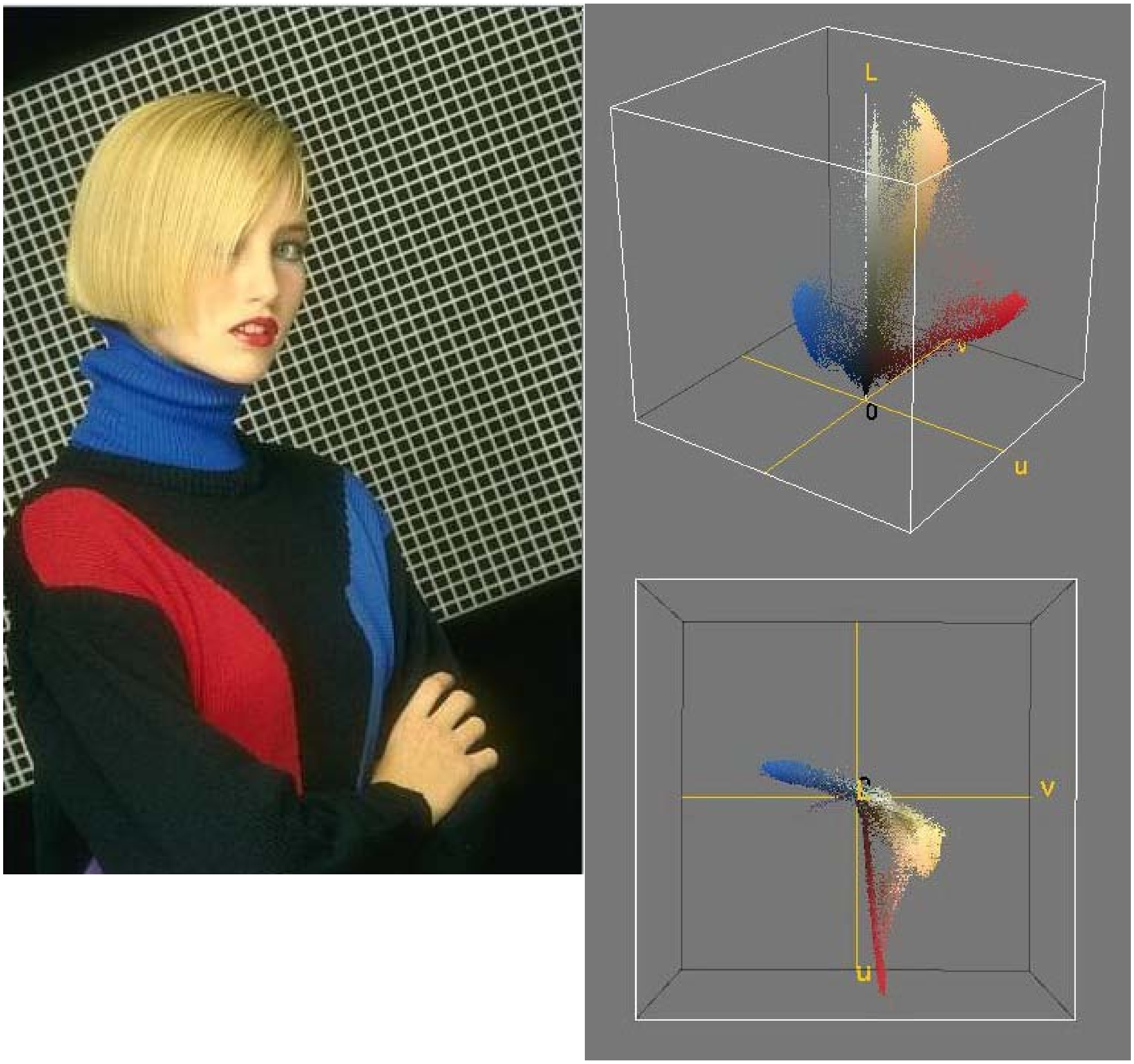

Figure 2. Image 2. Situation: Large objects homogeneous in color. No distraction from background. The five selected objects are: 1- right red patch on pullover, 2-blonde hair, 3-blue pullover neck, 4-hand, and 5-left blue patch on the pullover. The LUV space is very wide and non-compact.

