DOI 10.3233/ICA-2014-0000
IOS Press

A new image segmentation framework based on two-dimensional hidden Markov models

Josef Baumgartner [a,*], Ana Georgina Flesia [b], Javier Gimenez [c] and Julian Pucheta [a]

[a] *Laboratorio de Investigación en Matemática Aplicada al Control, Universidad Nacional de Córdoba, Argentina*
[b] *Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Argentina and CIEM-CONICET*
[c] *Instituto de Automática, Universidad Nacional de San Juan, Argentina and CONICET*

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. Our algorithm, called *Complete Enumeration Iteration* (CEP), is flexible in the sense that it allows the use of different probability distributions as emission probabilities. Not only do we compare the performance of different probability functions plugged into our framework but also propose three methods to update the distributions of each state "online" during the segmentation process. We compare our algorithm with a 2D-HMM standard algorithm and Iterated Conditional Modes (ICM) using real world images like a radiography or a satellite image as well as synthetic images. The experimental results are evaluated by the kappa coefficient ($\widehat{\kappa}$). In those cases where the average $\widehat{\kappa}$ coefficient is higher than 0.7 we observe an average relative improvement of 8% of CEP with respect to the benchmark algorithms. For all other segmentation tasks CEP shows no significant improvement. Besides that, we demonstrate how the choice of the emission probability can have great influence on the segmentation results. Surprisingly, we observe that the normal distribution is an appropriate density function for many segmentation tasks.

Keywords: Image Segmentation, Hidden Markov Models, Viterbi Training, Probability Density Function, kappa coefficient

1. Introduction

Image segmentation is a key competence of computer vision and an active research field [15,20,23]. In this work we present a new segmentation algorithm that is based on the theory of hidden Markov models (HMM). Our goal is to provide reliable segmentations of images whenever the human eye cannot make a solid classification. One example are satellite images where it is important to estimate the quantity of certain crops like soybeans in order to predict the market prices. Another example is the detection of pedestrians in far infrared images [24].

In the last years, many techniques from the broad field of computational intelligence have been successfully applied to image segmentation [25,29]. Unfortunately the classical HMM are generally limited to those areas where the observed data has only one dimension, such as protein sequences [2], the analysis of biometric data [22] or supply chains [28]. To overcome this limitation, there were attempts to first reorder two-dimensional data in a one-dimensional way and then run classical HMM. In image segmentation one can think of lining up the rows or the columns of the image before actually applying HMM [14]. The drawback of such an ordering is clearly the loss of information because adjacent pixels in the original image are torn apart.

Therefore, efforts were made to extend the classical one-dimensional HMM to higher dimensions

[*] Corresponding Author: Josef Baumgartner, Laboratorio de Investigación en Matemática Aplicada al Control, UNC, Córdoba, Vélez Sarsfield 1611, X5016GCA Córdoba, Argentina; E-mail: jbaumgartner@efn.uncor.edu.

[19]. The problem hereby is, that the standard method of parameter estimation for one-dimensional HMM, the Baum-Welch algorithm [1], is not feasible for higher dimensions. Hence, the main question for two-dimensional hidden Markov models (2D-HMM) is: How can the computational complexity be reduced in order to make the n-dimensional HMM feasible? Hereby, one of the most common approaches is the so called Path Constrained Viterbi Training (PCVT) [17,18].

The problem of PCVT is that a pre-selection of states is carried out before running a HMM decoder. Thus, it is probable, that optimal states are discarded before applying HMM-algorithms. To overcome this problem we propose a new algorithm that considers all states of all pixels and still is feasible. We call our approach *Complete Enumeration Propagation* (CEP) because the state probabilities are completely enumerated and then propagated through the image.

We integrate the mathematical foundations of CEP in a 2D-HMM framework that allows any probability density function to be used as an emission probability. Besides that, we propose three methods to update the emission probabilities of each state "online" during the segmentation process.

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 the 2D-HMM feasible. Thereafter, we present two feasible approximations of a complete 2D-HMM in Section 3: PCVT and CEP. Next, we evaluate PCVT and CEP for five test images in Section 4. To compare the performance of PCVT and CEP to benchmark algorithms like Iterated Conditional Modes (ICM) and Maximum Likelihood (ML) we use Cohen's $\widehat{\kappa}$ coefficient [3]. Finally, we present the conclusions in Section 5 where we point out some important differences between PCVT and CEP.

