

# Efficiently Clustering Earth Mover’s Distance

Jenny Wagner, Björn Ommer

Interdisciplinary Center for Scientific Computing, Heidelberg University (Germany)  
wagner@math.uni-heidelberg.de, ommer@uni-heidelberg.de

**Abstract.** The two-class clustering problem is formulated as an integer convex optimisation problem which determines the maximum of the Earth Movers Distance (EMD) between two classes, constructing a bipartite graph with minimum flow and maximum inter-class EMD between two sets. Subsequently including the nearest neighbours of the start point in feature space and calculating the EMD for this labellings quickly converges to a robust optimum. A histogram of grey values with the number of bins  $b$  as the only parameter is used as feature, which makes run time complexity independent of the number of pixels. After convergence in  $\mathcal{O}(b)$  steps, spatial correlations can be taken into account by total variational smoothing. Testing the algorithm on real world images from commonly used databases reveals that it is competitive to state-of-the-art methods, while it deterministically yields hard assignments without requiring any a priori knowledge of the input data or similarity matrices to be calculated.

## 1 Introduction

The mathematical concept of a distance between two probability distributions, which later became known as the Wasserstein metric, was formulated by L.N. Vaserstein in 1969 [1]. Twenty years later, the special case of the first Wasserstein metric was applied in image processing for the first time, when [2] used it for their multiple resolution analysis of images. Today, this distance measure, also called Earth Mover’s Distance (EMD), mainly serves as a similarity measure for point sets in feature space in a vast number of different applications. In [3], for instance, the EMD of two colour or texture histograms of different images is defined as a measure of similarity between those images for content-based image retrieval. [4] focus on an efficient large scale EMD implementation for shape recognition and interest point matching, which is also based on a robust histogram comparison.

Contrary to all known applications of EMD so far, Efficiently Clustering Earth Mover’s Distance (ECEMD) uses EMD to *directly* separate the feature space into two sets/ classes of points by finding the class assignment configuration for which the EMD between those classes is at maximum. There is no need to calculate large scale similarity matrices and no further (complex) algorithm for clustering is required. Using histograms as feature spaces, all pixels of the image

are taken into account, i.e. subsampling of the image or restrictions of the distance to take into account only a limited number of neighbours, as for large scale similarity matrices, are not necessary. Similar to [5], the algorithm can also be extended to subsequently segment an image into  $k$  regions of interest, extracting one class after the other out of the background set.

Section 2 introduces the mathematical problem formulation as a convex optimisation problem, the related choice of feature space and discusses the theoretical advantages this definition implicates. Section 3 then covers all implementational aspects, starting from the algorithm that solves the clustering problem to the best initialisation that guarantees robust and fast convergence. In Section 4, the algorithm is applied to selected example images from online databases and the segmentation results are compared to the ground truth, to a standard segmenting approach [6] and a clustering approach that uses similarity measures [5]. Finally, Section 5 summarises the strengths and limits of the method presented here and gives an outlook of future research to improve the algorithm.

## 2 Problem Formulation and Related Work

Let  $n$  be the number of points  $x_i$  ( $i \in \{1, \dots, n\}$ ) in feature space and denote the class labels  $c_i = -1$  or  $c_i = +1$  for back- and foreground assignment, respectively. Given a weight  $w_i$  for each  $x_i$ , assumed to be a pile of earth of height  $w_i$  at point  $x_i$ , it is denoted by  $w_i^{\mathcal{F}}$  if  $x_i$  is in the foreground  $\mathcal{F}$  and  $w_i^{\mathcal{B}}$  if  $x_i$  is in the background. The optimal class assignments to  $\mathcal{F}$  and  $\mathcal{B}$  are chosen such that the EMD between these classes is maximal. The EMD itself can be understood as finding the minimum work required to transport all piles from one class to the other, respecting that the *entire* amount of earth has to be moved and that each  $x_i$  can only acquire or transport a pile up to its own  $w_i$ . This leads to the optimisation problem of (1) and (2). The first constraint assures that the work, also called flow,  $f_{ij}$  between each point  $x_i \in \mathcal{F}$  and  $x_j \in \mathcal{B}$  is unidirectional, the second and third that the flow from/ to one point  $x_i \in \mathcal{F}, \mathcal{B}$  does not exceed the weight  $w_i^{\mathcal{F}, \mathcal{B}}$  of this point. The fourth forces all weights of one class to be moved.

