A new approach to image segmentation with two-dimensional hidden Markov models

Josef Baumgartner
FCEFyN - UNC
Córdoba, Argentina
jbaumgartner@efn.uncor.edu

Ana Georgina Flesia
FAMAF - UNC, Conicet - UTN
Córdoba, Argentina
flesia@famaf.unc.edu.ar

Javier Gimenez
FAMAF - UNC
Córdoba, Argentina
jgimenez@famaf.unc.edu.ar

Julián Pucheta
FCEFyN - UNC
Córdoba, Argentina
jpucheta@efn.uncor.edu

***Abstract*—Image segmentation is one of the fundamental problems in computer vision. In this work, we present a new segmentation algorithm that is based on the theory of two-dimensional hidden Markov models (2D-HMM). Unlike most 2D-HMM approaches we do not apply the Viterbi Algorithm, instead we present a computationally efficient algorithm that propagates the state probabilities through the image. This approach can easily be extended to higher dimensions. We compare the proposed method with a 2D-HMM standard algorithm and Iterated Conditional Modes using real world images like a radiography or a satellite image as well as synthetic images. The experimental results show that our approach is highly capable of condensing image segments. This gives our algorithm a significant advantage over the standard algorithm when dealing with noisy images with few classes.**

I. Introduction

Recently hidden Markov models (HMM) have gained the attention of researchers from different fields. Still the classical HMM are generally limited to those areas where the observed data has only one dimension such as speech recognition [1], the analysis of genome data [2] or edge detection [3]. There have been early attempts to use HMM for higher-dimensional tasks like image segmentation but only for the price of previously converting the two-dimensional data of an image into a single vector by lining up the rows or the columns of the image [4]. The drawback of such an ordering is clearly the loss of information because adjacent pixels in the original image are torn apart.

In the last years efforts were made to extend the classical one-dimensional HMM to higher dimensions [5]. The problem hereby is, that the standard method of parameter estimation for one-dimensional HMM, the Baum-Welch Algorithm [6], is not feasible for higher dimensions. Hence, the main issue is to reduce the computational complexity in order to keep the n-dimensional HMM feasible. Hereby, one of the most promising approaches is the so called Viterbi Training [7], [8].

In this work we introduce a new method called *Complete Enumeration Propagation* (CEP) and compare it with *Path-Constrained Viterbi Training* (PCVT) as presented by [9] and Iterated Conditional Modes (ICM) as presented by [10]. The PCVT is a well established 2D-HMM algorithm for image segmentation, [3], [11]. Like most 2D-HMM approaches the PCVT has shown good results for noisy images that can be considered a Gaussian Mixture Markov Random Field [7], [12]. The other benchmark algorithm – ICM – is a well studied segmentation technique evaluated in many works like [13], [14]. To compare PCVT, ICM and CEP we use several real world images with a low signal-to-noise ratio and run two experiments with synthetic images.

This paper is organized as follows: In section 2 we present the mathematical background of a 2D-HMM and explain why further assumptions are necessary to make 2D-HMM feasible. Thereafter, we present two feasible approximations of a complete 2D-HMM in section 3: PCVT and CEP. In section 4 we evaluate the results of PCVT, ICM and CEP for the test images using Cohen's Kappa coefficient [15]. Finally we discuss the results in section 5 where we give a clear recommendation when to use PCVT and when to prefer CEP.

II. Theory of two-dimensional Hidden Markov Models

Two-dimensional data – like the pixels of an image – can be handled by a 2D-HMM if we assume the data to be a Markov Random Field. This means, that, given the image, the hidden state of pixel (i, j) is conditionally independent of the pixels outside a certain neighborhood. For pixel (i, j) we define $(i^{'}, j^{'}) \prec (i, j)$ if $i^{'} < i$ or $i^{'} = i$ and $j' < j$. It can be shown that this definition leads to a 2nd order Markov Mesh which specifies for state $s_{i,j}$:

$$P(s_{i,j}|s_{i',j'} : (i^{'}, j^{'}) \prec (i, j)) = P(s_{i,j}|s_{i,j-1}, s_{i-1,j}).$$

In figure 1 the relevant pixels of the 2nd order Markov Mesh are shown. The two pixels $(i, j-1)$ and $(i-1, j)$ can be understood as the "past" of pixel (i, j). In other words we are moving from the top-left pixel to the bottom-right pixel. Hence, the initial probabilities for the 2D-HMM depend only on the first state $s_{0,0}$ and we can wright

$$\pi_l = P(s_{0,0} = l) \qquad \forall l \in \mathcal{S}.$$

Next, we assume that the observed pixel intensities of one class are normally distributed. For the sake of completeness let us consider multispectral images where each pixel is a vector from R^k so we can calculate the emission probabilities of state

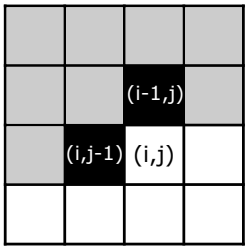

Fig. 1. Transitions among states in a 2nd order Markov Mesh. The gray and the black pixels fulfill $(i^{'}, j^{'}) \prec (i, j)$ but the two black pixels are sufficient statistics for pixel (i, j) under the Markov assumption.

$l \in \mathcal{S}$ with mean μ_l and covariance matrix Σ_l:

$$\begin{aligned} b_l(x) &= P(x|s_{i,j} = l) \\ &= \frac{1}{(2\pi)^{k/2}|\Sigma_l|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_l)^T\Sigma_l^{-1}(x-\mu_l)\right\} \end{aligned}$$

Besides that, we consider $P(s_{i,j}|s_{i,j-1}, s_{i-1,j})$ to be independent of the current pixel so we can gather the transition probabilities in a matrix A where

$$a_{m,n,l} = P(s_{i,j} = l|s_{i,j-1} = m, s_{i-1,j} = n).$$

These assumptions are the basis on which we try to find the optimal hidden state map s^*

$$s^* = \arg\max_s P(s|O, \theta). \tag{1}$$

where O are the observations and s is any admissible hidden state map. In θ we gather all the parameters of a 2D-HMM which are: the inicial probabilities, the means and standard deviations of each hidden state and the transition probabilities A.

The exact formulas for the parameters are:

$$\mu_m = \frac{\sum_{i,j} L_m(i,j) u_{i,j}}{\sum_{i,j} L_m(i,j)} \tag{2}$$

$$\Sigma_m = \frac{\sum_{i,j} L_m(i,j) \left(u_{i,j} - \mu_m\right)\left(u_{i,j} - \mu_m\right)^T}{\sum_{i,j} L_m(i,j)} \tag{3}$$

$$a_{m,n,l} = \frac{\sum_{i,j} H_{m,n,l}(i,j)}{\sum_{l'=1}^{M} \sum_{i,j} H_{m,n,l'}(i,j)} \tag{4}$$

where L and H are sums over all possible state maps. See [16] for the details of the formulas. Note, that for an image of size $(w \times z)$ with M hidden states $\mathcal{S} = \{1, 2, \ldots, M\}$ there are M^{w*z} possible hidden state maps. This huge number of state maps, even for small images, leads to infeasibility because L and H can not be calculated.

