Bayesian Fusion of Color and Texture Segmentations

Roberto Manduchi

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91030
manduchi@jpl.nasa.gov

Abstract

In many applications one would like to use information from both color and texture features in order to segment an image. We propose a novel technique to combine "soft" segmentations computed for two or more features independently. Our algorithm merges models according to a mean entropy criterion, and allows to choose the appropriate number of classes for the final grouping. This technique also allows to improve the quality of supervised classification based on one feature (e.g. color) by merging information from unsupervised segmentation based on another feature (e.g., texture.)

1 Introduction

Image segmentation is a fundamental task in Computer Vision. Color and texture provide powerful cues for segmenting a still image, and much work has been devoted to developing grouping algorithms based on these two features [1],[3],[5]. In fact, most of the literature deals with segmentation based on either color or texture; this work was originated by the intuition that using information provided by *both* features, one should be able to obtain more robust and meaningful results.

Underlying our approach is the hypothesis that in typical images color and texture features are not statistically independent. Perhaps the simplest way to exploit this dependency is to concatenate the color and texture feature vectors together, and then run the grouping algorithm of choice on these super-vectors. This approach, however, may give the feeling of "comparing apples with oranges". Indeed, color and texture features often have very different statistical behaviors; one may prefer to use the most suitable grouping algorithm for each feature separately, and then somehow combine the results of the two segmentations together.

This work introduces a strategy to merge together in a Bayesian framework segmentations computed on color and texture features independently. The only requirement is that the segmentations are expressed in terms of posterior probabilities [2]. Note that most clustering algorithms based on mixture models explicitly compute estimates of the posterior distributions, and do the final assignment by Bayesian classification (i.e., they assign a feature to the component of the mixture model that most likely generated that feature.)

For example, in Figure 2 (b) and (c) we show instances of color and texture segmentation of the image in Figure 2 (a). The texture features are formed by the absolute values of the outputs of a bank of Gabor filters, smoothed by a gaussian kernel to enforce spatial coherence [3]. The mixture model in both cases has been estimated by Expectation Maximization [2]; the "hard" segmentation shown in the figures is the result of Bayesian classification based on such mixture models. Both mixture models have four classes, although our algorithm can accept any combination of classes. The scene in figure 2(a) is composed by a small number of homogeneous parts: two bushes, a paved road on the right, dirt soil on the left, a shadow area near a bush and piece of dark background. The color segmenter (figure 2(b)) successfully separates the "bush", the "background" and the "road" areas, but is unable to discriminate the "road" from "soil" parts, which have very similar color. The texture segmenter does separate the "road" and "soil" areas, but cannot discriminate the "road" from the "background" parts; in addition, it assigns the "soil" area to two distinct classes of the mixture model.

Our technique for model fusion involves two steps. First, the two models are merged by a "cartesian product" operator, discussed in section 2. This operation preserves all the information about the models, but has the disadvantage of creating a large number of classes, equal to the product of the number of classes of the two original models. Then, the number of classes

of the combined model is reduced by a technique, presented in section 3, that "clips together" sets of classes. Such classes are selected on the basis of a mean entropy criterion that minimizes the loss of "descriptiveness"; the mean entropy criterion also provides useful information for choosing the appropriate number of classes for the final model. An intriguing application of our algorithm is discussed in section 4, and involves information fusion from supervised classification (e.g., based on color) and unsupervised segmentation (e.g., based on texture.) The unsupervised segmentation is used to leverage the estimates provided by the trained model, resulting in a more accurate classification.