$$\max_{c_1, \dots, c_n} \{\text{EMD}(\mathcal{F}, \mathcal{B})\} = \max_{c_1, \dots, c_n} \left\{ \min_f \sum_{i=1}^m \sum_{j=1}^{n-m} d_{ij} \cdot f_{ij} \right\} \quad s.t. \quad (1)$$

$$f_{ij} \geq 0 \quad \sum_j f_{ij} \leq w_i^{\mathcal{F}} \quad \sum_i f_{ij} \leq w_j^{\mathcal{B}} \quad \sum_i \sum_j f_{ij} = \min \left\{ \sum_i w_i^{\mathcal{F}}, \sum_j w_j^{\mathcal{B}} \right\} \quad (2)$$

The distance  $d_{ij}$  between the pairs of  $x_i, x_j$  of the opposite classes in the objective function can be calculated as the Euclidean distance

$$d_{ij} = \|x_i - x_j\|_2^2 \quad \forall x_i : c_i = 1, \quad \forall x_j : c_j = -1. \quad (3)$$

As can be read off the indices of the sums,  $m$  points have been assumed to belong to the foreground class and  $n - m$  points to be background for  $1 \leq m < n$ ,  $m$  to be determined by the class assignments.

The problem defined in (1) and (2) aims at constructing a bipartite graph with minimal flow between the nodes of the fore- and background set, such that the fore- and background set have maximum distance. It belongs to the category of EXPTIME problems as the global optimum can be determined by  $2^n$  times solving the linear minimisation problem for all possible permutations of class assignments of the weighted points in feature space to the fore- and background.

The idea is similar to maximum margin clustering, which can be formulated as the convex integer optimisation problem introduced in [7]. Yet, while the latter is often relaxed to a semi-definite program for applications or uses random class assignments as [8], the algorithm presented here does not require a relaxation to soft cluster assignments and deterministically converges to a robust optimum.

Contrary to k-means or spectral clustering approaches, all linked by a general optimisation problem by [9], which minimise the *intra*-class variance, the approach presented here focuses on maximising the *inter*-class variance.

Since histograms of intensity values have often been used successfully in combination with the EMD, these are also adapted here to show how the algorithm works in principle. To do so, the grey values of the image (without loss of generality assumed to be in the interval  $[0, 1]$ ) are divided into  $n$  (equally distributed) bins, whose center coordinates are given by  $x_i$ ,  $i = 1, \dots, n$ . Then, the weights  $w_i^{\mathcal{F}}$  and  $w_j^{\mathcal{B}}$  are determined as the normalised number of entries in the bins of the foreground set and the background set. It is important to note that the weights are normalised with respect to the total number of pixels in the histogram and not with respect to the number of pixels in the fore- or background class, as usually applied for calculating the EMD as a similarity feature. The overall normalisation used here takes account for the weight of each  $x_i$  in the context of the image and not in the context of the cluster it is currently assigned to. Specifying the objects of interest further, data-adapted features of a different kind are also applicable.

Having determined an optimal cluster configuration that mainly relies on proximity of vectors in feature space, continuous graph cuts as described in [10] can be applied to the resulting segmented image in order to account for spatial correlations in image space as well. This yields a denoised final segmentation, so that the boundaries between fore- and background are minimised by assembling contiguous regions, depending on the weight parameter  $\lambda$  that controls the degree of smoothing. Then,

$$\inf_{\tilde{c} \in [0,1]^n} \{ \langle (c_1, \dots, c_n)^T, \tilde{c} \rangle + \lambda \cdot \text{TV}(\tilde{c}) \} \quad (4)$$

yields the final integer class labelling vector  $\tilde{c}$  ( $\tilde{c}_i = 0$  for background and  $\tilde{c}_i = 1$  for foreground) of all feature vectors  $x_i$ .