2. Theory of two-dimensional Hidden Markov Models

In this section we describe the theoretical background of how to estimate the parameters of a 2D-HMM. The presented algorithm can be derived directly from the Expectation-Maximization algorithm (EM) [4,16].

A HMM is a probabilistic model that is used to analyze and describe correlated noisy data. Given some radiometric image observation O, the main goal is to find the hidden state of each pixel. Therefore, we assume that all states S, like the background, or objects, are observed through a certain probability function.

Besides that, it is necessary to introduce transition probabilities, which indicate the probability of being in state s_t given the previous state s_{t-1}. Note, that transition probabilities are independent of the observations and thus add pure contextual information to our model. Without this information our model would reduce to a simple ML classifier as show in Algorithm 3.

In the case of one-dimensional data, the use of transition probabilities is straightforward. For two-dimensional data – like the pixels of an image – we need a more advanced notation.

First of all let's assume, that the hidden states of the data represent a Markov random field (MRF). 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 under the Markov assumption this definition leads to a 2nd order Markov Mesh which specifies for state $s_{i,j}$:

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

The two pixels $(i,j-1)$ and $(i-1,j)$ can be understood as the "past" of pixel (i,j) as shown in Figure 1. In other words we are moving from the top-left pixel to the bottom-right pixel. Along this way, we assume, that the transition probabilities from states $s_{i,j-1}$ and $s_{i-1,j}$ to state $s_{i,j}$ do not depend on the current pixel. Therefore, we can gather all transition probabilities from Equation (1) in a three-dimensional matrix A which consists of

$$a_{m,n,l} = P(s_{i,j}=l|s_{i,j-1}=m, s_{i-1,j}=n). \tag{2}$$

Now, for a given image with M hidden states, the transition probabilities $a_{m,n,l}$ are given by

$$a_{m,n,l} = \frac{\sum_{i,j} H_{m,n,l}(i,j)}{\sum_{l'=1}^{M}\sum_{i,j} H_{m,n,l'}(i,j)}. \tag{3}$$

In Equation (3), $H_{m,n,l}(i,j)$ is the probability of a transition from states m and n to state l at pixel (i,j). The exact formula for $H_{m,n,l}(i,j)$ – except a normal-

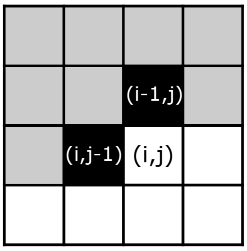


Fig. 1. Transitions among states in a 2nd order Markov Mesh. The gray and the black pixels fulfill $(i^{'},j^{'}) \prec (i,j)$ but the two black pixels are sufficient statistics for pixel (i,j) under the Markov assumption.

ization term – is given by

$$H_{m,n,l}(i,j) \propto \sum_{\mathbf{s}} I\left(m{=}s_{i-1,j}, n{=}s_{i,j-1}, l{=}s_{i,j}\right) \prod_{(i',j')\in\mathbb{N}} P\left(O_{i',j'} \mid \theta\right) a_{s_{i'-1,j'}, s_{i',j'-1}, s_{i',j'}} . \tag{4}$$

In Equation (4), $I(\cdot)$ is the indicator function, $O_{i',j'}$ is the observation of pixel $(i^{'},j^{'})$, $\mathbb{N}$ stands for all the pixels of the image and θ represents the parameters of the emission probabilities of the different states. Without going into detail, note, that $H_{m,n,l}(i,j)$ depends on the transition probabilities $a_{m,n,l}$ from Equation (2) which we were trying to calculate in first place. One solution to this problem is to update $a_{m,n,l}$ and $H_{m,n,l}(i,j)$ iteratively, following the EM framework for latent variables. Still, there are some severe complexity issues involved in the calculation of $H_{m,n,l}(i,j)$. In the next section we discuss this topic with more detail.

To complete our description of a 2D-HMM, we have to define the initial probabilities and talk about the emission probabilities of the hidden states. For our definition of "past states", the initial probabilities of the 2D-HMM depend only on the state of pixel $(0,0)$:

$$\pi_l = P(s_{0,0} = l) \qquad \forall l \in \mathcal{S}. \tag{5}$$

In Equation (5), $s_{0,0}$ refers to the state of the upper left pixel of an image.