2 Cartesian product of mixture models

Our merging technique starts from K given mixture models [2] (called "models" in the following.) The i-th model, $\mathcal{M}_i$, is composed by N_i classes, and defines a probability density function $p_i(z_i)$:

$$p_i(z_i) = \sum_{j=1}^{N_i} p_i(z_i|j) P_i(j) \qquad (1)$$

where z_i, the observed feature, lives in a space Z_i. For example, z_i may be a color vector, or a texture feature in a multiscale/multiorientation space. The conditional likelihood functions $p_i(z_i|j)$ and the priors $P_i(j)$ specify the model completely. The posterior distributions are given by Bayes' rule:

$$P_i(j|z_i) = \frac{p_i(z_i|j) P_i(j)}{p_i(z_i)} \qquad (2)$$

$P_i(j|z_i)$ is the probability that the observed feature z_i was generated by the class of index j. The Bayesian classifier for $\mathcal{M}_i$ assigns a feature z_i to the class indexed by the location of the maximum of $P_i(j|z_i)$. To simplify our presentation, we will assume in the following that all the priors are strictly positive: if a prior $P_i(j)$ is null, we can safely remove the class with index j from the model.

The *cartesian product* $\mathcal{M}$ of the models $\mathcal{M}_i$ is a new model with probability distribution over $Z_1 \times \ldots \times Z_N$. $\mathcal{M}$ is completely specified by the following axioms:

1. $\mathcal{M}$ has $N = \prod_{i=1}^{K} N_i$ classes, corresponding to the cartesian product of the classes of the models $\mathcal{M}_i$: $j \leftrightarrow (j_1, \ldots, j_N)$.
2. The conditional likelihood of the feature $z = (z_1, \ldots, z_K)$ given the class of index j is equal to $p(z|j) = \prod_{i=1}^{K} p_i(z_i|j_i)$.
3. The priors factorize as $P(j) = \prod_{i=1}^{K} P_i(j_i)$.

It follows straightforwardly that the likelihood and the posteriors of the cartesian product of models factorize as well:

$$p(z) = \prod_{i=1}^{K} p_i(z_i) \,, \; P(j|z) = \prod_{i=1}^{K} P_i(j_i|z_i) \qquad (3)$$

Note that all the information about the K original models is preserved in their cartesian product $\mathcal{M}$. The Bayesian classifier for $\mathcal{M}$ assigns a feature z to the model $j \leftrightarrow (j_1, \ldots, j_N)$ such that j_i is the class assigned to z_i by the Bayesian classifier for $\mathcal{M}_i$. Figure 2 (d) shows the Bayesian segmentation relative to the cartesian product of the color and texture models of figure 2 (b) and (c). The new model has 16 classes. In the next section we describe a procedure to reduce the dimensionality (i.e., the number of classes) of a model, in such a way that the loss of "descriptiveness" of the model is minimized.

3 Dimensionality reduction

Assume we are given a model $\mathcal{M}$ with N classes. We introduce here a technique to build a new model that has fewer classes than $\mathcal{M}$ but explains the data exactly as $\mathcal{M}$, i.e., it defines the same likelihood $p(z)$ as $\mathcal{M}$. Suppose for example that we want to reduce the dimensionality of the model to $N - M$. Our strategy is very simple: we just "clip together" $M+1$ classes of $\mathcal{M}$ into a new super-class, leaving the other classes untouched. We may decide, for instance, to clip together the classes of index $N - M, \ldots, N$. The probability that a feature z was generated by the union of such classes according to $\mathcal{M}$ is equal to the sum of the corresponding posteriors. This is the value that we assign to the posterior $P^{new}(N - M|z)$ for the new model; the posteriors for the other classes are the same as in $\mathcal{M}$:

$$P^{new}(j|z) = P(j|z) \,, \; 1 \le j < N - M$$

$$P^{new}(N - M|z) = \sum_{j=N-M}^{N} P(j|z)$$

If in addition we impose that the likelihood function $p(z)$ is the same in both models, the new model is completely specified.

In general, to reduce the model dimension from N to $N-M$, we may choose any $L < M$ disjoint groups of classes with L_l components each, such that $\sum_{l=1}^{L} L_l = L + M$, and clip together the classes in each group. A criterion for the selection of the most appropriate clipping scheme is presented in the next section.

