Fuzzy C-Means Stereo Segmentation

Michal Krumnikl(✉), Eduard Sojka, and Jan Gaura

Faculty of Electrical Engineering and Computer Science,
Department of Computer Science, VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{michal.krumnikl,eduard.sojka,jan.gaura}@vsb.cz
http://www.cs.vsb.cz/

Abstract. An extension to the popular fuzzy c-means clustering method is proposed by introducing an additional disparity cue. The creation of the fuzzy clusters is driven by a degree of the stereo match and thus it enables to separate the objects not only by their different colours but also on their different spatial depth. In contrast to the other approaches, the clustering is not performed on the individual input images, but on the stereo image pairs and takes into accounts the stereo matching properties known from the stereo matching algorithms. The proposed method is capable of calculating the output segmentations, as well as the disparity maps. The results of the algorithm show that the proposed method can improve the segmentation in difficult settings. However, the drawback of this approach is that it requires the stereo image pairs of the segmented scenes that are not always easily obtainable.

Keywords: Fuzzy c-means · Segmentation · Stereo matching · Disparity

1 Introduction

In this paper we would like to propose an extension to the popular fuzzy c-means clustering method by introducing an additional disparity cue. The reason for introducing the additional cues is to improve the segmentation. This can be achieved by using the following approach, but only under the condition of having the stereo image pair of the segmented scene. Beside the segmentation with the additional depth constraints, our method is also capable of producing the disparity map of the input image pair and hence can be also considered as a form of the stereo matching algorithm.

The following text describes the adaptation of the fuzzy c-means algorithm to perform the clustering in space extended by the dimension of the disparity. The creation of the clusters will be driven by a degree of the stereo match (this measure will be described later on). An attractive aspect of this strategy is that we are able to take advantage of known number of depth levels or objects (if this information is available).

A. Fred et al. (Eds.): ICPRAM 2014, LNCS 9443, pp. 35–49, 2015.
DOI: 10.1007/978-3-319-25530-9_3

The motivation for our work was to provide an algorithm that can separate objects based on their different colour and spatial depth. We regard this method as more suitable in specific cases (will be described later on) than the segmentation with the final disparity maps of the stereo matching algorithms. The distance, based on both dissimilarities (spatial and colour), provide more sensitive segmentation (especially on segment borders) than the segmentation performed on the filtered disparity maps which contain only the best matches, and do not take into account segment properties. The algorithm was originally developed for a very specific purpose – the segmentation of the moss clusters (as a part of a biological research involving these species). Therefore, we have tested and evaluated the algorithm mainly on the “Map” dataset, introduced in [26], as it strongly resembles the stone structures which are frequently covered by the moss layers. However, as we will show in the next paragraphs, the algorithm can be used in more general cases. The main domain of application of our algorithm is defined by the following constraints:

- The images should contain relatively small number of segments.
- The segments should be preferably planes or linear gradients.
- There should be no or minimal occlusions.

The clustering technique is usually described as a process of forming partitions from a data set on the basis of a performance function, also known as an objective function. The underlying idea of our algorithm is to consider the disparity space (e.g., in disparity maps) as a specific type of the data set, consisting of clusters representing the three dimensional objects of the scene. The fuzzy c-means algorithm has already been used to create the segmentations based on the depth information or disparity maps, e.g., [1,22], and was also adapted to incorporated the spatial neighbourhood information, e.g., [7,17,19], but in these approaches, the algorithms were run on the input data already containing the depth information. In contrast, the proposed algorithm does not need the depth information in advance, since it calculates it itself by means of the stereo matching.

The stereo matching problem itself is a multicriterion decision problem. The most common classification of the stereo matching algorithms is based on the size of the processed area. In this way, we recognize the local and global methods. In the local matching methods, the correspondence of a pixel is based on the similarity of its neighbourhood. The similarity itself can be computed using the measures such as the sum of the absolute differences (SAD), sum of the squared difference (SSD), normalized SSD, normalized cross-correlation etc. A comparison of the different similarity measures can be found in [9]. The global methods usually tend to minimize an energy function, e.g., by using the dynamic programming [8,30], graph cuts [6,16], Markov random fields [5] or belief propagation with segmentation [15].

The problem has been also solved by the fuzzy aggregation operators [31] or fuzzy relaxation technique [23]. The last method improves the matching in case of partially occluded objects. In [3], a fuzzy integral was introduced to improve the results obtained with the classical fuzzy averaging operators. The basic idea of using the clustering technique together with the stereo matching process was introduced in [4,28] and further developed in [18,27,29,32]. Compared to these, our approach differs in several aspects. The clustering is not performed on the individual input images, but on both stereo images simultaneously, and takes into account the matching properties. In each step, the clusters are adjusted to minimize the matching cost.