To solve the problem of computational complexity several algorithms were proposed in recent years [7], [8]. Almost all of them apply the Viterbi Algorithm [17] in some way. In the next section we describe an advanced version of the Viterbi Training, the *Path-Constrained Viterbi Algorithm* [9] and present a new algorithm called *Complete Enumeration Propagation*.

III. Approaches to two-dimensional Hidden Markov Models

In this section we will introduce two methods that approximate the optimal hidden state map as defined by equation 1. Note, that both methods use almost the same assumptions but only the PCVT runs the Viterbi Algorithm.

A. Path-Constrained Viterbi Training

Based on the theoretical foundations from section II the PCVT can be derived as follows. First of all remember our notion of "past" as shown by figure 1. We now consider each diagonal of the image as one step in time, starting with the top-left pixel. Thus, the diagonals $T_0, T_1, T_2 \ldots$ are

$$T_0 = (s_{0,0}); \quad T_1 = (s_{1,0}, s_{0,1}); \quad T_2 = (s_{2,0}, s_{1,1}, s_{0,2}); \quad \ldots$$

Because we are dealing with a 2nd order Markov Mesh we can make the Markov assumption and get

$$\begin{aligned} P(s) &= P(T_0)P(T_1|T_0)\ldots P(T_{z+w-2}|T_{z+w-3}, \ldots, T_0) \\ &= P(T_0)P(T_1|T_0)\ldots P(T_{z+w-2}|T_{z+w-3}). \end{aligned} \tag{5}$$

Note in equation (5), that each diagonal operates as an "isolating" element between neighboring diagonals. Hence, we have transformed the complex two-dimensional model to a pseudo one-dimensional HMM. The problem we are facing here, is that each diagonal consists of up to $\min(w, z)$ states: $T_0 \in \mathcal{S},\ T_1 \in \mathcal{S}^2,\ T_2 \in \mathcal{S}^3,\ \ldots,\ T_{z+w-2} \in \mathcal{S}$.

From now on we denote each combination of states on one diagonal a sequence. Keep in mind, that for M states a diagonal can have up to $M^{\min(w,z)}$ sequences – a number generally too high to be feasible.

The first step to simplify the computation of the 2D-HMM is to reduce the number of sequences on each diagonal to N. If we set N to a value much smaller than $M^{\min(w,z)}$ we have drastically reduced the computational burden, but the question arises: How do we select the N sequences? For the moment we assume that we can evaluate the posterior of a given diagonal state sequence by simply multiplying the posteriors of each pixel without considering statistical dependencies between pixels, i.e.

$$P(s_{i,j} = l|O_{i,j}, \theta) \propto P(O_{i,j}|s_{i,j} = l, \theta)P(s_{i,j} = l|\theta).$$

By doing so, it is computationally easy to classify the possible sequences as more or less probable. Once we have evaluated the posteriors of all sequences of one diagonal we keep the most likely N sequences and forget about the rest. This is clearly a significant simplification but even though we run the risk of throwing away the sequence that belongs to the optimal hidden state map s^* we expect to keep at least some sequences that are close to the optimal one.

After cutting the number of state sequences on each diagonal down to N we are ready to run the Viterbi Algorithm. We call each diagonal state sequence $\mathbf{s}_{d,k}$ where d is the index for the diagonal with $d = 0, 1, \ldots, z+w-2$ and $k = 1, 2, \ldots, N$ indicates the state sequence. The initial state probabilities $\tilde{\pi}_k$ for pixel $(0, 0)$ are

$$\tilde{\pi}_k = P(T_0 = \mathbf{s}_{0,k}).$$

We denote $\delta_d(l)$ the maximum joint probability of the observations $\mathbf{O}_0, \ldots, \mathbf{O}_d$ and sequences from T_0 to T_d, where l is a certain sequence on diagonal d . Given the parameters of the 2D-HMM we can write

$$\delta_d(l) = \max_{k_0,\ldots,k_{d-1}} P(\mathbf{s}_{0,k_0}, \ldots, \mathbf{s}_{d-1,k_{d-1}}, \mathbf{s}_{d,l}, \mathbf{O}_0, \ldots, \mathbf{O}_d|\theta), \quad for \quad d = 0, \ldots, z+w-2; \quad l = 1, \ldots, N. \tag{6}$$

Furthermore we collect the pixels on diagonal d in a variable $\Delta(d)$ and define

$$b_{\mathbf{s}_{d,k}}(\mathbf{O}_d) = \prod_{(i,j)\in\Delta(d)} b_{\mathbf{s}_{d,k}(i,j)}(O_{i,j}) \tag{7}$$

where $b_{\mathbf{s}_{d,k}}(\mathbf{O}_d)$ is the emission probability of sequence k on diagonal d under the assumption that each pixel is statistically independent from its neighbors. Finally we can calculate the transition probability from sequence k on diagonal d to sequence l on diagonal $d+1$:

$$\begin{aligned}\tilde{a}_{d,k,l} &= P(T_{d+1} = \mathbf{s}_{d+1,l} | T_d = \mathbf{s}_{d,k}, \theta) \\ &= \prod_{(i,j)\in\Delta(d+1)} a_{\mathbf{s}_{d,k}(i-1,j),\mathbf{s}_{d,k}(i,j-1),\mathbf{s}_{d+1,l}(i,j)} \\ &for \quad d = 0, \ldots, z+w-3; \quad k, l = 1, \ldots, N.\end{aligned} \tag{8}$$

In figure 3 an intuition of what $\tilde{a}_{d,k,l}$ stands for is given. Now we are ready to initialize the *Viterbi Algorithm* with the values

$$\delta_0(k) = P(T_0 = \mathbf{s}_{0,k}), \quad b_{\mathbf{s}_{0,k}}(\mathbf{O}_0) = \tilde{\pi}_j b_{\mathbf{s}_{0,j}}(O_{0,0}) \quad \forall k = 1, 2, \ldots, N.$$

Then we start the recursion using equations (6), (7) and (8)

$$\delta_{d+1}(l) = \left[\max_{1\le k\le N} \delta_d(k)\tilde{a}_{d,k,l}\right] b_{\mathbf{s}_{d+1,l}}(\mathbf{O}_{d+1}) \quad \forall d = 0, 1, \ldots, z+w-3 \quad \forall l = 1, 2, \ldots, N.$$

After each step we save the index of the most probable sequence on diagonal d that leads to sequence l on diagonal $d+1$ in a variable called φ:

$$\varphi_{d+1}(l) = \arg\max_{1\le k\le N}\{\delta_d(k)\tilde{a}_{d,k,l}\} \quad \forall d = 0, 1, \ldots, z+w-3 \quad \forall l = 1, 2, \ldots, N$$

When the algorithm reaches the last diagonal we use the values saved in φ to track back the most probable path through the image starting with the bottom-right pixel

$$s^*_{z+w-2} = \arg\max_{1\le k\le N} \delta_{z+w-2}(k)$$

$$s^*_d = \varphi_{d+1}(s^*_{d+1}) \qquad \forall d = z+w-3, z+w-4, \ldots, 1$$

The final result s^* contains the optimal path through the N sequences at each diagonal. Note that this is equal to knowing the complete hidden state map for the whole image. In figure 2 an example of the PCVT is shown.