### 3 Implementation

The implementational difficulty lies in solving the maximum problem in (1), i.e. finding the class assignments, which is an NP-hard combinatorial problem, while calculating the EMD for a given class labelling is solving a linear program. Without a priori information about the object of interest, the algorithm can start with assigning a single arbitrary feature vector to the foreground, dividing the set of feature vectors into  $\mathcal{F}$  consisting of only one point and  $\mathcal{B}$  containing  $n - 1$  points. For these class assignments the EMD is calculated. After that, the nearest neighbours of the feature vector in  $\mathcal{F}$  that are stored in a previously calculated array are subsequently added to  $\mathcal{F}$  until the EMD of the new cluster configuration becomes smaller than the previous one or all vectors are assigned to be in  $\mathcal{F}$ . This procedure can be summarised in Algorithm 1.

---

**Algorithm 1** Algorithmic implementation of ECEMD
 

---

```

1 program (out:  $\mathcal{F}, \mathcal{B}$ ) = ECEMD (in:  $x_1, \dots, x_n$ )
2  $\mathcal{F} \leftarrow \{x_1\}$ ; initialise one vector in  $\mathcal{F}$ 
3  $\mathcal{B} \leftarrow \{x_2, \dots, x_n\}$ ; rest of vectors in  $\mathcal{B}$ 
4  $kNN(x_1) \leftarrow [1NN(x_1), 2NN(x_1), \dots, (n-1)NN(x_1)]$ ; calculate neighbours
5  $EMD_{last} \leftarrow 0$ ; initialise EMD-variable
6  $EMD_{next} \leftarrow EMD(\mathcal{F}, \mathcal{B})$ ; calculate first EMD
7  $k \leftarrow 1$ ; initialise loop to include nearest neighbours in  $\mathcal{F}$ 
8 while ( $\mathcal{B}$  NOT  $\{\}$ ) OR ( $EMD_{next} < EMD_{last}$ ) do
9    $\mathcal{F} \leftarrow \mathcal{F} \cup \{kNN(x_1)\}$ ; append next nearest neighbour to  $\mathcal{F}$ 
10   $\mathcal{B} \leftarrow \mathcal{B} / \{kNN(x_1)\}$ ; truncate this nearest neighbour from  $\mathcal{B}$ 
11   $EMD_{last} \leftarrow EMD_{next}$ ; store last EMD solution
12   $EMD_{next} \leftarrow EMD(\mathcal{F}, \mathcal{B})$ ; calculate next EMD solution
13   $k \leftarrow k + 1$ ;
14 end while
15 return  $\mathcal{F}, \mathcal{B}$ ;

```

---

To assure fast and robust convergence in the histogram feature space considered here, the first foreground vector is best defined to be the bin with the smallest or the largest bin coordinate. For the one-dimensional case, using the first or last bin is the best choice, since this results in the desired intensity value separation with convergence after including those of the bins to the foreground that lead to an almost equally weighted cluster distribution. In the (multi-dimensional) general case, however, the initialisation can be either done by the user or by a more problem-adapted initialisation, making the approach semi-supervised.

Using this k-nearest-neighbour approach takes into account the similarities in feature space but only requires  $\mathcal{O}(n)$  distance calculations and amount of memory contrary to algorithms applying pairwise similarity matrices.

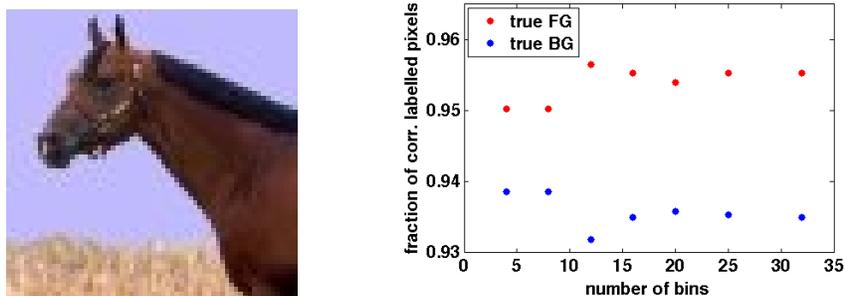
Extending the idea of EMD-clustering to the multi-class case, iteratively extracting classes out of the feature space again yields hard cluster assignments.