The paper is organized as follows. In Sect. 2 we briefly introduce the classical fuzzy c-means algorithm. Then, the extension of the fuzzy c-means is described in Sect. 3 in order to provide the depth segmentation based on the differences of the two stereo images. The experimental validation and the benchmark results are provided in Sect. 4. Finally, conclusions are presented in Sect. 5.

2 Fuzzy C-Means

Let us briefly introduce the original method. Fuzzy c-means is a widely used clustering technique, developed by [10] and improved by [2]. It is based on a standard least squared error model that generalizes an earlier and popular non-fuzzy c-means mode [20]. Fuzzy c-means can be generalized in many ways to include, e.g., Minkowski, Hamming, Canberrar or hybrid distances.

The fuzzy c-means algorithm attempts to partition a collection of n data points $\{x_k\}_{k=1}^{n}$ into a collection of c fuzzy clusters (represented by the cluster centres) on the basis of a distance d between the cluster centre and the data point. The algorithm is minimizing the objective function $J(U, V)$, where $V = (v_1, \dots, v_c)$ is the set of cluster centres and $U = [u_{ki}]$ is the $n \times c$ membership matrix. The space of all possible values of U is denoted as U_f. The elements of the matrix U are organized as follows. The column i gives the membership of all n input data points (rows) in the cluster i for $i = 1 \dots c$. The u_{ki} stands for the membership of the k-th point of the i-th cluster. The idea is that the closer the data point is to the cluster centre, the larger is its membership value towards that specific cluster. Consequently, the sum of all memberships of the data point across all clusters is equal to one. The fuzzy membership is formally given by the following constraint

$$\begin{aligned} U_f = \{U = (u_{ki}) : \sum_{j=1}^{c} u_{kj} = 1, 1 \leq k \leq n; \\ u_{ki} \in [0,1], 1 \leq k \leq n, 1 \leq i \leq c\}. \end{aligned} \tag{1}$$

The minimized objective function $J(U, V)$ is defined as [2]

$$J(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ki})^m d(x_k, v_i), (1 \leq m \leq \infty), \tag{2}$$

where u_{ki} is a degree of membership of x_k in the cluster i, and v_i represents the centre of the cluster. The parameter m is called the weighting exponent of the model. For $m = 1$, the memberships converge to 0 or 1, producing a crisp partitioning. The best choice for m is probably in the interval $[1.5, 2.5]$, where $m = 2$ is the most common choice [24]. The distance $d(x_k, v_i)$ represents (usually) Euclidean distance between the k-th data point and the i-th cluster centre.

We should notice that the minimization of the objective function $J(U, V)$ is not an exact minimization but an iteration procedure of so called "alternate minimization". In essence, the algorithm is searching for a local optimal solution, which we will denote with stripe (e.g., $\bar{U}$). The overall iterative process may be summarised as follows.

Algorithm Steps

1. Initialize the matrix U by randomly generated u_{ki} membership coefficients for all cluster centres $\bar{V} = (\bar{v}_1, \ldots, \bar{v}_c)$.
2. Find the optimal U by iteratively calculating $\bar{U} = \arg\min_{U \in U_f} J(U, \bar{V})$. The following solution can be derived using the Lagrange multiplier method [20]

$$\bar{u}_{ki} = \left[\sum_{j=1}^{c} \left(\frac{d(x_k, \bar{v}_i)}{d(x_k, \bar{v}_j)} \right)^{\frac{2}{m-1}} \right]^{-1}, (x_k \neq v_i). \tag{3}$$

   The solution for $(x_k = v_i)$ is obviously $\bar{u}_{ki} = 1$.
3. Find the optimal V by calculating $\bar{V} = \arg\min_V J(\bar{U}, V)$. The solution is computed by differentiating J with respect to V [20]:

$$\bar{v}_i = \frac{\displaystyle\sum_{k=1}^{n} (\bar{u}_{ki})^m x_k}{\displaystyle\sum_{k=1}^{n} (\bar{u}_{ki})^m}. \tag{4}$$

4. Repeat from step 2 until $\bar{U}$ and $\bar{V}$ is convergent.

The convergence is achieved when $\max_{k,i} |\bar{u}_{ki} - u_{ki}| < \epsilon$, where $\bar{u}$ is the new solution, u is the value from the previous iteration and ϵ is a small positive number, the threshold. Alternatively, we can use $\max_{1 \leq i \leq c} \|\bar{v}_i - v_i\| < \epsilon$ as a convergence condition.