Once we know the hidden state of every pixel we can update the parameters of the 2D-HMM. Instead of the exact formulas 2, 3 and 4 we use approximate formulas for iteration step p, where $I(\cdot)$ is the indicator function. One can think of these simplified formulas as "count instead of evaluate":

$$\mu_l^{(p)} = \frac{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right) O_{i,j}}{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right)} \tag{9}$$

$$\Sigma_l^{(p)} = \frac{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right) (O_{i,j}-\mu_l)(O_{i,j}-\mu_l)^T}{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right)} \tag{10}$$

$$a_{n,m,l}^{(p)} = \frac{\sum_{i=1}^{z-1}\sum_{j=1}^{w-1} I\left(s_{i-1,j}^{(p-1)} = n, s_{i,j-1}^{(p-1)} = m, s_{i,j}^{(p-1)} = l\right)}{\sum_{i=1}^{z-1}\sum_{j=1}^{w-1} I\left(s_{i-1,j}^{(p-1)} = n, s_{i,j-1}^{(p-1)} = m\right)} \tag{11}$$

After updating a, μ and Σ we run the PCVT with the new parameters and iterate until convergence. In summary we can describe the segmentation algorithm as

Algorithm 1: Path-Constrained Viterbi Training (PCVT)

1) Initialize parameters μ_l and Σ_l for $l \in \mathcal{S}$.
2) Initialize state map using Maximum Likelihood Classification.
3) Calculate transition probabilities $a_{n,m,l}$ for every $n, m, l \in \mathcal{S}$ using equation (11).
4) Choose the best N state sequences for each diagonal and run Viterbi decoding.
5) Update parameters $a_{n,m,l}$, μ_l and Σ_l using equations (9), (10) and (11).
6) Iterate steps 4) and 5) until convergence.

In algorithm 1 step 4) will take time in the order of $\mathcal{O}((2w-1)N^2)$ for an image of size $w \times w$ whereas step 5) requires no remarkable computational effort. In section V we compare the complexity of the PCVT and other characteristics with a new algorithm that we present in the next section.

B. Complete Enumeration Propagation

In this section we present a new algorithm, called CEP, that estimates the parameters of a 2D-HMM and finds a suboptimal solution of the hidden state map. Like in the case of the PCVT we suppose, that the image is a 2nd order Markov Mesh with the past states shown in figure 1. Besides that, we consider the emission probability of pixel (i, j) to depend only on the current state. Furthermore we assume, that the transition probabilities do no depend on the current pixel, or, in other words, we will use the same transition matrix A like before. With these assumptions we get for the state of pixel (i, j):

$$\begin{aligned}P(s_{i,j}|s_{i,j-1}, s_{i-1,j}, O_{i,j}) &\propto P(s_{i,j}, s_{i,j-1}, s_{i-1,j}, O_{i,j}) \\ &= P(s_{i,j-1}, s_{i-1,j}) * P(s_{i,j}|s_{i,j-1}, s_{i-1,j}) P(O_{i,j}|s_{i,j})\end{aligned}$$

If we now replace $P(s_{i,j}|s_{i,j-1}, s_{i-1,j})$ by the transition matrix A and consider two diagonal pixels to be independent we can write

$$\begin{aligned}&P(s_{i,j}|s_{i,j-1}, s_{i-1,j}, O_{i,j}) \propto \\ &P(s_{i,j-1}) * P(s_{i-1,j}) * a_{s_{i,j-1},s_{i-1,j},s_{i,j}} * P(O_{i,j}|s_{i,j})\end{aligned} \tag{12}$$

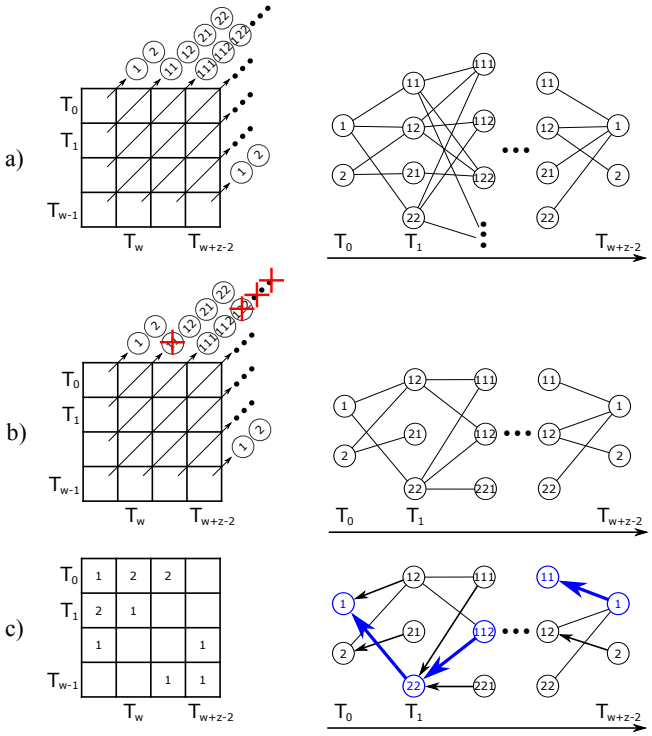

Fig. 2. Example of Path-Constrained Viterbi Training for two possible states and $N = 3$. a) complete 2D-HMM (infeasible); b) number of sequences on each diagonal is constrained to 3; c) tracking back the optimal path of state sequences.

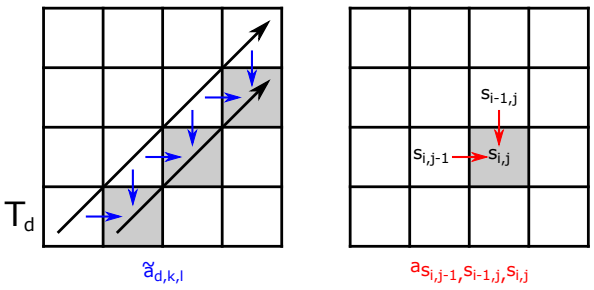

Fig. 3. While PCVT uses $\tilde{a}_{d,k,l}$ which represents the transition probability from one diagonal to another, CEP applies the transition probability $a_{s_{i,j-1},s_{i-1,j},s_{i,j}}$ for each pixel.

This is the main formula to calculate the state probabilities of pixel (i, j) given the observation and the two past states. In figure 3 the difference between CEP and PCVT is illustrated. Note, that the PCVT makes the same assumption when searching for the N best sequences.