3.1 Mean entropy criterion

Dimensionality reduction via class-clipping involves some loss of "descriptiveness" of the model. If for example two classes that "explain" well two different portions of the image are clipped together, the new model will probably assign those two portions of the image to the same class. This observation suggests the criterion for selecting a clipping scheme introduced in this section. Our criterion is based on the notion of *mean entropy*, a well-known concept in the fields of statistical physics and mixture estimation [4],[6].

Given a feature z, the entropy of the posterior distribution $P(j|z)$ is defined by [2]

$$s(z) = -\sum_{j=1}^{N} P(j|z) \log P(j|z) \qquad (4)$$

The entropy $s(z)$ measures the softness of the class assignment. A distribution with null entropy assigns z to exactly one class; the maximum value of the entropy is $\log N$, and is attained if all classes are equally likely to have generated z.

The *mean entropy* S of a model is defined by the expectation of $s(z)$ with respect to the "real" distribution of z. In practice, the mean entropy can be estimated by averaging $s(z)$ over the observed image. A model with null mean entropy can only perform "hard" classification, and will be called *degenerate*. Note that the mean entropy of a model estimated via Expectation Maximization is a function of the "temperature" of the algorithm [6].

The following result, whose proof is in the Appendix, characterizes the relation between mean entropy and dimensionality reduction.

Fact 1 *Class-clipping never increases the mean entropy of a model.*

In general, if a new super-class "takes over" two different portions of the image that the previous model assigns to two classes separately, a significant decrease of the mean entropy is expected. Hence, a suitable criterion for dimensionality reduction is the following one: choose the clipping scheme that minimizes the decrement of the mean entropy.

Unfortunately, the number of possible clipping schemes may be very high even for small model dimension. For example, in order to reduce the number of classes from 16 to 13 we may choose among 45,500 different combinations of class clipping. Measuring the decrease of mean entropy for each one of those schemes may require a prohibitive computational cost. A suboptimal solution can be found using a fast greedy algorithm that builds a sequence of clippings involving only two classes at a time. At each step, the two classes that minimize the decrease of the mean entropy are selected. The algorithm is described in detail in figure 1.

Greedy algorithm for dimensionality reduction: $N \to N - M$

Given the set of posteriors $P(j|z)$, $1 \le j \le N$:

Build auxiliary vector R and matrix D:
$R(j) = E[-P(j|x) \log P(j|x)]$, $1 \le j \le N$:

$$D(j,k) = \begin{cases} R(j) + R(k) + E[(P(j|x) + P(k|x)) \cdot \\ \cdot \log(P(j|x) + P(k|x))] & , 1 \le k < j \le N \\ \infty & , \text{otherwise} \end{cases}$$

Initialize an empty list L;
Repeat M times:
  $(\bar{j}, \bar{k}) = \arg\min D(j,k)$;
  Add $\bar{k}$ to the list L;
  Update $P(\bar{j}|z) \leftarrow P(\bar{j}|z) + P(\bar{k}|z)$, $P(\bar{k}|z) \leftarrow 0$;
  Update $R(\bar{j})$;
  Update $D(\bar{j}, k)$ for $k < \bar{j}$, $k \notin L$;
  Update $D(k, \bar{j})$ for $k > \bar{j}$, $k \notin L$;
  Set $D(j, \bar{k}) = \infty$ for $j > \bar{k}$;
  Set $D(\bar{k}, j) = \infty$ for $j < \bar{k}$;
Remove the classes indexed by the elements of L.

Figure 1: The greedy algorithm to select a class-clipping scheme (see section 3.1.)

Figure 2(g) shows an example of Bayesian classification after dimensionality reduction from 16 to 4 classes, based on the mean entropy criterion. Each class of the reduced dimension model now represents a characteristic area of the image. The computation of the optimal clipping scheme, by a Matlab implementation of our greedy algorithm, requires about 50 seconds of computation time on a Power Mac G3 266 Mhz (the image size is 256 × 380 pixels.)