Since the total variational smoothing is also able to handle several classes, the clustering result can still be plugged into (4) for spatial correlation.

## 4 Experimental Results

For the sake of brevity, the advantages and limits of the algorithm can only be highlighted by a few examples and comparisons. Tuning of the method itself to be optimally adapted to a specific application is left to further work.

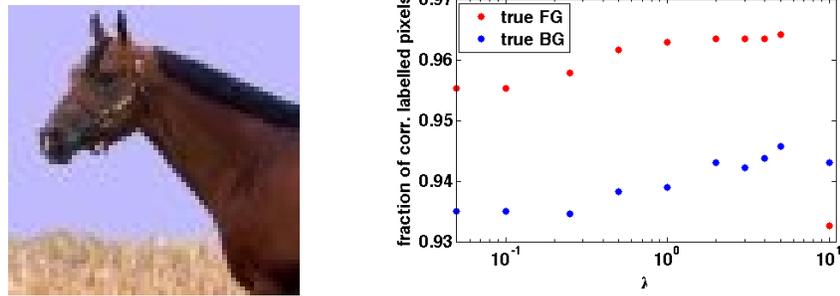
**Bin number.** First, the effect of varying the number of histogram bins is investigated. In order to do so, the algorithm is applied to the image of size  $64 \times 64$  shown in Fig. 1 (left) to divide its contents into two classes while the number of bins is varied from 4 to 32.



**Fig. 1.** Left: image Right: Dependence of the segmentation results of this image on the number of bins in feature space: *red markers*: correctly assigned percentage of pixels to the foreground, *blue markers* correctly assigned percentage of pixels to the background

Comparing the segmentation results with the ground truth, it can be observed that the number of correctly assigned fore- and background pixels is over 93% and varies only in the range of 1% with increasing number of bins. Counting the number of bins that are assigned to be fore- and background, it is interesting to note that the algorithm converges to almost equal numbers of fore- and background bins.

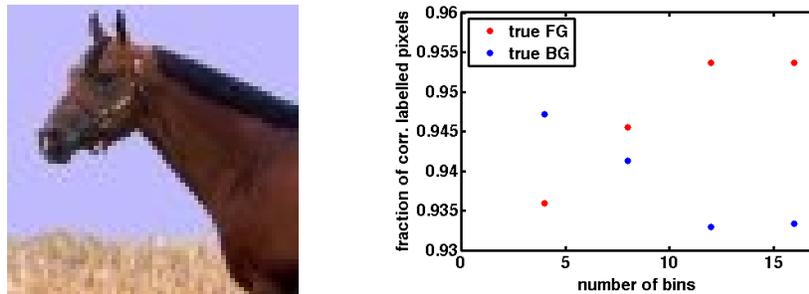
**Total variational smoothing.** Using the same image for a fixed number of bins (16 in this case), total variational smoothing is applied after convergence and the dependence of the number of correctly assigned pixels to the smoothing parameter  $\lambda$  is investigated as summarised in Fig. 2, where  $\lambda$  is varied from 0 to 10. The graphs for the correctly assigned percentage of pixels show an increase in coincidence with the ground truth up to  $\lambda = 5$ , then the percentage decreases. This tendency can be explained by oversmoothing, observing the segmentations shown in Fig. 3



**Fig. 2.** Left: image Right: Dependence of the segmentation results of this image on the weight parameter  $\lambda$  of subsequent total variational smoothing: *red markers*: correctly assigned percentage of pixels to the foreground, *blue markers* correctly assigned percentage of pixels to the background



**Fig. 3.** From left to right: segmentation without total variational smoothing, with  $\lambda = 0.5$  (visually closest to ground truth), with  $\lambda = 5$  and with  $\lambda = 10$



**Fig. 4.** Left: image Right: Dependence of the globally optimal segmentation results of this image on the number of bins in feature space: *red markers*: correctly assigned percentage of pixels to the foreground, *blue markers* correctly assigned percentage of pixels to the background

**Combinatorial solution.** Taking into account that the algorithm follows a greedy strategy, the local optimum of the result should be compared to the global optimum of the model defined by (1) and (2). By means of this, it is possible to evaluate the segmentation strength of the model in order to determine, whether the ansatz is appropriate in the first place. Furthermore, comparing the result

obtained by the greedy algorithm to the global optimum, the approximation gap to the NP-hard problem can be computed. Hence, for a small number of bins, a comparison of the globally optimal solution to the ground truth is performed. The number of bins is limited due to the exponential run time complexity of finding the combinatorial optimum of (1) under the constraints of (2).