The main idea behind CEP is to use equation (12) to calculate $P(s_{i,j} = l)$ for $l = 1, 2, \ldots, M$ for all possible combinations of past states, i.e. $s_{i,j-1} = m$, $s_{i-1,j} = n$ for $m, n = 1, 2, \ldots, M$ according to equation (13).

$$P(s_{i,j} = l|O_{i,j}) \propto \sum_{m=1}^{M}\sum_{n=1}^{M} a_{s_{i,j-1}=m,s_{i-1,j}=n,s_{i,j}=l} \quad (13)$$
$$P(s_{i,j-1} = m)P(s_{i-1,j} = n)P(O_{i,j}|s_{i,j} = l)$$

This procedure is nothing else than complete enumeration of $P(s_{i,j} = l)$. Keep in mind, that before we can go on with the next pixel it is necessary to normalize $P(s_{i,j} = l)$ such that $\sum_{l=1}^{M} P(s_{i,j} = l) = 1$. Hence the CEP algorithm can be described as

Algorithm 2: Complete Enumeration Propagation (CEP)

1) Initialize parameters μ_l and Σ_l for $l \in \mathcal{S}$.
2) Initialize state map using Maximum Likelihood Classification.
3) Calculate transition probabilities $a_{n,m,l}$ for every $n, m, l \in \mathcal{S}$ using equation (11).
4) Find new state map using equations (12) and (13).
5) Update parameters $a_{n,m,l}$, μ_l and Σ_l using equations (9), (10) and (11).
6) Iterate steps 4) and 5) until convergence.

Calculation of step 4) of algorithm 2 is of order $\mathcal{O}(w^2M^3)$ for an image of size $w \times w$ with M states. A problem arises for the pixels on the left and upper edge of the image because there are no past states $s_{i,j-1}$ or $s_{i-1,j}$. To solve this issue one can think of two possible solutions. First, copy the first row and the first column and use maximum likelihood to determine the probabilities of these auxiliary pixels. Second, suppose a uniform distribution for the nonexistent terms $P(s_{i,j-1} = m)$ and $P(s_{i-1,j} = n)$. This is equal to leaving out the corresponding terms in equation (12). We prefer the second option because otherwise noisy observations on the edges are encouraged to stay in a maximum likelihood state instead of adapting themselves to their neighborhood.

Once we have calculated the probabilities of all the pixels we assign each pixel the most probable state. The result is a hidden state map which, for now, is our best guess of $s^* = \arg\max_s P(s|O, \theta)$. From this point on we use the formulas 9, 10 and 11 from the PCVT-framework to update the parameters of the 2D-HMM. Then we iterate this procedure until convergence. In the next section the experimental results for PCVT and CEP are shown.

IV. Experimental Results: Image Segmentation

In this section we present the experimental results. To evaluate the algorithms we use Cohen's $\widehat{\kappa}$ coefficient [15] which is defined as

$$\widehat{\kappa} = \frac{P_O - P_E}{1 - P_E}.$$

where $P_O = \sum_{i=1}^{k} p_{ii}$ is the relative observed agreement among segmented image and ground truth and $P_E = \sum_{i=1}^{k} p_{i+}p_{+i}$ is the hypothetical probability of chance agreement.

Besides PCVT and CEP, we will present the results of Maximum Likelihood Classification (ML) and Potts- Iterated Conditional Modes (ICM). While ML, as described in algorithm 3, is a classical non-contextual classification method, ICM, as described in algorithm 4, is a well known contextual

Algorithm 3: Maximum Likelihood Classification

1) Initialize parameters μ_l and Σ_l for $l \in \mathcal{S}$.
2) Calculate $P(s_{i,j} = l|I_{i,j}, \theta) = P(I_{i,j}|s_{i,j} = l, \theta)$ for each pixel (i, j) and for each state l.
3) Assign pixel (i, j) the label given by $s_{i,j} = \arg\max_{l \in \mathcal{S}} P(I_{i,j}|s_{i,j} = l, \theta)$.

algorithm. Geman and Geman [18] consolidated the use of Gibbs laws as prior evidence in the processing and analysis of images. Such distributions are able to capture the spatial redundancy of the visual information in a tractable manner. Among them, the Potts model has become a commonplace for describing classes. ICM is an iterative algorithm that rapidly converges to the local maximum of the function $P(s|I, \theta)$ closest to the initial segmentation provided by the user. In this work the initial segmentation for ICM is provided by ML and the parameter β is estimated as described in [19]. The

Algorithm 4: Iterated Conditional Modes (ICM)

1) Initialize parameters μ_l and Σ_l for $l \in \mathcal{S}$.
2) Maximum Likelihood segmentation of I.
3) Estimate parameter β.
4) Choose a pixel's visit scheme for the image.
5) For each pixel (i, j), change the label given in the previous iteration for the label $l \in \mathcal{S}$ that maximizes

$$g(l) = \ln p(I_{i,j}|l, \mu_l, \Sigma_l) + \beta U_{i,j}(l) \qquad (14)$$

where $U_{i,j}(l)$ is the number of pixels in the neighborhood of (i, j) with hidden state l.
6) Iterate step 5) until convergence.

first term of equation (14) is equivalent to the ones used by the ML classifier. The second term is the contextual component scaled by the parameter β. If $\beta > 0$, ICM smooths out the initial segmentation, if $\beta < 0$, ICM reduces clusters coherence and for $\beta = 0$ the rule is reduced to ML.

We propose five different scenarios to evaluate and compare ML, ICM, PCVT and CEP. The experiments consist of three real images – an inverse digitalized X-ray image, a multispectral optical Landsat image and a standard test image – as well as an artificial image and a database of 300 synthetic images, where each image contains between two and six classes.

A. *Multimodal X-ray image*

First we use an image from the field of Diagnostic Radiography. Due to the sensing method and the posterior digitization process, this type of imagery has a very low signal-to-noise ratio [20]. It shows a Wistar rat's jaw and forms part of a growth study of rats [21]. The interesting aspect about this image is that it contains four mixed classes (bone and tooth, tissue and flesh, cartilage and background) which can also be grouped into two or three super-classes, corresponding to the smooth modes of the intensity histogram. We run ML, ICM, PCVT and CEP for all three cases (four classes, two or three