3.2 Dimension selection

Mean entropy can also be used as an indicator to determine an appropriate number of classes for the reduced dimensionality model. In figure 2(k) we plotted the mean entropy as a function of the number of classes for our example. Note that the algorithm for the greedy selection of classes, which reduces the dimension by one at a time, allows us to easily compute these values as a by-product. It is interesting to note that the mean entropy does not decrease uniformly as the dimension is reduced; in fact, a number

of "phase transitions" are observed, corresponding to a few "representative" dimensions. As noted above, we may expect the result of the segmentation to change dramatically in correspondence of an abrupt decrease of the mean entropy. For example, figure 2(e) and (f) show results corresponding to dimension 10 and 6 respectively. The two segmentations look very similar; indeed, the mean entropy in the two cases is almost the same. However, if we reduce the number of classes, the mean entropy changes abruptly: this phenomenon explains the strong difference between the segmentations of figure 2(f) (6 classes) and (g) (4 classes). The mean entropy undergoes another large decrement if we reduce the dimension form 4 to 3: as shown in figure 2(h), this is due to the fact that the class representing the "soil" and the class representing the "bushes" have been clipped together.

3.3 Equalization

In the previous sections we have described a strategy for model fusion that first builds the cartesian product of two models, and then performs dimensionality reduction via class-clipping. An implicit assumptions was that the two original models give the same contribution to the final segmentation. This hypothesis does not hold true if the two models have very different values of the mean entropy. In this case, the model with the smallest entropy "dominates" the combined model.

We propose a simple equalization procedure that allows to merge two models with different mean entropies; the procedure can be applied to either one of the models. The equalization operator starts from a model $\mathcal{M}$ and produces a new model with the same number of classes N. The entropy of this new model can be tuned to match any desired value $S_0 < \log N$, and the associated Bayesian classifier yields the same results as the Bayesian classifier for $\mathcal{M}$.

The equalization operator simply replaces each posterior distribution $P(j|z)$ with the new distribution $P^{eq}(j|z)$, defined as follows:

$$P^{eq}(j|z) = c(z)P(j|z)^{\alpha} \ , \ \alpha \geq 0 \qquad (5)$$

where c is a normalizing coefficient:

$$c(z) = \frac{1}{\sum_{j=1}^{N} P(j|z)^{\alpha}} \qquad (6)$$

The mean entropy properties of the equalization operator are summarized by the following result:

Fact 2 *Equalization decreases the mean entropy of a non-degenerate model if* $\alpha > 1$, *and increases it if* $\alpha < 1$.