3 Introducing the Matching Constraint to Fuzzy C-Means

In a simplified way, we can say that the original fuzzy c-means algorithm (when used in image processing) is usually based only on the pixel positions and their intensities (colours). In our approach, we have extended this algorithm to include the matching constraints. First, by expanding the dimension of the data vector to include the disparity (depth), and then, by evaluating the dissimilarity of the stereo pair (which will be explained later).

As stated in Sect. 2, the algorithm attempts to partition the elements with respect to a given criterion, defined as a degree of belonging that is related inversely to the distance. However, for the depth segmentation, we need to add additional components measuring the intensity (colour) difference between the point and its supposed projection and the distance between the point disparity and the disparity of its supposed cluster. The sources of the spatial information are the small differences in the stereo images. In that way, we associate the clusters with the disparity space. Therefore, we have to define the vector of the cluster centre as

$$v = (v_X, v_Y, v_I, v_D), \tag{5}$$

where v_X,v_Y stand for the spatial position, v_I for the brightness and v_D for the disparity value. For the clarity, the capitalized subscripts, X, Y, I and D, are used to indicate the vector elements (e.g., v_X). The small subscripts will later be used to specify a particular vector from the set (e.g., x_k).

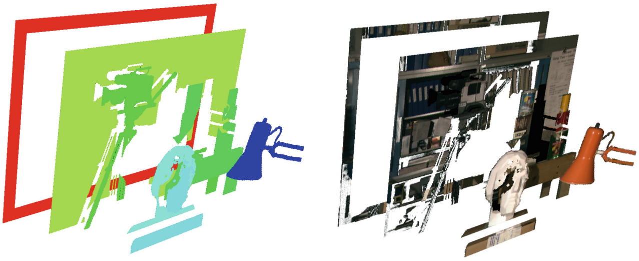


Fig. 1. Illustration of the rationale behind the algorithm. The left figure shows the coloured disparity levels of the dataset [25], while the right one depicts the original pixel colours. The algorithm is based on the observation that the objects share the similar disparity, as well as similar colour. This can be clearly seen on the red lamp in the foreground or the white statue on the left (Color figure online).

Our new membership function takes into account the dissimilarity of the left image pixel ($\phi_L(x_X, x_Y)$) and the right image pixel shifted by the average cluster disparity ($\phi_R(x_X + v_D, x_Y)$). Basically, we use the disparity in the similar fashion as the intensity, grouping the pixels sharing the same, or almost the same disparity value (see Fig. 1). For this, we need to adapt the membership function to penalize the pixels having the incorrect match (not similar to their projections on the other image) and provide the way of measuring the distance between the cluster centres and pixels with associated disparity value.

We propose the use of the extended vector space model with the additional dimensions reflecting the disparity and pixel dissimilarity in the stereo image pair. The distance in the proposed vector space is, for clarity, separated into the two components (d and d_s), described later on.

The proposed fuzzy stereo partitioning is carried out using the following membership function (the subscripts k, i, j are the indexes)

$$\bar{u}_{ki} = \left[\sum_{j=1}^{c} \left(\frac{d^2(x_k, \bar{v}_i) + d_{\mathrm{s}}^2(x_k, \bar{v}_i)}{d^2(x_k, \bar{v}_j) + d_{\mathrm{s}}^2(x_k, \bar{v}_j)} \right)^{\frac{1}{m-1}} \right]^{-1}, (x_k \neq v_i). \tag{6}$$

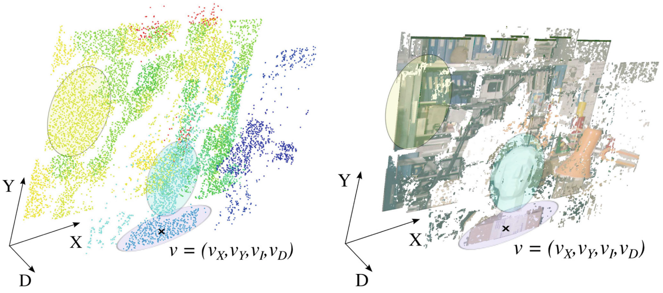


Fig. 2. Visualisation of the data points and their clusters taken from our experiments. The points on the left figure are coloured according to the disparity levels associated with them. The right figure shows their real colour. The both figures shows the depth levels as obtained from the calculations of the proposed modification of the fuzzy c-means algorithm (Color figure online).