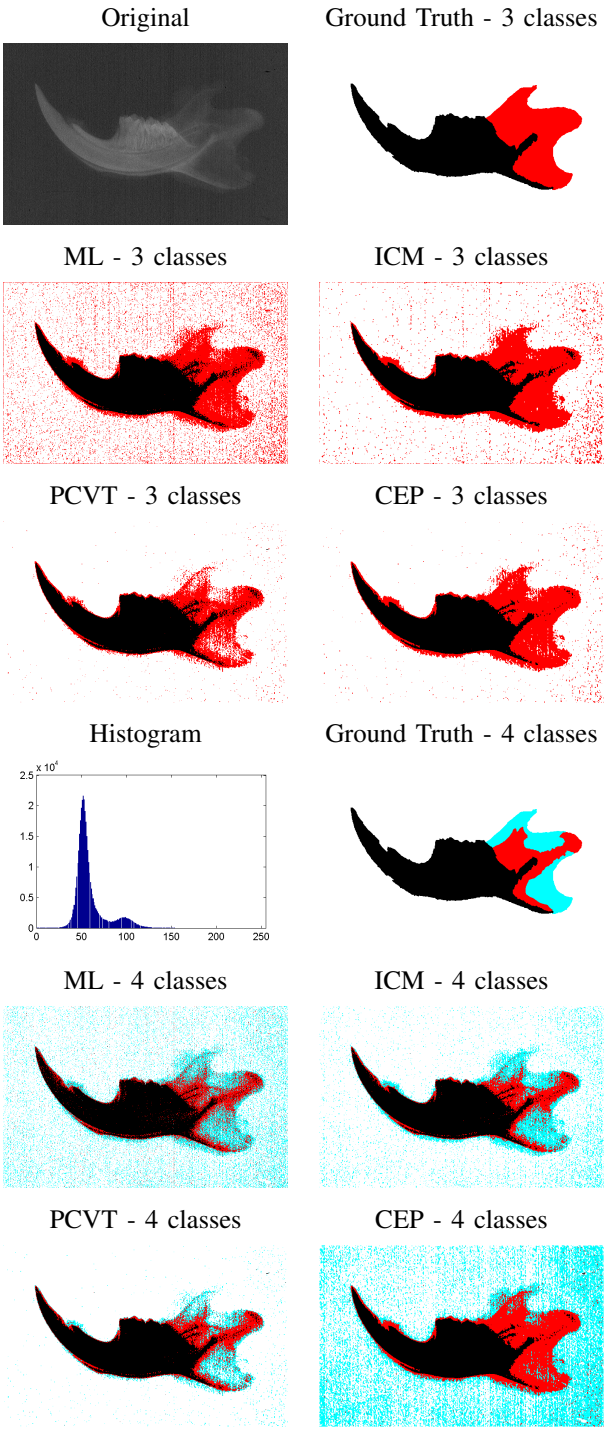

Fig. 4. Segmentations of an X-ray image.

super-classes) and evaluate the performance with kappa. Some of the segmentation results of this scenario are shown in figure 4. The evaluation of this experiment is shown in figure 5.

B. *Multispectral, multimodal satellite image*

The second experiment is a multispectral Landsat TM image of an agricultural area in the humid pampa of Argentina. It

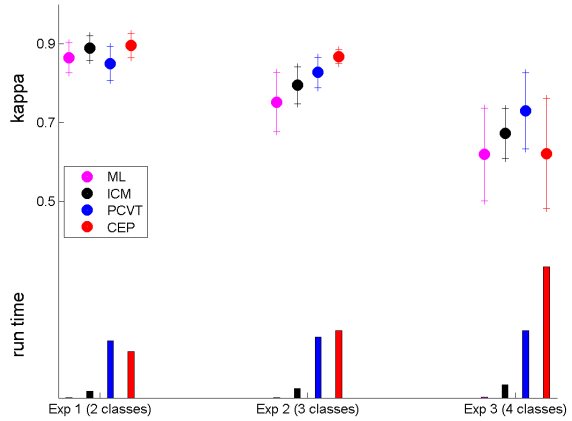

Fig. 5. Segmentation of an X-ray image. The image can be considered to have two, three or four classes. The dots are the mean $\widehat{\kappa}$ while the vertical lines indicate the standard deviations that were obtained by running the algorithms with different initial conditions and by pre-filtering the image. For the experiment with three classes CEP clearly outperforms the other methods whereas PCVT shows the best performance for four classes. Note, that the computational complexity of CEP rises with the number of states.

shows agricultural fields of different sizes and orientations and two center-pivot irrigations. In this case, the performance is evaluated in the parts of the image that are shown in figure 6, since we only have ground truth labels for these regions. The ground truth data corresponds only to three classes, but we have no information of how many classes are present in whole the image, so we evaluate all algorithms for three to eight classes and show the results in figure 7.

C. Synthetic imagery

We also test ML, ICM, PCVT and CEP on a database of synthetic images that were generated by a causal Hidden Markov Model. The database contains 300 synthetic images, arranged in groups of 15 images. To generate the 300 images we proposed between two and six classes and observed the state maps for four different normal distributions ($15 * 5 * 4 = 300$). The exact procedure to obtain the synthetic images is described in algorithm 5.

Some of the images segmented by ML, ICM, PCVT and CEP are presented in figure 8 (three classes, means of classes close to each other) and figure 9 (four classes, means of classes well separated). In figure 10 an overall evaluation is given. Note, that the closer the class-means the worse the performance of CEP.

D. Binary image

In this experiment we try to segment the logo of the National University of Córdoba. The ground truth and some segmentations of this two-class problem are shown in figure 11. Like in the case of the synthetic images we vary the observation-means of the two classes "logo" and "background". As a result we get images that are hard to segment for close means and easy to segment for separate classes.

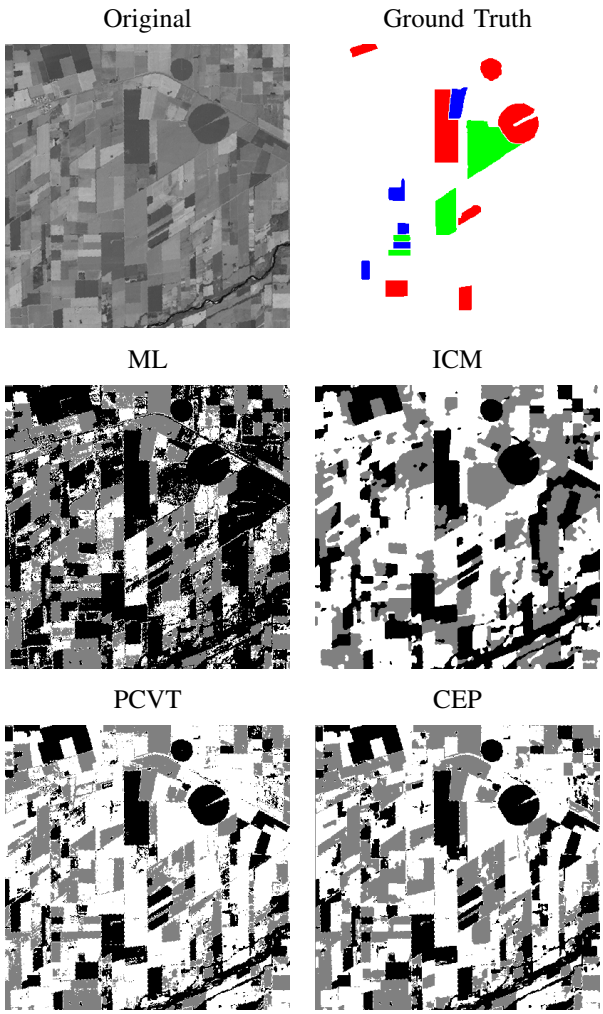

Fig. 6. Segmentations of band five of a Landsat image using three classes.