As can be observed from the right hand side of Fig. 4, the results lie in the same range as the results obtained by the greedy algorithm. Quantifying the gap between the global optimum and the local one found by the algorithm, the deviations are less than 1.5%, where the correct foreground assignments are higher in the case of the greedy algorithm. On the average, the greedy and combinatorial algorithm assign 94.4% of all pixels correctly in the range of 4 to 16 bins. From this can be concluded that the model as well as the algorithm are reasonably chosen in the sense that they yield segmentations close to the ground truth.

**Comparison to competitive algorithms.** For the comparison of the Efficiently Clustering EMD to the ground truth and competitive approaches by means of example images from online databases shown below, the number of bins is set to 16 and  $\lambda = 0.5$ . The comparisons are made for the images shown in Tab. 1, calculating the relative confusion matrices

$$C = \begin{pmatrix} \text{true FG} & \text{false FG} \\ \text{false BG} & \text{true BG} \end{pmatrix}. \quad (5)$$

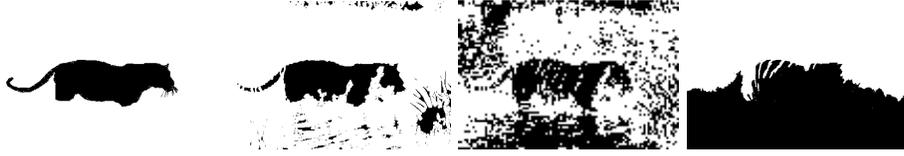
In the implementation of [5] the images were subsampled in order to handle the large amount of data in the similarity matrices, so that only up to 5% of the image pixels determined the segmentation and a simple Gaussian distance of the RGB colour vectors was chosen as similarity measure between those points.

As the results show, the Efficiently Clustering EMD yields better recognition rates than the standard approaches in those cases where the simple intensity value histogram features are appropriate to separate fore- from background. Furthermore, total variational smoothing does only lead to improvements in recognition of about 1%, as already found out in Fig. 2 (right).

In the special case of the tiger (fourth image of Tab. 1), the gain in recognition applying total variational smoothing is much higher, as ECEMD assigns the orange pixels of the tiger to foreground and the dark stripes to background, so that spatial correlation completes the striped coat to one contiguous region. Visually comparing the outcome of ECEMD with total variation to the competitive segmentations, as shown in Fig. 5, the former comes closest to the ground truth (left), detailed confusion matrices of each algorithm can be found in Tab. 1.

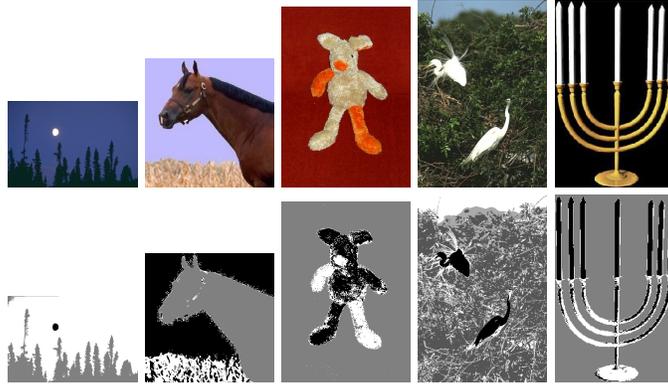
**Table 1.** Example images and comparison results to ground truth in form of relative confusion matrices for own approaches and competitive methods, images can be found in Appendix A