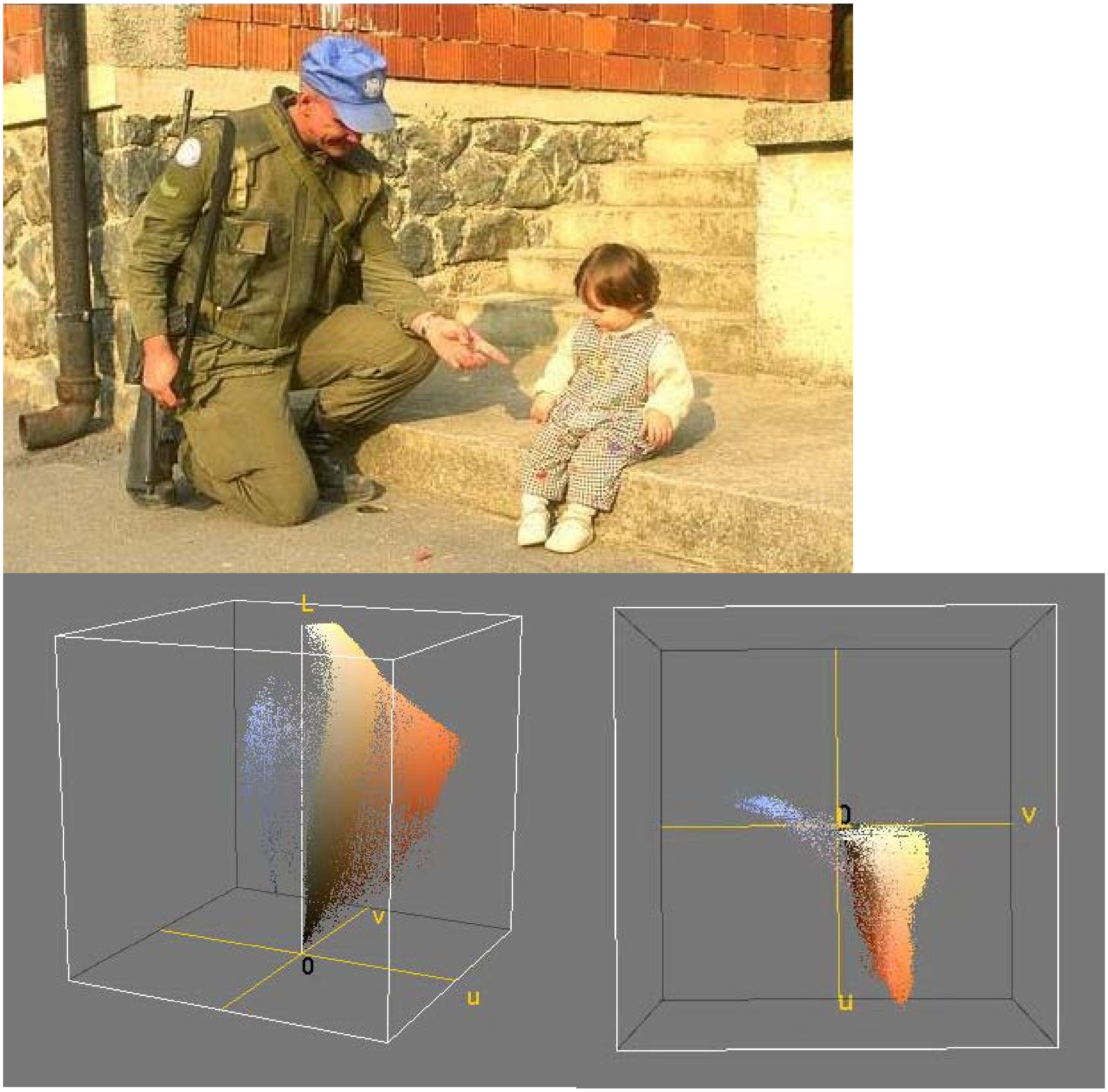

Figure 3. Image 3. Differently sized objects on cluttered background. The four selected objects are 1-child one piece suit, 2 - child head, 3- soldier hat and 4 - soldier uniform. The LUV

Figure 4. Image 4. Similarly sized objects on distinct background. The four selected objects are: 1- blouse and 2 - pants for the viewer in the back; 3- blouse and 4 - pants for the golf player. The LUV space is very compact and elongated and the object colors represent a small fraction of the total space.