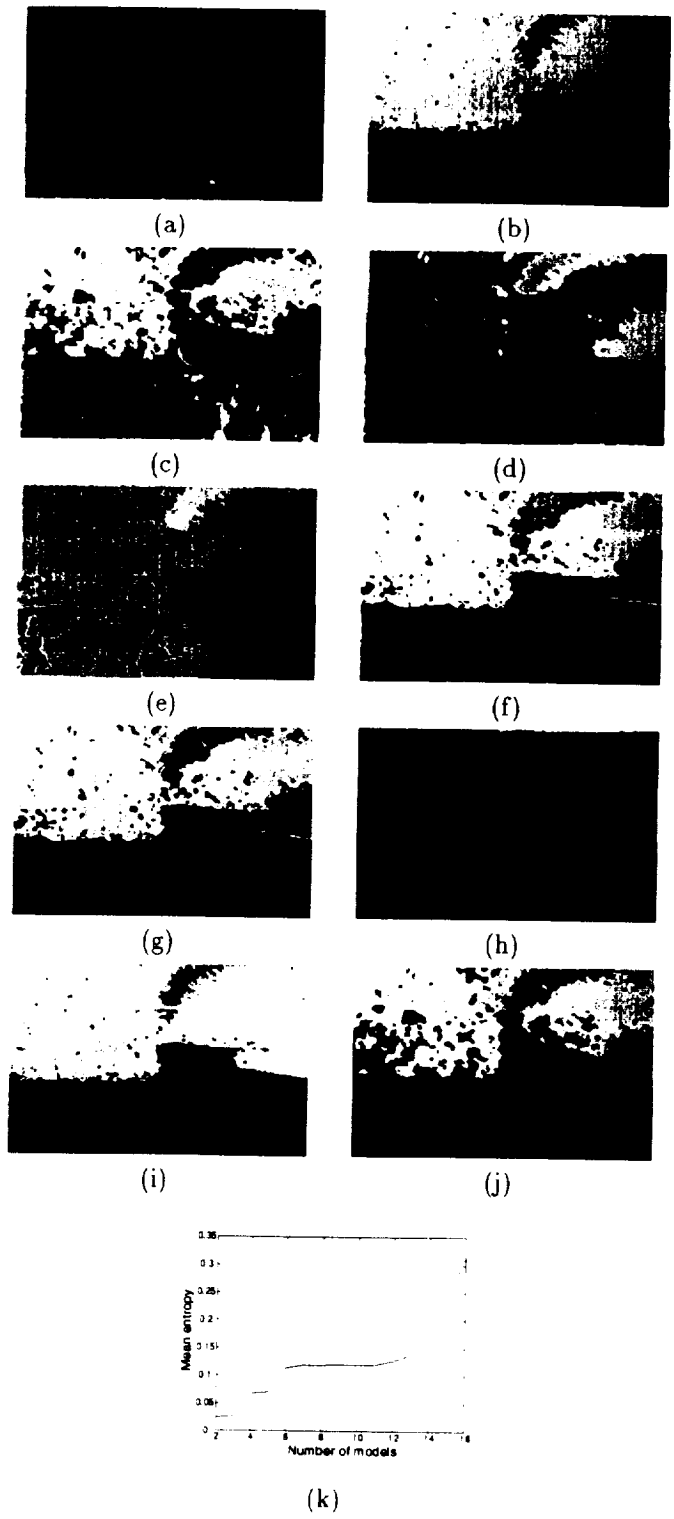

Figure 2: (a): Test image. (b) Color based segmentation (4 classes.) (c) Texture based segmentation (4 classes.) (d) Segmentation after cartesian product (16 classes.) (e) (h): Segmentation after model merging ((e): 10 classes, (f): 6 classes, (g): 4 classes, (h): 3 classes.) (i),(j): Segmentation after model merging (4 classes), with mean entropy of color based model set to 10 times smaller (i) or 10 times larger (j) than texture-based model. (k) Mean entropy as a function of model dimension.

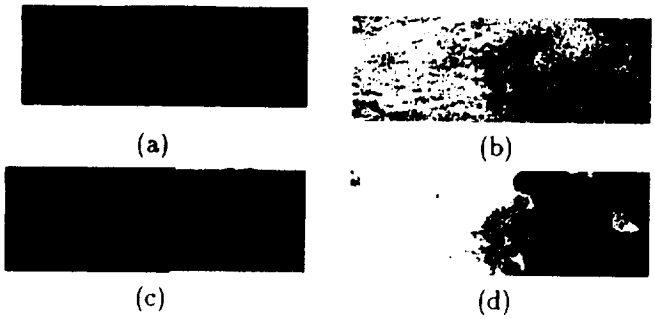

Figure 3: (a): Test image. (b) Color-based supervised classification into the "road" class (yellow) and the "grass" area (green.) (c) Texture-based unsupervised segmentation (3 classes.) (d) Hybrid supervised/unsupervised classification.

The proof can be found in the Appendix. Note that $\alpha = 0$ implies that the mean entropy of $P^{eq}(j|z)$ is equal to $\log N$; the mean entropy of $P^{eq}(j|z)$ can be made as small as desired by a suitable large value of α. Also note that for each feature z the location of the maximum of the posterior distribution is not changed by the equalization.

Now, suppose that the two models to be merged have different mean entropies. We may modify one of the models via the equalization operator, so that its mean entropy matches the mean entropy of the other model. The appropriate value of the parameter α may be found using any non-linear one-dimensional minimization technique.