Image details	own	own + TV	[5]	[6]
holes 100 × 100 own creation	$\begin{pmatrix} 0.99 & 0.16 \\ 0.01 & 0.84 \end{pmatrix}$	$\begin{pmatrix} 1.00 & 0.05 \\ 0.00 & 0.95 \end{pmatrix}$	$\begin{pmatrix} 1.00 & 0.31 \\ 0.00 & 0.69 \end{pmatrix}$	$\begin{pmatrix} 0.94 & 0.03 \\ 0.06 & 0.97 \end{pmatrix}$
horse part 256 × 256 Weizmann DB	$\begin{pmatrix} 0.96 & 0.07 \\ 0.04 & 0.93 \end{pmatrix}$	$\begin{pmatrix} 0.96 & 0.07 \\ 0.04 & 0.93 \end{pmatrix}$	$\begin{pmatrix} 0.97 & 0.33 \\ 0.03 & 0.67 \end{pmatrix}$	$\begin{pmatrix} 0.85 & 0.20 \\ 0.15 & 0.80 \end{pmatrix}$
horse 717 × 525 Weizmann DB	$\begin{pmatrix} 0.99 & 0.24 \\ 0.01 & 0.76 \end{pmatrix}$	$\begin{pmatrix} 0.99 & 0.24 \\ 0.01 & 0.76 \end{pmatrix}$	$\begin{pmatrix} 0.99 & 0.53 \\ 0.01 & 0.47 \end{pmatrix}$	$\begin{pmatrix} 0.64 & 0.53 \\ 0.36 & 0.47 \end{pmatrix}$
tiger 481 × 321 Berkeley Segm. DB	$\begin{pmatrix} 0.63 & 0.03 \\ 0.37 & 0.97 \end{pmatrix}$	$\begin{pmatrix} 0.85 & 0.05 \\ 0.15 & 0.95 \end{pmatrix}$	$\begin{pmatrix} 0.82 & 0.33 \\ 0.18 & 0.67 \end{pmatrix}$	$\begin{pmatrix} 0.74 & 0.43 \\ 0.26 & 0.57 \end{pmatrix}$
llama 513 × 371 GrabCut DB	$\begin{pmatrix} 0.14 & 0.02 \\ 0.86 & 0.98 \end{pmatrix}$	$\begin{pmatrix} 0.27 & 0.03 \\ 0.73 & 0.97 \end{pmatrix}$	$\begin{pmatrix} 0.77 & 0.52 \\ 0.23 & 0.48 \end{pmatrix}$	$\begin{pmatrix} 0.93 & 0.45 \\ 0.07 & 0.55 \end{pmatrix}$
man 321 × 481 Berkeley Segm. DB	$\begin{pmatrix} 0.77 & 0.00 \\ 0.23 & 1.00 \end{pmatrix}$	$\begin{pmatrix} 0.77 & 0.00 \\ 0.23 & 1.00 \end{pmatrix}$	$\begin{pmatrix} 0.56 & 0.00 \\ 0.44 & 1.00 \end{pmatrix}$	$\begin{pmatrix} 0.60 & 0.08 \\ 0.40 & 0.92 \end{pmatrix}$
horses 481 × 321 Berkeley Segm. DB	$\begin{pmatrix} 0.88 & 0.24 \\ 0.12 & 0.76 \end{pmatrix}$	$\begin{pmatrix} 0.89 & 0.24 \\ 0.11 & 0.76 \end{pmatrix}$	$\begin{pmatrix} 0.99 & 0.37 \\ 0.01 & 0.63 \end{pmatrix}$	$\begin{pmatrix} 0.72 & 0.68 \\ 0.27 & 0.32 \end{pmatrix}$
swimmer 481 × 321 Berkeley Segm. DB	$\begin{pmatrix} 0.82 & 0.30 \\ 0.18 & 0.70 \end{pmatrix}$	$\begin{pmatrix} 0.82 & 0.30 \\ 0.28 & 0.70 \end{pmatrix}$	$\begin{pmatrix} 0.82 & 0.29 \\ 0.18 & 0.71 \end{pmatrix}$	$\begin{pmatrix} 0.64 & 0.59 \\ 0.35 & 0.41 \end{pmatrix}$
birds 481 × 321 Berkeley Segm. DB	$\begin{pmatrix} 0.95 & 0.43 \\ 0.05 & 0.57 \end{pmatrix}$	$\begin{pmatrix} 0.96 & 0.43 \\ 0.04 & 0.57 \end{pmatrix}$	$\begin{pmatrix} 0.92 & 0.35 \\ 0.08 & 0.65 \end{pmatrix}$	$\begin{pmatrix} 0.18 & 0.76 \\ 0.82 & 0.24 \end{pmatrix}$
astronauts 481 × 321 Berkeley Segm. DB	$\begin{pmatrix} 0.85 & 0.07 \\ 0.15 & 0.93 \end{pmatrix}$	$\begin{pmatrix} 0.86 & 0.07 \\ 0.14 & 0.93 \end{pmatrix}$	$\begin{pmatrix} 0.90 & 0.48 \\ 0.10 & 0.52 \end{pmatrix}$	$\begin{pmatrix} 0.88 & 0.01 \\ 0.12 & 0.99 \end{pmatrix}$