The membership $\bar{u}_{ki}$ is related inversely to the distance between the processed point and the cluster centre (as calculated in the previous iteration). The new term d_{s} reflects the correctness of the stereo match between the pixel of the left (ϕ_L) and its projection on the right (ϕ_R) image (the subscripts X, Y, D denotes the vector elements):

$$d_{\mathrm{s}}^2(x, v) = \lambda_m (\phi_L(x_X, x_Y) - \phi_R(x_X + v_D, x_Y))^2, \tag{7}$$

where x is the data point (vector) and v is the cluster centroid. The uppercase subscript of the vector denotes its component. The constant λ_m stands for the weight of the matching term. For ϕ_L and ϕ_R we assume the rectified images. It is possible to replace the difference $\phi_L(x_X, x_Y) - \phi_R(x_X + v_D, x_Y)$ by the difference of the aggregating windows (SAD, SSD, etc.), but as the aggregation of the pixels is inherently given by the fuzzy c-means, it does not provide any further advantage and even worsens the results by blurring the edges. The distance $d(x, v)$ is calculated (as in original method) using the Euclidean distance:

$$\begin{aligned} d^2(x, v) = \lambda_i (x_I - v_I)^2 + \lambda_d (x_D - v_D)^2 + \\ \lambda_s (x_X - v_X)^2 + \lambda_s (x_Y - v_Y)^2, \end{aligned} \tag{8}$$

where x_X, x_Y are the pixel coordinates, x_I colour intensity and x_D is the disparity value. When compared to the original method, we have added the term measuring the disparity distance of the processed point and the cluster (see Fig. 2). For the pixel disparity x_D we can take an initial guess since, as we will show later, the algorithm is quite insensitive to this value. Basically, it only helps in the beginning to form the initial clusters. The values λ_i, λ_d and λ_s denote the intensity, disparity and spatial weights. The effects of these weights are discussed with results (Sect. 4).

The iteration steps remain the same as in Sect. 2. The outline of the algorithm can be summarized as follows: (i) choose the proper parameters, especially the number of clusters (discussed in Sect. 3.1), (ii) to each point assign random cluster membership coefficients, (iii) in each iteration compute the centroid for each cluster (Eq. 4), followed by the computation of the membership coefficients for all points (Eq. 6). Repeat this step until the algorithm has converged. Finally, create the output disparity map based on the cluster disparities (iv).

The algorithm was tested on several types of real images (depicting the processed botanical samples) and also on the standard dataset used for the evaluation of the stereo matching algorithms [25]. While our approach is not intended to be used as the general purpose stereo matching algorithm, we would like to give the reader an opportunity to examine the results in the standard stereo matching benchmark tests (see Sect. 4).

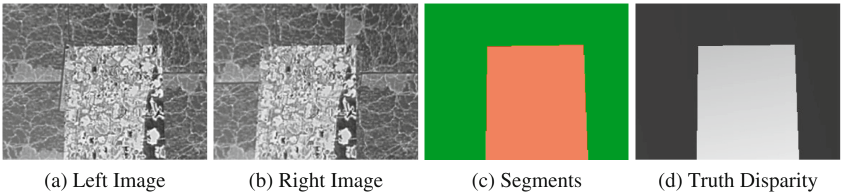

(a) Left Image (b) Right Image (c) Segments (d) Truth Disparity

Fig. 3. The reference images (a, b), the ground truth disparity map (d), and its segments (c) used for the demonstration of the cluster count problem.

3.1 Cluster Count Problem

The disadvantage of the fuzzy c-means (as well as k-means) is the result dependency on the initial choice of weights. This is also true for our method. Despite the algorithm minimizes the intra-cluster variance, calculated minimum is still only a local minimum. But more serious problem of the fuzzy c-means algorithm is that it requires the number of clusters to be known in advance.

The correct choice of the cluster count is ambiguous, with interpretations depending on the shape and scale of the data point distribution in the input data set and the desired resolution. This may seem as a disadvantage for general settings, but may be an advantage for special cases, where the number of segments or number of disparity planes is already known. For example, in Fig. 3

the box is the only object in the foreground, and can be easily represented by only a small number of segments. As you can see (Fig. 4), with only a few clusters, we are able to acquire very precise disparity map and by increasing the number of the segments, we are able to capture even smaller changes in the disparity gradient (the box in the example is slightly tilted). We can say that by choosing the number of clusters, we can set, whether we are more interested in large segments covering the whole objects, or small fine-grained parts.

The results of our approach surpass (but only for the specific types of scenes, similar to the sample images) the performance of the majority of the standard state-of-the-art algorithms (see Table 1, "Map" column). However, due to the algorithm specialization, it is less suitable for the other types of scenes. But still, the additional cue improves the segmentation results.

4 Tests and Results

This section describes the experiments and shows the results confirming the anticipated segmentation features and proper depth discrimination.