Algorithm 5: Synthetic images

1) For M classes draw M^3 random numbers according to $r_i \sim N(0,1)$ for $i = 1, \ldots, M^3$.
2) Rearrange the drawn random numbers r_i in a matrix A with dimension $(M \times M \times M)$.
3) Set all negative values in A to zero and normalize A such that $\sum_{n=1}^{M} \sum_{m=1}^{M} a_{n,m,l} = 1$.
4) Simulate the (10×10) state map of a causal 2D-HMM using the transition probability matrix A.
5) Extend the state of each pixel to a region of 10×10 pixels. Hence the complete state map will have dimension (100×100).
6) Observe the state map through a normal distribution.

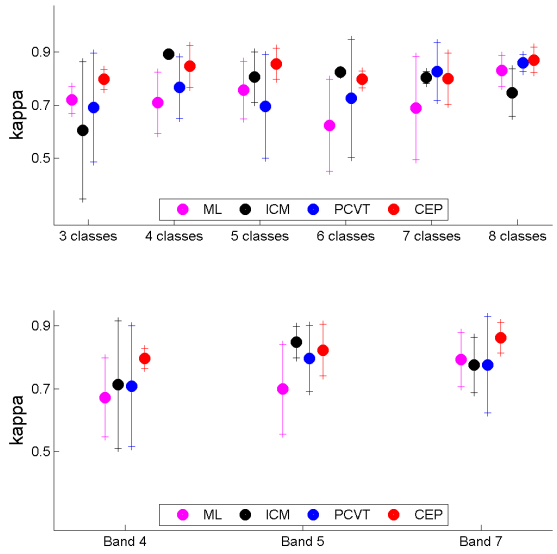

Fig. 7. Evaluation of Landsat satellite images for bands 4, 5 and 7 using between 3 and 8 classes. The dots are the mean $\widehat{\kappa}$ while the vertical lines indicate the standard deviations.

We observe that CEP has problems separating the two classes when their means are close to each other but shows excellent results for class-means that vary more than eight. The complete evaluation of this experiment is presented in figure 12.

E. Standard test image

Finally we evaluate the segmentation algorithms for a more complex image, denominated "Cameraman". There is no texture in the image, besides a slight decoloration in the sky behind the cameraman. We consider five classes in this image and we initialized the algorithms with the same reference sample. Maximum Likelihood Classification (ML) is the initial point of all contextual methods. In figure 13 we show all segmentations. Note, that CEP is the only method that finds the building in the background, but its sensitivity also makes it highlight shadows in the sky behind the cameraman, most likely produced by jpeg compression. Another point is, that CEP - just like ICM - shows very good results when it comes to segmenting the solid ground of the picture, whereas ML and PCVT do not assign one unique class to the ground.

To evaluate this experiment we show the intensity histogram of the original picture and the normal distributions found by the four segmentation algorithms in figure 14. One can see that ICM does not end up far away from the initial classes given by ML. In contrast to that PCVT and CEP have made reasonable adjustments to the initial normal distributions. In the next section we discuss the experimental results and draw conclusions when to prefer PCVT and when to use CEP.

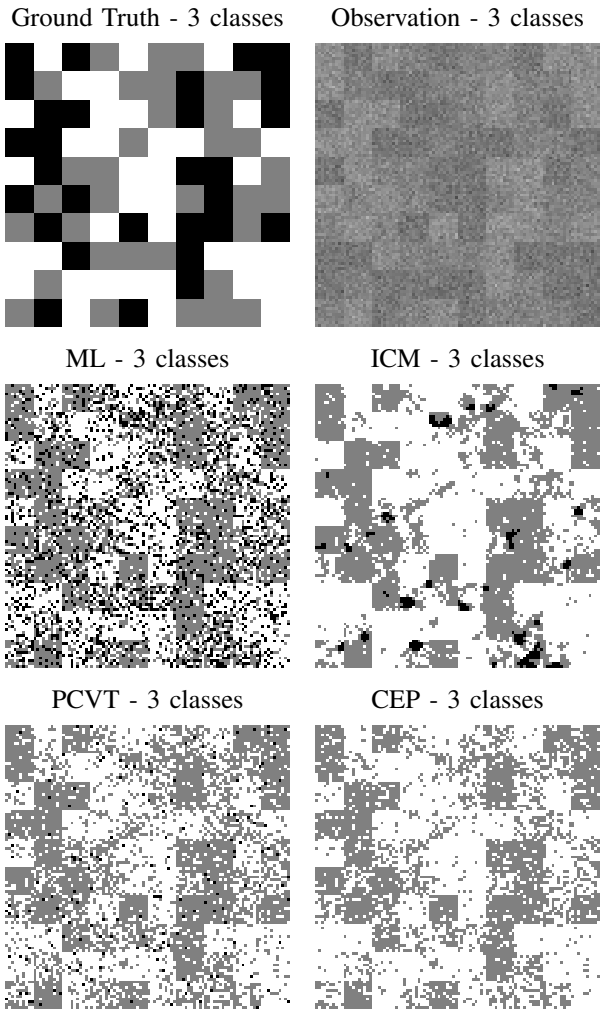

Fig. 8. Synthetic images with three classes observed at $C_1 \sim N(50, 25)$, $C_2 \sim N(55, 25)$ and $C_3 \sim N(60, 25)$.

V. Conclusions

The two main algorithms compared in this work - PCVT and CEP - use the same suppositions to make the 2D-HMM framework feasible. Still the most severe assumption of two diagonal pixels being independent is only necessary to find the N most probable sequences in the case of the PCVT, whereas the CEP is using this assumption permanently. On the other hand PCVT is intentionally throwing away state sequences hoping to keep the optimal sequence, or, at least some of the sequences that are close to the optimal one.

When it comes to the experimental results it can be seen that CEP outperforms ML, ICM and PCVT in all cases where the number of segmentation classes is small or when the normal distributions are well separated. Above all the staellite image and the artificial image demonstrate the strong sides of CEP, which are condensing a noisy image with only a few classes. Besides that, CEP was the only algorithm that recognized the building in the background of the benchmark

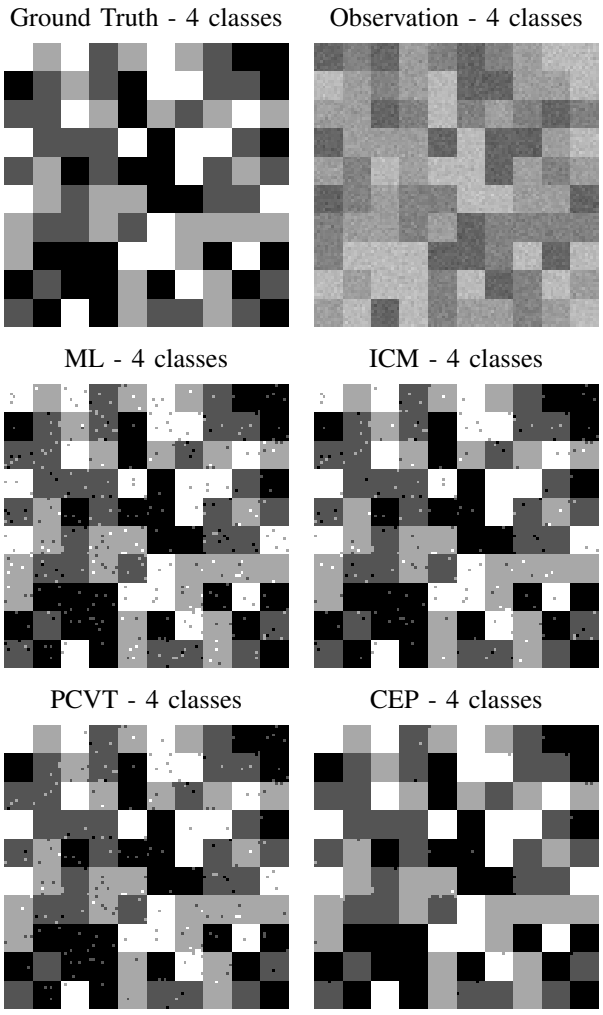

Fig. 9. Synthetic images with four classes observed at $C_1 \sim N(50, 25)$, $C_2 \sim N(70, 25)$, $C_3 \sim N(90, 25)$ and $C_4 \sim N(110, 25)$.