In the example of figure 2, the mean entropies of the color and texture models were very similar, and equalization was not needed. Equalization, however, may be used to make either of the models dominant. For example, figure 2(i) and (j) show the results of Bayesian segmentation after equalizing the color-based model to a value of the mean entropy respectively 10 times smaller and 10 times larger than the mean entropy of the texture-based model (the combined model dimension was reduced to 4 by class-clipping.) This example shows that equalization is a practical and simple method for assigning different "weights" to the two models to be merged.

4 Hybrid classification

The main differences between supervised classification and unsupervised clustering are:

1. The classes ("labels") of a supervised classifier usually represent "physical" causes, and therefore are not logically interchangeable;
2. The statistical model is learned from training data.

The Bayesian classifier assigns a feature z to the maximizer of the posterior [2]. In many instances, only the conditional likelihoods $p(x|j)$ are learned; however, reasonable assumptions about the class priors $P(j)$ are often available, and the posteriors can be computed using Bayes' rule.

In this section we propose to merge a model $\mathcal{M}^s$ for supervised classification with a model $\mathcal{M}^u$ for unsupervised segmentation (based on a different feature space,) to create a "hybrid" classifier which assigns each image point to some label of $\mathcal{M}^s$. The intuition is that information from the "supervised model" (which identifies clusters in the feature space based on the current image) may be used to leverage the classification performed by the "unsupervised model", which is learned from a large training data set and may not be optimal for the current instance.

The merging algorithm discussed in the previous sections defines a model $\mathcal{M}$ with classes that are the union of cartesian products of classes from $\mathcal{M}^s$ and $\mathcal{M}^u$. If $\mathcal{C}$ represents a generic class of $\mathcal{M}$, we may write

$$\mathcal{C} = \bigcup_{v \in V} \bigcup_{w(v)} (\mathcal{C}^s_v, \mathcal{C}^u_{w(v)}) \tag{7}$$

where $\mathcal{C}^s$ and $\mathcal{C}^u$ are classes of $\mathcal{M}^s$ and $\mathcal{M}^u$ respectively, indexed by the corresponding subscripts. To complete the definition of the hybrid classification model, we need to identify each class $\mathcal{C}$ with some class of $\mathcal{C}^s$. If the set V of classes of $\mathcal{M}^s$ that form the super-class $\mathcal{C}$ is composed by just one element v, than we simply identify $\mathcal{C}$ with $\mathcal{C}^s_v$. In general, however, V may have more than one element; in this case, we identify $\mathcal{C}$ with the class $\mathcal{C}^s_v$ that maximizes the *contribution* to $\mathcal{C}$, defined by

$$E[P(\bigcup_{w(v)} (\mathcal{C}^s_v, \mathcal{C}^u_{w(v)})|z)] = \tag{8}$$
$$= E[P_s(v|z_s) \sum_{w(v)} P_u(w(v)|z_u)]$$

where $E[\cdot]$ represents the expectation operator, computed with respect to the "real" distribution of $z = (z_1, z_2)$, and $P_s(\cdot|\cdot)$, $P_u(\cdot|\cdot)$ and $P(\cdot|\cdot)$ represent the posteriors of the models $\mathcal{M}_s$, $\mathcal{M}_u$ and $\mathcal{M}$ respectively.

We present an example of hybrid classification in Figure 3. Figure 3(a) shows a scene with a dirt road on the left and dry grass on the right. Supervised color-based classification (figure 3(b)) is performed using a trained gaussian model. The "road" class and the "grass" class have very similar colors; this is the reason

why pixels in the top–right quadrant are misclassified as belonging to the "road" class. Figure 3(c) shows the results of unsupervised texture segmentation with three classes, computed via Expectation Maximization. The segmenter isolates uniform regions corresponding to the road and to the grass areas, plus a region corresponding to the border of the road. The two models are merged into a new model with four classes; the final hybrid classification is shown in Figure 3(d). The hybrid classifier has correctly labeled each one of the four classes of the merged model as either the "road" or the "grass" class. The information from the texture model has helped to correctly classify most pixels that were misclassified in figure 3(b).