First, we have performed the tests on the images fulfilling the assumptions, we made in the beginning – the scenes with only a few objects, each having almost the flat depth. The "Map" dataset (Fig. 4) complies with these requirements. The results for this specific dataset are very satisfactory (Table 1, column "Map"); however, the results for the other types of image pairs (from the dataset) are not very encouraging. We do not consider this as a disadvantage, since the intentions of this algorithm are different than the general purpose stereo matching algorithms. The explanation for the results on the other samples is that these pairs violate the initial presumptions of our algorithm; the scenes contain a lot of objects with fine-grained disparity. The limits of our algorithm – the number of clusters and plane disparities – do not offer many opportunities for improvements in such general cases.

To illustrate the algorithm performance on the images with optimal object configurations, we have chosen several samples from the Adobe Open Source Data Sets[1]. The data set contains stereo images and ground truth segmentation of the foreground object. The results of the selected images are visible in Fig. 6. We have to point out that these images illustrate the optimal cases.

Nevertheless, we have also performed the tests on the images that are not very suitable for our approach. The absolute results with the comparison of the other algorithms are shown in Table 1. The evaluation has been performed on the Middlebury dataset [25]. The full list of algorithms is available on the Middlebury stereo vision website. While our algorithm is not the typical stereo matching algorithm, due to the lack of more suitable, generally accepted dataset for segmenting the stereo images, we decided to perform the tests on these images. The parameters were maintained the same for all images – cluster count $n = 200$, $\lambda_d = 1.0$, $\lambda_s = 0.1$, $\lambda_i = 0.05$, and $\lambda_m = 0.1$.

[1] http://sourceforge.net/adobe/adobedatasets/.

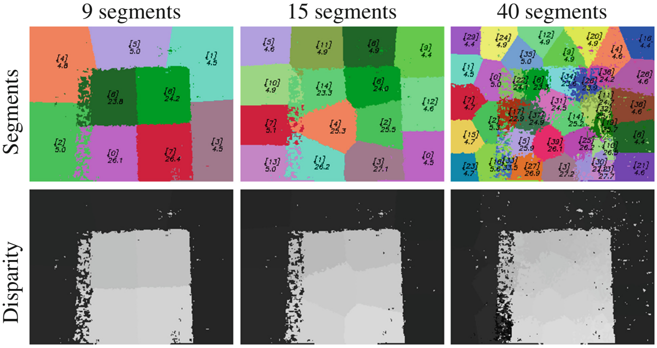


Fig. 4. The influence of the cluster count on the output disparity map. For better reading, the segments are coloured, numbered (number in brackets), and marked with their disparity values (the value below the number in brackets). The reference images are depicted in Fig. 3. The subfigures show the results of the modified fuzzy c-means algorithm set to 9, 15 and 40 segments. The top subfigures show the segments, while the bottom ones show the disparity maps obtained from the segments disparity values (Color figure online).

Table 1. The performance of the modified fuzzy c-means algorithms according the Middlebury stereo test bed [25]. The overall performance is measured by the percentage of bad pixels in the non-occluded areas (nocc). The performance measured on the whole image (all) is provided as well. Our algorithm is denoted as FZ. The total cluster count was set to 200. In order to give a better idea of the performance of our methods compared to the state-of-the-art algorithms, we have included the results of the selected algorithms from the Middlebury evaluation.

	Tsukuba		Venus		Teddy		Cones		Map		
Algorithm	nocc	all	nocc	all	nocc	all	nocc	all	nocc	all	avg
[12]	2.61	3.29	0.25	0.57	5.14	11.8	2.77	8.35	1.09	2.82	5.33
[15]	0.97	1.75	0.16	0.33	6.47	10.7	4.79	10.7	3.39	5.79	5.85
[11]	3.26	3.96	1.00	1.57	6.02	12.2	3.06	9.75	1.12	2.97	6.09
[14]	1.94	4.12	1.79	3.44	16.5	25.0	7.70	18.2	0.74	6.82	11.51
[8]	4.12	5.04	10.1	11.0	14.0	21.6	10.5	19.1	6.04	12.12	13.77
SSD	5.23	7.07	3.74	5.16	16.5	24.8	10.6	19.8	8.49	14.57	14.28
FZ	12.7	14.3	12.5	13.5	32.3	37.8	32.0	36.9	**0.72**	**7.17**	21.93

The proposed algorithm converges approximately after 15 iterations on all images of the given set. The outputs with 100 segments are displayed in Fig. 7 (evaluated outputs with 200 segments were not used for the illustration purposes, due to the hard distinguishability of the small clusters). The images in the upper row show the segments. The disparity maps obtained from the segment properties are displayed below. As you can see, the proposed algorithm is