**Fig. 5.** From left to right: ground truth, ECEMD with total variational smoothing, segmentation by [5], segmentation by [6]

**Multi-class segmentation.** Multi-class segmentation can be implemented similar to [5] by iterative application of the algorithm. At first, one foreground class is extracted, then, the next class is segmented out of the remaining background points until the number of predefined classes is reached. This algorithm leads to results shown in Fig. 6. Comparing the original image with the segmentation results, good coincidence can be observed, noting that the binning only depends on intensity values. As especially the third and fourth image show, however, total variational denoising or texture based features could further improve the results.



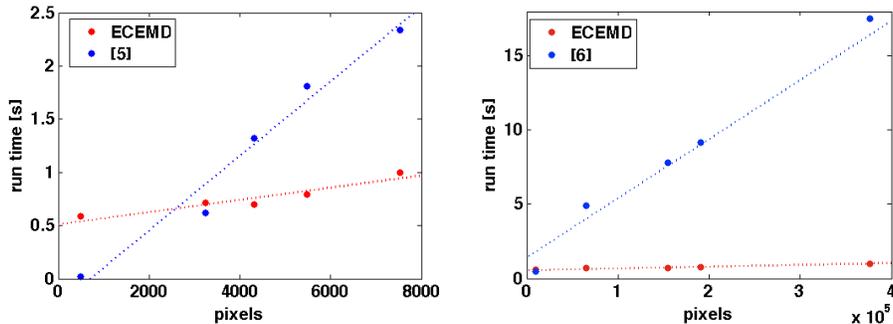
**Fig. 6.** Top: original images Bottom: multi-class segmentation with three classes

**Run time measurements.** As the algorithm is implemented in MATLAB, the MATLAB profiler can be used to determine the run time of the algorithm in a last experiment. Scaling the size of the horse head image from its original size of  $256 \times 256$  pixels over  $128 \times 128$ ,  $64 \times 64$  and  $32 \times 32$  down to  $16 \times 16$  pixels and measuring the overall run time of the algorithm (without variational smoothing but with image loading and histogram creation), it is observed that the total amount of actual CPU time used for calculations is always less than 2 seconds and independent of the number of pixels on a MacBook Pro Model 3.1 (2.4 GHz Intel Core 2 Duo, 4GB DDR2RAM). This result could have been expected be-

forehand, since the algorithm operates only on the histogram with fixed number of bins and already sorted bin coordinates.

The run time measurements in Fig. 7 on the left show the dependence on the number of pixels of five different images processed by [5] compared to ECEMD for the same images. It can be observed that ECEMD (without total variational smoothing) is weakly dependent on the number of pixels, which originates from the dependence of the histogram creation on the number of pixels. ECEMD, processing all pixels, is faster than [5], taking into account that [5] only uses 5% of all pixels for clustering and furthermore requires the calculation of a similarity matrix of complexity  $\mathcal{O}(n^2)$ , which takes 1200 seconds for 500 pixels and is not included in the time measurements. But, even without the calculation of the similarity matrix, [5] is slower for more than 3000 pixels.