5 Conclusions

We have presented a technique for merging together two segmentations, based on color and texture. Our technique is very general, and in principle can be applied also to other classes of features, such as motion; it only requires that the posterior distributions that originated the segmentations are available. The results show the effectiveness of the mean entropy criterion for reducing the dimensionality of the cartesian product of the two mixture models. We have also introduced a technique for hybrid supervised/unsupervised classification, based on our merging algorithm, that can improve the performance of supervised classification using consensus from different features.

Acknowledgments

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

This work was originated by fruitful discussions with Eric Mjolsness and Becky Castano. Matt Klimesh is kindly acknowledged for fruitful feedback.

Appendix

Proof of Fact 1. A class–clipping operation can always be implemented by a sequence of class–clippings involving two classes at a time. We show in the following that the mean entropy can never increase with any such step. Assume classes j and k are clipped together; the variation $\Delta_s(z)$ of the entropy of the posterior distribution $P(j|z)$ is equal to

$$\begin{aligned}
&\Delta_s(z) = \\
&= -(P(j|z) + P(k|z)) \log(P(j|z) + P(k|z)) \\
&\quad + P(j|z) \log P(j|z) + P(k|z) \log P(k|z) \\
&= -P(j|z) \log\left(1 + \frac{P(k|z)}{P(j|z)}\right) - P(k|z) \log\left(1 + \frac{P(j|z)}{P(k|z)}\right) \\
&\leq 0
\end{aligned}$$

Since the variation of the mean entropy is equal to the expectation of $\Delta_s(z)$, the claim is proved.

Proof of Fact 2. We just need to prove the claim for the case $\alpha < 1$. The proof is based on the following two results.

Lemma 1. The entropy of a probability distribution increases if two values of the distribution are moved closer to each other, while the other values are left untouched.

Proof. The claim is a direct consequence of the convexity of the function $x \log x$.

Corollary 1. Let $P(j), 1 \leq j \leq N$ be a probability distribution and, for a given $K < N$, let J_1 and J_2 be the sets of the indices of the K smallest values and of the $N - K$ largest values of $P(j)$ respectively. Now form a new distribution $\bar{P}(j)$ from $P(j)$ by increasing some of the values with index in J_1 while at the same time decreasing some of the values with index in J_2, with the requirement that

$$\max\{\bar{P}(j), j \in J_1\} \leq \min\{\bar{P}(i), i \in J_2\}$$

Then the entropy of $\bar{P}(j)$ is higher than the entropy of $P(j)$.

Proof. The transformation from $P(j)$ to $\bar{P}(j)$ can be decomposed into a sequence of steps, each one involving just one value with index in J_1 and just one value with index in J_2. Therefore, by Lemma 1, the entropy is increased at each such step.

Now, it is easy to prove that the function $c(z)x^\alpha - x$, with $c(z)$ defined in (6), vanishes in correspondence of the value $x = c(z)^{\alpha-1}$, which is located between the smallest and the largest values of $P(j|z)$. Therefore, if $P(j|z)$ has non-null entropy, the equalization operator (5) with $\alpha < 1$ falls into the class of transformations considered in Corollary 1: the set J_1 is composed by all the j such that $P(j|z) \leq c(z)^{\alpha-1}$, the set J_2 is composed by all the other indices. This proves that for any z the entropy of $P(j|z)$ increases as a consequence of equalization with $\alpha < 1$.

References

[1] S. Belongie, C. Carson. H. Greenspan, J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. 675–682, *6th ICCV*, New Delhi, India, January 1998.

[2] C.M. Bishop. *Neural networks for pattern recognition*. Clarendon Press, Oxford, U.K., 1995.

[3] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, August 1996.

[4] R.M. Neal and G.E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Submitted to *Biometrika*.

[5] Y. Rubner, L. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. *Proc. ARPA Image Understanding Workshop*, May 1997.

[6] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. *Proc. IEEE CVPR'96*, 321–326, San Francisco, 1996.