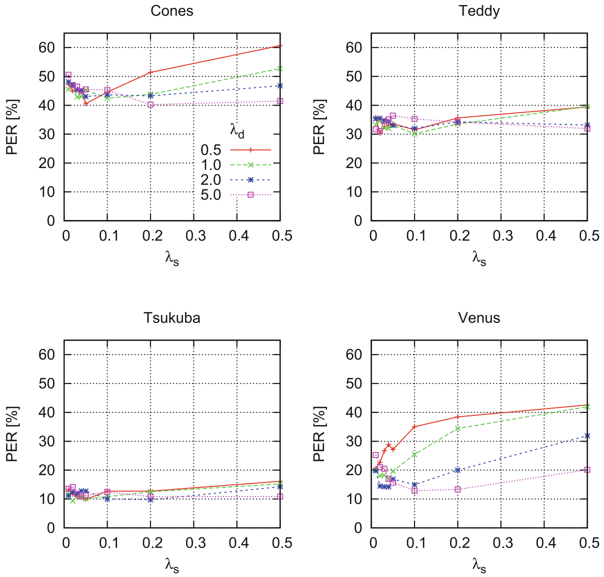


Fig. 5. The algorithm results achieved with different disparity and spatial weights (λ_d and λ_s). The algorithm was set to generate 100 segments. The different disparity weights (λ_d) are represented by the different line colours. The significant effects of the disparity weight (λ_d) can be seen only on the images containing the planar objects (e.g., the "Venus" pair) (Color figure online).

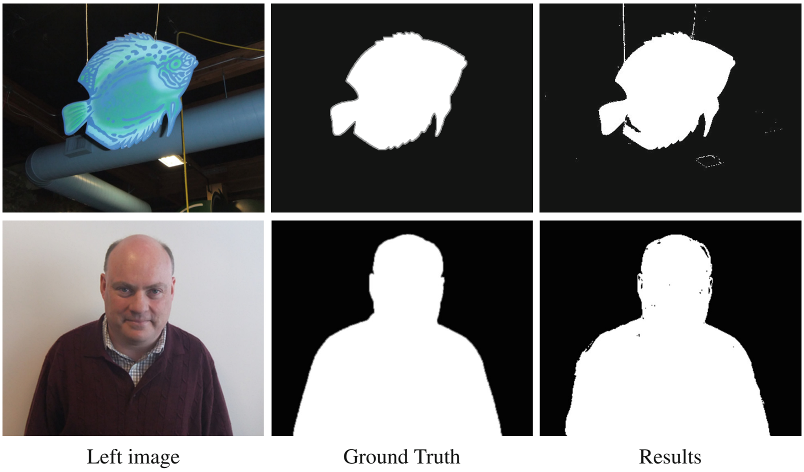


Fig. 6. Segmentation results of two samples from the Adobe Open Source Data Sets. These samples illustrate the ideal configurations for the proposed algorithm – raised flat foreground objects.

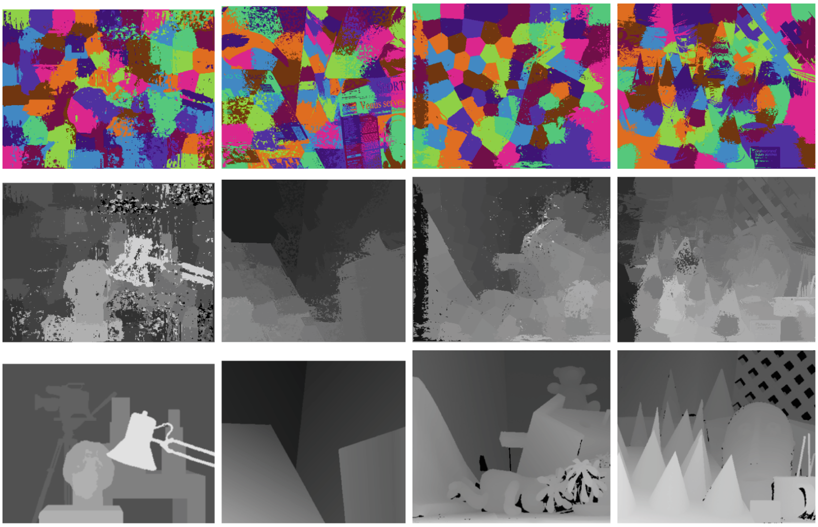
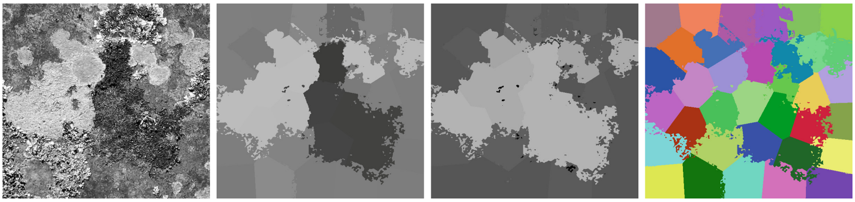