image "Cameraman". When analyzing the normal distributions found by CEP it can be seen that our algorithm adjusts very well to the underlying intesity histogram.

Still, for experiments with many classes CEP revealed some weaknesses, not only considering the $\widehat{\kappa}$ coefficient but also the run time. Especially in the case of the radiography with four classes the runtime and the performance of CEP is worse than the compared methods. In this particular experiment CEP needed more than 200 iterations to converge, which is a strong indicator that the algorithm has problems of finding a locally optimal hidden state map. Here, the PCVT has a clear advantage over ICM and CEP, because it pre-selects the possible sequences and thus finds a state map closer to the one obtained by ML. For noisy images with many hidden states this characteristic is favorable, while a too intense condensing of image segments can absorb other classes and lead to bad results.

In conclusion the CEP is not only a challenge to the PCVT and the other segmentation methods but also a complementary.

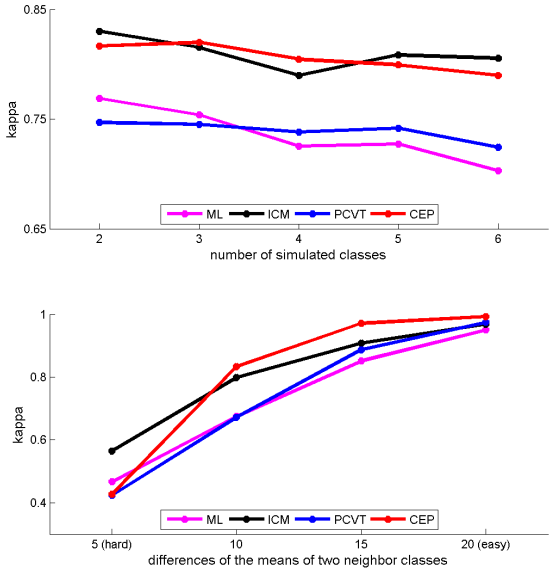

Fig. 10. Results of the segmentation of synthetic images. Above: Average $\widehat{\kappa}$ in function of the numbers of classes. Below: Average $\widehat{\kappa}$ for different observation distributions. The closer the means of two neighboring classes the harder the segmentation task.

Especially for simple images with a low signal-to-noise ratio the CEP should be preferred to the other presented algorithms. Besides that, the computational complexity of CEP - depending mainly on the number of states - is a great incentive to extend the 2D-CEP-framework to higher dimensions in future works.

ACKNOWLEDGMENT

This work has been partially supported by Argentinean grants ANPCyT-PICT 2008-00291 and Secyt UNC-PID 2012 05/B504. J.B. and J.G. are supported by a PhD student grant from CONICET. The authors would like to thank the National University of Córdoba as well as J. G. Flesia for the X-ray image.

REFERENCES

[1] M. Rabiner and S. Young, "The application of hidden markov models in speech recognition," in *Foundations and Trends in Signal Processing* ***1*** *(3)*, 2007, pp. 195 – 304.

[2] Y. Shen, Y. Gu, and I. P. er, "A hidden markov model for copy number variant prediction from whole genome resequencing data," in *Proc. of the BMC Bioinformatics* ***12*** *(Suppl 6) S4*, 2011.

[3] A. G. Flesia, J. Gimenez, and J. Baumgartner, "On segmentation with markovian models," XIV Argentine Symposium on Artificial Intelligence - ASAI, Crdoba, Argentina, Setiembre 2013.

[4] S.-S. Kuo and O. E. Agazzi, "Automatic keyword recognition using hidden markov models," *Journal of Visual Communication and Image Representation*, vol. 5(3), pp. 265 – 272, 1994.

[5] X. Ma, D. Schonfeld, and A. Khokhar, "Video event classification and image segmentation based on noncausal multidimensional hidden markov models," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 1304–1313, 2010.

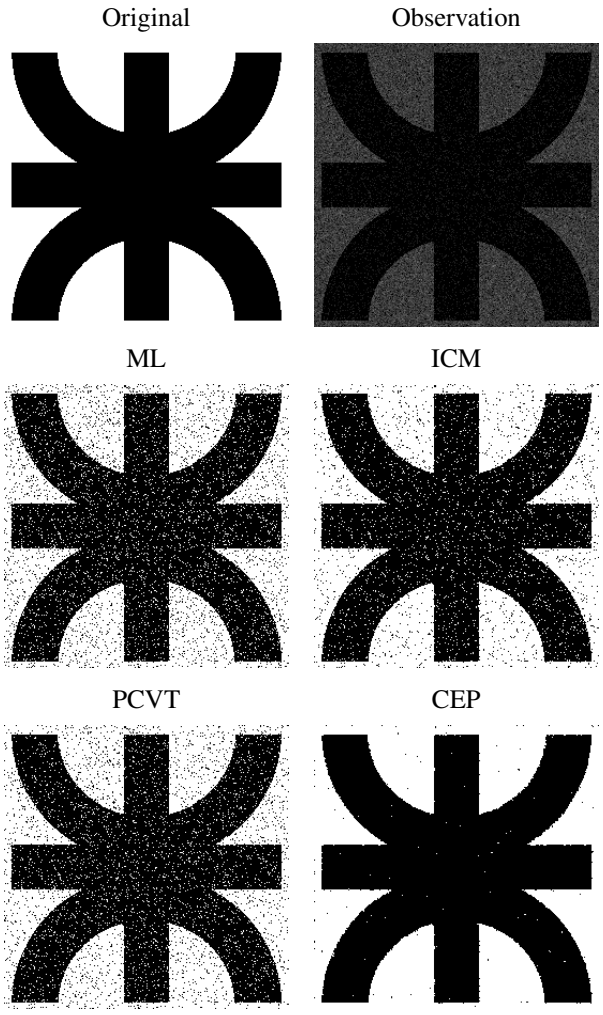

Fig. 11. Segmentation of the logo of the National University of Córdoba. In this case the two classes are observed according to $Logo \sim N(50, 25)$ and $Background \sim N(65, 25)$.