When it comes to emission probabilities, we would like to point out, that the mathematical framework of 2D-HMM does not require a particular probability distribution to be used as an emission probability. In the following, we show exemplary how to estimate the parameters of a normal distribution.

Let's suppose we have observed state $l \in \mathcal{S}$ through a normal distribution with mean μ_l and standard deviation σ_l. Thus the emission probability $b_l(x)$ of state l is the normal density function.

$$b_l(x) = P(x|s_{i,j} = l) = \\ = \frac{1}{\sigma_l\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma_l}\right)^2\right\} \tag{6}$$

The parameters μ_l and σ_l of Equation (6) can be estimated by

$$\mu_l = \frac{\sum_{i,j} L_l(i,j) O_{i,j}}{\sum_{i,j} L_l(i,j)} \tag{7}$$

$$\sigma_l = \frac{\sum_{i,j} L_l(i,j)\left(O_{i,j}-\mu_l\right)\left(O_{i,j}-\mu_l\right)^T}{\sum_{i,j} L_l(i,j)} \tag{8}$$

where $L_l(i,j)$ indicates the probability of pixel (i,j) being in state l. $L_l(i,j)$ is given by

$$L_l(i,j) \propto \sum_{\mathbf{s}} I(l = s_{i,j}) \prod_{(i',j')\in\mathbb{N}} P\left(O_{i',j'} | \mu_{s_{i',j'}}, \sigma_{s_{i',j'}}\right) a_{s_{i'-1,j'}, s_{i',j'-1}, s_{i',j'}} \tag{9}$$

Note in Equation (9), that $L_l(i,j)$ depends on $\mu_{s_{i',j'}}$ and $\sigma_{s_{i',j'}}$. Hence, the only way to find the latent variables μ_l, σ_l and $L_l(i,j)$ is to run the iterative EM. And just like in case of $H_{m,n,l}(i,j)$ we are facing computational problems when calculating $L_l(i,j)$, as we will show in the next section.

In summary, the main goal of a 2D-HMM is to find the optimal hidden state map s^* through an iterative approach derived from EM. This objective can be formally expressed by

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{s}|O,\theta). \tag{10}$$

In Equation (10), O are the observations and θ are the parameters of the 2D-HMM, such as the initial probabilities π_l, the transition probabilities $a_{m,n,l}$ and the emission parameters μ_l and σ_l of each state. Because we have strictly followed the EM framework so far we have a theoretical guarantee that the iterative algorithm described in this section converges to a state map. The problem with Equation (10) is, that s stands for all possible state maps which is usually a infeasible high number.

3. Approaches to two-dimensional Hidden Markov Models

In the previous section we pointed out, that the calculation of some parameters of the 2D-HMM is infeasible even for small images. In this section we investigate the complexity problems with more detail and show different ways of how the exact 2D-HMM can be approximated. Note, that by using approximations instead of exact formulas, we leave the EM framework and thus have no more a theoretical guarantee that our 2D-HMM converges. Still, it is our only choice if we want to come up with a feasible model that – at least – approximates the optimal hidden state map as defined by Equation (10).

3.1. Parameter estimation

The exact formulas for the parameters of a 2D-HMM are given by Equations (3) to (9) where we have to sum over all possible state maps s. If you think of an image of size $w \times z$ with M states there are $(wz)^M$ possible state maps. This number is generally so high that the exact algorithm is infeasible – not to mention that for every single state map there have to be carried out complex operations.

In order to reduce the computational burden, we propose an approximation of the transition probabilities $a_{m,n,l}$ and of the emission parameters μ_l and σ_l. Instead of summing over all possible state maps we suggest to estimate all parameters by using only the state map of the current iteration. For $a_{m,n,l}$ in iteration step p we get

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

For the means and standard deviations of each state we use

$$\mu^{(p)}_l = \frac{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s^{(p-1)}_{i,j}=l\right)O_{i,j}}{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s^{(p-1)}_{i,j}=l\right)} \tag{12}$$

$$\sigma^{(p)}_l = \frac{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s^{(p-1)}_{i,j}=l\right)(O_{i,j}-\mu_l)(O_{i,j}-\mu_l)^T}{\sum_{i=0}^{z-1}\sum_{j=0}^{w-1} I\left(s^{(p-1)}_{i,j}=l\right)} \tag{13}$$

One can think of these simplified formulas as “count instead of evaluate”, because we are only taking into account our currently best guess of the hidden state map instead of evaluating all possible state maps. Therefore, the approximations presented in Equations (11), (12) and (13) are computationally extremely simple. All we have to do, is to process the state and the observation of each pixel.