Compared to [6], which has a linear time complexity dependent on the number of pixels as shown in the right graph of Fig. 7, ECEMD including histogram creation is faster than [6]. For small images, the run time is comparable, while for increasing number of pixels, ECEMD is at least six times faster yielding comparable segmentation results.



**Fig. 7.** Left: Run time comparison of ECEMD and [5] without creation of similarity matrices Right: Run time comparison of ECEMD and [6]

## 5 Summary and Outlook

In summary, the ECEMD is formulated as an unsupervised convex integer program that applies the Earth Mover’s Distance to directly separate foreground and background, inspired by the principles of maximum margin clustering. The implementation starts at one foreground point in feature space and calculates a short series of linear programs that subsequently include the nearest neighbours of the starting point into the foreground set until it deterministically converges to an integer optimum that lies close to ground truth. If the chosen feature space

was not distinctive enough to separate the classes correctly, spatial correlation among the pixels can be taken into account afterwards, applying total variational smoothing to enhance the result. As was shown in Section 4, the latter can only improve the image by a significant amount in those cases when the feature space is inappropriate to segment the data. But recognition rates that surpass those of [5] and [6] and approximate the ground truth with more than 90% correctly assigned pixels in cases where the feature space is suitably chosen, there is no space for more than fine tuning.

Advantages of ECEMD certainly lie in the facts that the bin number of the histogram is the only parameter to be put in, that no subsampling of the image and no previous training step is required to process the data. Compared to other approaches that use the EMD, the hard cluster assignments of ECEMD are furthermore advantageous, as well as the short run time of the ECEMD which is independent of the number of input pixels in its current implementation that uses an intensity value histogram as feature space (see Section 3).

Yet, exact benchmark tests, for example the Berkeley Benchmark, still remain to be evaluated for the algorithm in order to prove its qualitatively high segmentation power on a large amount of various images. From the results of this test could additionally be concluded whether the simple intensity feature suffices for all kind of object categories or in how far category adapted features (e.g. texture, edges, colour) can improve these results. A benchmark test also offers the opportunity to find a common platform to compare ECEMD to other approaches like k-means or maximum margin clustering to investigate in which cases the maximisation of inter-class-variance of ECEMD excels over these ansatzes.

In the future, improvements to the algorithm itself can be made in form of user interaction defining the starting point of the algorithm as one image pixel, patch or region that is contained in the object of interest. To achieve this, the algorithm could be included in a user-in-painting framework as developed for GrabCut [11] or Ilastik [12].

As far as parallelisation is concerned, it is possible to use the experimental observation that ECEMD usually converges to clusters with equal numbers of bins, so that the EMD calculation of the different cluster partitions could be run in parallel and after termination, the results are compared to find the optimal assignments.

**Acknowledgement.** JW gratefully acknowledges the financial support of the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences.

## References

1. Vaserstein, L.N.: On the stabilization of the general linear group over a ring. *Math. USSR-Sbornik* **8** (1969) 383–400

2. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (1989) 739–742
3. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vision* **40** (2000) 99–121
4. Ling, H., Okada, K.: An efficient earth mover’s distance algorithm for robust histogram comparison. *IEEE Trans. Pat. Anal. Mach. Intell.* **29** (2007) 840–853
5. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. *IEEE Trans. Patt. Anal. Mach. Intell.* **29** (2007) 167–172
6. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: *CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 2*, Washington, DC, USA, IEEE Computer Society (2005) 1124–1131
7. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In Saul, L.K., Weiss, Y., Bottou, L., eds.: *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA (2005) 1537–1544
8. Gieseke, F., Pahikkala, T., Kramer, O.: Fast evolutionary maximum margin clustering. In: *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, ACM (2009) 361–368
9. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: *Proc. ICCV. Volume 1.* (2005) 294–301
10. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program.* **103** (2005) 127–152
11. Rother, C., Kolmogorov, V., Blake, A.: ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **23** (2004) 309–314
12. Sommer, C., Straehle, C., Köthe, U., Hamprecht, F.: Interactive learning and segmentation tool kit 0.8 (rev 475) (2010)

## A Images

**Table 2.** Images used for experimental evaluation