Fig. 7. The images show the disparity and cluster maps obtained for the default Middlebury dataset using our expanded fuzzy c-means algorithm (fz). The ground truth data are provided in the last row. As you can see, the output disparity maps are not as good as the results from the "Map" dataset (Fig. 4). The reason is that the Middlebury dataset contains images with a lot of details and a set of various objects, which contradicts the initial algorithm assumptions. In order to improve the performance it is necessary to significantly increase the number of the clusters, which consequently leads to a much longer processing time. Unfortunately, this still does not guarantee for all inputs the results comparable to the best algorithms.

capable of obtaining the disparity maps of more sophisticated scenes, but not at the level of detail as the generally used stereo matching approaches.

To increase the overall performance, it is possible to increase the number of clusters, which in result leads to a more grained segmentation, where each segment can have different disparity. The drawback of a huge number of clusters is the increasing computational time. At the certain level, the additional increasing of cluster count starts to be inefficient. We have used no more than 200 segments.

During the development, we have also performed several experiments to investigate the effects of the algorithm parameters on the segmentation performance. The parameter settings may vary from scenario to scenario, but generally, only two parameters appear to be particularly influential - the spatial and disparity weight (λ_s and λ_d). Figure 5 shows the influence of these weights on the output segmentation consisting of 100 segments. The experiment showed the significant effect of the disparity weight (λ_d) mainly on the images containing the planar objects (e.g., the "Venus" pair). This is a predicted behaviour as our algorithm favours planar disparities. On an example of "Venus" pair, you can see that the increasing disparity weight forces the algorithm to create segments with less disparity deviations from the cluster centroid, leading to the better results.



(a) Source Image (b) Segm. Brightness (c) Segm. Disparity (d) Output Segments

Fig. 8. The segmentation results of the moss sample using the modified fuzzy c-means algorithm (Sect. 3). The figure shows (from left to right): the left image of the input pair depicting the moss layers on the stone base, the segments coloured according the average colour, the segments coloured according the disparity and the visualization of the clusters itself. As you can see, our modification of the fuzzy c-means still retains the properties of the original algorithm and in addition provides the disparity values (Color figure online).

However, for images not containing such objects (e.g., “Teddy” or “Tsukuba”) the change in these parameters has only a small impact on the results. We have not evaluated all possible parameter configurations for all dataset images, but empirically, we can say that the best results were achieved with $\lambda_s = 0.1$. Increasing this value forced the algorithm to create too compact clusters and, vice versa, decreasing λ_s caused merging too distant pixels into one cluster.

In the application that the algorithm was originally developed for, it was important to separate the layer of the base (usually the stone) and the layer above, formed by the moss structures. An example is illustrated in Fig. 8. As you can see, the resulting segmentation strongly benefits from the inherit features of the algorithm. The design of the algorithm was strongly driven by the expected look of the captured samples.

5 Conclusions

In this paper, we have presented a modification of the fuzzy c-means algorithm. The fuzzy c-means algorithm is one of the most popular clustering techniques in image processing. In the past, it has been modified in many ways to take into account different constraints. In our case, we have added an additional disparity constraint and examined its impact on the segmentation performance and depth discrimination. In the context of the image segmentation, we see the advantage of the proposed joint analysis using brightness and depth constraints. We believe, such combination improves the segmentation by creating edges not only in places where brightness changes abruptly but also in places of the depth discontinuities. Objects of the similar colour in different depths may be connected by the classical algorithm but with an additional depth constraint they are separated correctly.

The motivation was to develop a segmentation technique that can be used in cases, where we have the possibility of obtaining the stereo images and, in such way, improve the segmentation by applying additional depth information. In the biological application (the segmentation of the moss layers), the method provided better results than the standard fuzzy c-mean algorithm. As the algorithm was intended for this specific application, we have mainly tested and evaluated the algorithm on the datasets that resemble stone structures (e.g., the standard "Map" dataset). For such cases, the algorithm provides very good results.