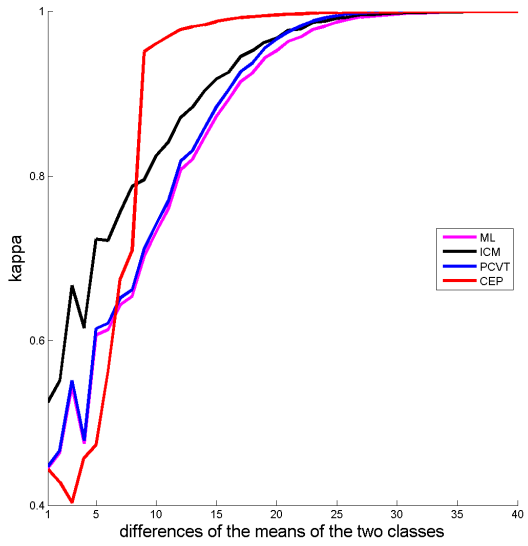

Fig. 12. Segmentations of a binary test image observed for different normal distributions. On the x-axis the difference between the two class-means are shown. In this experiment each class always has a standard deviation of 5. Note that the closer the Gaussians, the harder the segmentation task. Like in the case of simulated images CEP outperforms the other methods for separated class-means greater than eight.

[6] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains," in *Ann. Math. Stat 1*, 1970, pp. 164 – 171.

[7] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two dimensional hidden markov model," in *IEEE Transactions on Signal Processing*, vol. 48(2), 2000, pp. 517 – 533.

[8] X. Ma, D. Schonfeld, and A. Khokhar, "Image segmentation and classification based on a 2d distributed hidden markov model," in *Proc. SPIE 6822, Visual Communications and Image Processing*, 2008.

[9] D. Joshi, J. Li, and J. Z. Wang, "A computationally efficient approach to the estimation of two- and three-dimensional hidden markov models," in *IEEE Transactions on Image Processing*, vol. 15(7), 2006, pp. 1871 – 1886.

[10] J. E. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society*, vol. 48(3), p. 259 302, 1986.

[11] S. Y. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, 2012, article ID 814356, 18 pages, vol. 2012.

[12] K. P. Pyun, J. Lim, and R. M. Gray, "A robust hidden markov gauss mixture vector quantizer for a noisy source," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1385 – 1394, 2009.

[13] F. A. Huang, S. B. Narayan, D. B. Wilson, D. B. Johnson, and G. Q. Zhang, "A fast iterated conditional modes algorithm for water-fat decomposition in mri," *IEEE Transactions on Medical Imaging*, vol. 30(8), pp. 1480 – 1492, 2011.

[14] R. T. Collins, "Multitarget data association with higher-order motion models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 1744 1751, 2012.

[15] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20(1), pp. 37 – 46, 1960.

[16] J. Li and R. Gray, *Image Segmentation and Compression using Hidden Markov Models*. Kluwer Academic Publishers, 2000.

[17] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13(2), pp. 260 – 269, 1967.

[18] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721 – 741, 1984.

[19] J. Gimenez, A. C. Frery, and A. G. Flesia, "Inference strategies for the smoothness parameter in the potts model," *To appear in Proc. of the IEEE International Geoscience and Remote Sensing Symposium*, 2013.

[20] J. G. Flesia and A. G. Flesia, "The influence of processing in the accuracy of measurements in indirect digitalized intra-oral radiographic imaging for forensic applications," *The Forensic Oral Pathology Journal*, vol. 2(4), pp. 20 – 24, 2011.

[21] P. A. Fontanetti, P. M. Mandalunis, and N. T. Vermouth, "Response of bone tissue associated with eruption of rat first mandibular molar of pups born from mothers subjected to constant light during pregnancy," *Bone*, vol. 49, no. 6, pp. 1380 – 1385, 2011.

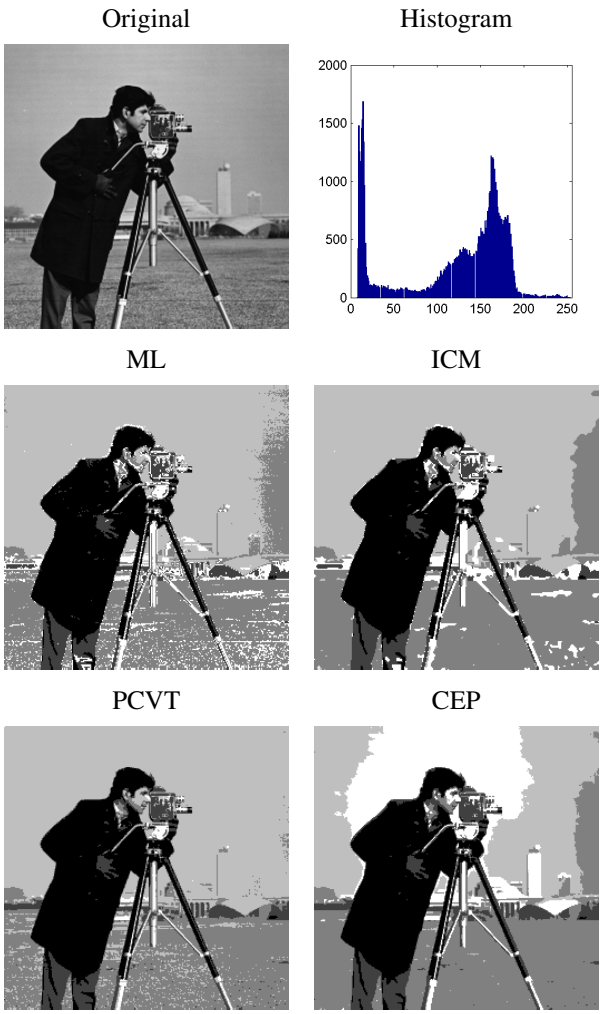

Fig. 13. Segmentations of a standard test image "Cameraman" using five classes.

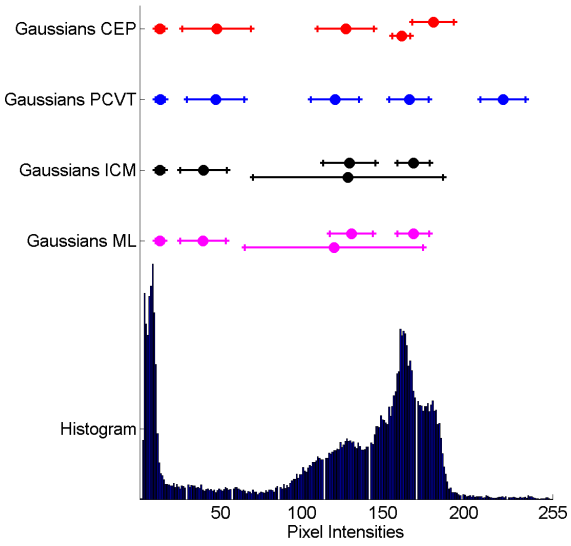

Fig. 14. Histogram of the test image shown in figure 13 and the means and standard deviations of the five classes of each segmentation method.