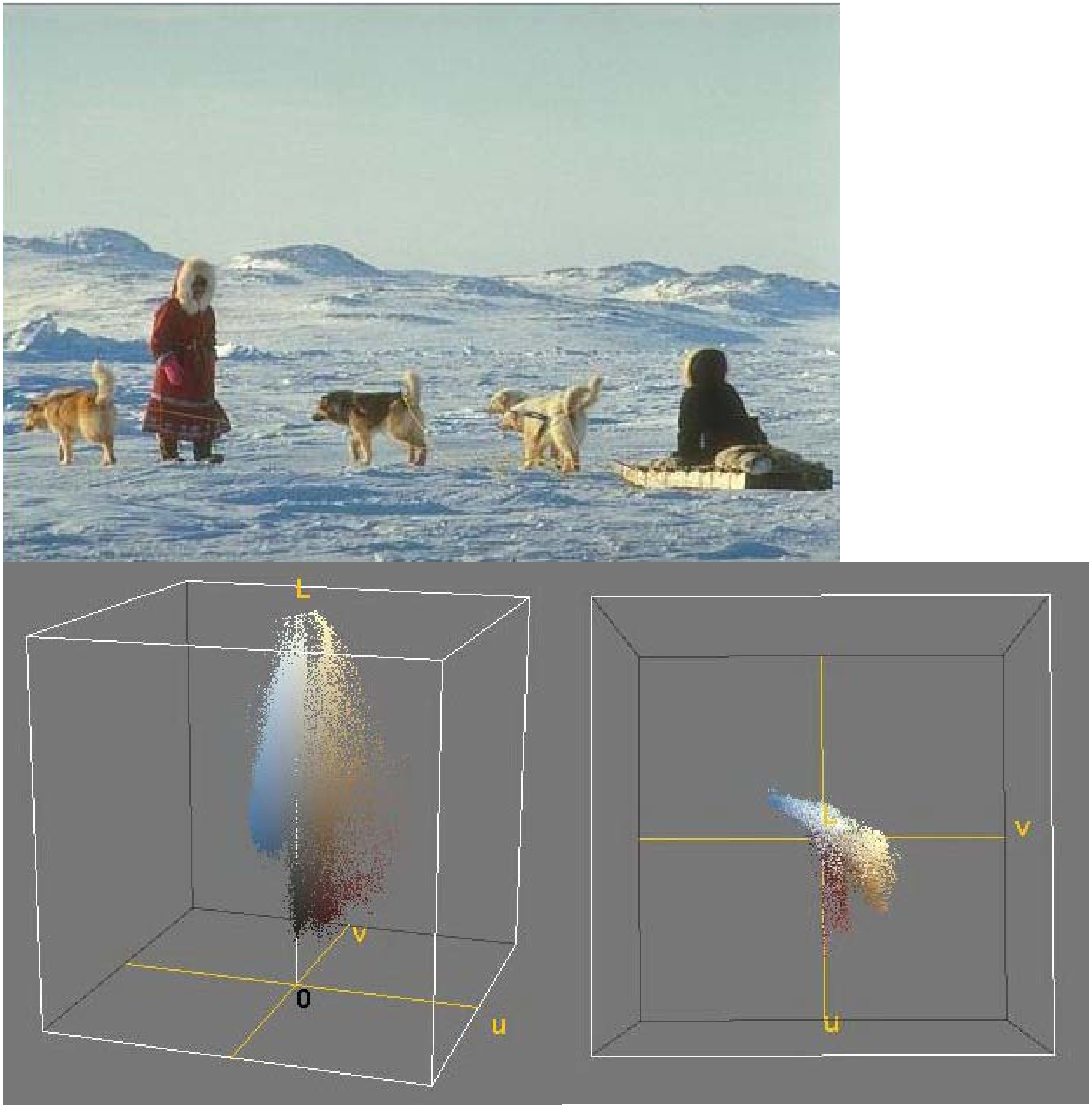

Figure 5. Image 5. Small objects on clear background. The five selected objects are: 1- leftmost fox colored dog, 2- Eskimo in red dress, 3- brown dog, 4- the two white dogs, 5- Eskimo in black dress. The LUV space is wide and relatively non-compact.

Figure 6. The segmentation results using the same parameter values for all the objects in the picture, applied to the first 3 images. Note that the soldier image 3 does not have an optimal set, we chose one set of values that gave small errors for objects 1, 2 and 3.

Figure 7. The segmentation results using the same parameter values for all the objects in the picture, applied to the images 4 and 5.