To sum up, the paper proposed the method that improves the segmentation in cases where the pixel intensities are not sufficient for correct segmentation and the stereo images are available. This area of research, however, still offers the space for improvements. The results can be further improved by tuning the distance weights. The goal is to create an algorithm that can automatically adapt the weight variables according to the input dataset. Similar approaches were already published for the closely related k-means clustering, e.g. [13,21], and should be applicable to the fuzzy c-means as well.

Acknowledgements. This work was partially supported by the SGS grant No. SP2014/ 170 of VŠB - Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science.

References

1. Aik, L.E., Choon, T.W.: Enhancing passive stereo face recognition using pca and fuzzy c-means clustering. Int. J. Video Image Process. Netw. Secur. **11**(4), 1–5 (2011)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell (1981)
3. Bigand, A., Bouwmans, T., Dubus, J.: A new stereomatching algorithm based on linear features and the fuzzy integral. Pattern Recogn. Lett. **22**(2), 133–146 (2001)
4. Bleyer, M., Gelautz, M.: A layered stereo algorithm using image segmentation and global visibility constraints. In: Proceedings of the IEEE International Conference on Image Processing, pp. 2997–3000 (2004)
5. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–655 (1998)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
7. Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J.: Fuzzy c-means clustering with spatial information for image segmentation. Comput. Med. Imaging Graph. **30**(1), 9–15 (2006)
8. Cox, I.J., Hingorani, S.L., Rao, S.B., Maggs, B.M.: A maximum likelihood stereo algorithm. Comput. Vis. Image Underst. **63**, 542–567 (1996)
9. Cyganek, B.: An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley, New York (2007)
10. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. **3**(3), 32–57 (1973)
11. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 807–814 (2005)

12. Hirschmüller, H.: Stereo vision in structured environments by consistent semi-global matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2386–2393. IEEE Computer Society (2006)
13. Huang, J., Ng, M., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 657–668 (2005)
14. Kim, J., Kolmogorov, V., Zabih, R.: Visual correspondence using energy minimization and mutual information. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1040 (2003)
15. Klaus, A., Sormann, M., Karner, K.F.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 3, pp. 15–18 (2006)
16. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 508–515 (2001)
17. Liew, A.W.C., Leung, S.H., Lau, W.H.: Fuzzy image clustering incorporating spatial continuity. IEEE Proc. Vis. Image Sig. Process. **147**, 185–192 (2000)
18. Liu, T., Zhang, P., Luo, L.: Dense stereo correspondence with contrast context histogram, segmentation-based two-pass aggregation and occlusion handling. In: Proceedings of the Pacific-Rim Symposium on Image and Video Technology, pp. 449–461 (2009)
19. Meena, A., Raja, R.: Spatial fuzzy c means pet image segmentation of neurodegenerative disorder. CoRR abs/1303.0647 (2013)
20. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for Fuzzy Clustering: Methods in C-Means Clustering with Applications. Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2008)
21. Modha, D., Spangler, S.: Feature weighting in k-means clustering. Mach. Learn. **52**, 217–237 (2003)
22. Ntalianis, K. S., Doulamis, A., Doulamis, N., Kollias, S.: Unsupervised segmentation of stereoscopic video objects: investigation of two depth-based approaches. In: Proceedings of the 14th International Conference of Digital Signal Processing, vol. 2, pp. 693–696 (2002)
23. Ogawa, H.: A fuzzy relaxation technique for partial shape matching. Pattern Recogn. Lett. **15**(4), 349–355 (1994)
24. Pal, N., Bezdek, J.: On cluster validity for the fuzzy c-means model. IEEE Trans. Fuzzy Syst. **3**(3), 370–379 (1995)
25. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**(1–3), 7–42 (2002)
26. Szeliski, R., Zabih, R.: An experimental comparison of stereo algorithms. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 1–19. Springer, Heidelberg (2000)
27. Taguchi, Y., Wilburn, B., Zitnick, C. L.: Stereo reconstruction with mixed pixels using adaptive over-segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
28. Tao, H., Sawhney, H.S., Kumar, R.: A global matching framework for stereo computation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 532–539 (2001)
29. Tombari, F., Mattoccia, S., di Stefano, L.: Segmentation-based adaptive support for accurate stereo correspondence. In: Proceedings of the Pacific-Rim Symposium on Image and Video Technology, pp. 427–438 (2007)

30. Veksler, O.: Stereo correspondence by dynamic programming on a tree. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 384–390 (2005)
31. Zimmermann, H.: Fuzzy Set Theory and its applications. Kluwer Academic, Dordrecht (1991)
32. Zitnick, C.L., Kang, S.B.: Stereo for image-based rendering using image over-segmentation. Int. J. Comput. Vis. **75**(1), 49–65 (2007)