Note, that switching to another emission probability – like a Gamma or a Weibull distribution – is straightforward in this context. All we have to do, is to replace Equations (12) and (13) by the corresponding parameter estimators. For maximum likelihood estimators of many common probability distributions please refer to [26].

3.2. Evaluation of a two-dimensional Hidden Markov Model

After solving the problem of parameter estimation in the previous section, we are left with the problem of decoding our 2D-HMM in order to obtain a hidden state map. First of all, remember our notion of “past” as shown in Figure 1. This definition allows us to think of each bottom-left to upper-right diagonal of the image as one step in time, starting with the top-left pixel. Thus, the diagonals $T_0, T_1, T_2 \dots$ are

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

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

$$\begin{aligned} P(\mathbf{s}) =& P(T_0)P(T_1|T_0)\dots \\ & P(T_{z+w-2}|T_{z+w-3},\dots,T_0) \\ =& P(T_0)P(T_1|T_0)\dots P(T_{z+w-2}|T_{z+w-3}). \end{aligned} \tag{14}$$

In Equation (14), each diagonal operates as an "isolating element" between neighboring diagonals. Hence, we have transformed the complex two-dimensional model into a pseudo one-dimensional HMM.

At this point we could think of running a one-dimensional HMM decoder like the Baum-Welch algorithm [1], but once again we are facing computational issues. Remember, that in a $w \times z$ image the diagonals consist of up to $min(w,z)$ pixels. If there are M states, a diagonal can be in one of $min(w,z)^M$ superstates. So we are dealing with a one-dimensional HMM with up to $min(w,z)^M$ superstates at each time step. Such a HMM is clearly not feasible.

Before we present two feasible 2D-HMM decoders we have to make the assumption that the emission probability of pixel (i,j) depends only on the current state and not on neighboring states. Even though this assumption is reasonable and not very restrictive, both approaches require it.

3.2.1. *Path Constrained Viterbi Training*

In the last years, several approximations were proposed to make the 2D-HMM feasible. One of the most promising approaches is to cut down the superstates of each diagonal to a certain number N and then run the Viterbi algorithm [12]. This algorithm is named *Path Constrained Viterbi Training* (PCVT) because the number of superstates – and hence the number of paths through the image – is constrained.

The selection of the N superstates is done by ML, which means, that almost all superstates are discarded without taking into account contextual information. Thus, there is no guarantee, that the optimal hidden state map is a subset of the chosen superstates. At least the Viterbi algorithm considers all dependencies among neighboring pixels when evaluating the N independently chosen superstates.

A sketch of the PCVT as presented by [12] is shown in Algorithm 1. Note, that the computation of step 5) has order $\mathcal{O}((w+z-1)N^2)$ for an image of size $w \times z$, whereas the other steps have negligible computational complexity.

Algorithm 1. Path-Constrained Viterbi Training (PCVT)

```
1. Initialize parameters $\mu_l$ and $\sigma_l$ for $l \in \mathcal{S}$.
2. Initialize state map using ML.
3. Calculate transition probabilities $a_{n,m,l}$ for every
   $n, m, l \in \mathcal{S}$ using Equation (11).
4. Choose the $N$ best superstates for each diagonal using ML.
5. Run Viterbi algorithm.
6. Update parameters $a_{n,m,l}$, $\mu_l$ and $\sigma_l$ using Equations (12), (13) and (11).
7. Iterate steps 4), 5) and 6) until convergence.
```

3.2.2. *Complete Enumeration Propagation*

In the following we present a new 2D-HMM decoding algorithm, that, in contrast to the PCVT, does not discard any possible hidden state and still is feasible. Instead of grouping the states of a diagonal in a superstate, we use complete enumeration to calculate the state probabilities of each pixel. Then we propagate the state probabilities through the image until we reach the bottom-right pixel. Therefore, we call our approach *Complete Enumeration Propagation* (CEP).

Let's start with the state probability of pixel (i,j) which is given by

$$\begin{aligned} & P(s_{i,j}|s_{i,j-1},s_{i-1,j},O_{i,j}) \propto \\ & P(s_{i,j},s_{i,j-1},s_{i-1,j},O_{i,j}) = \\ & = P(s_{i,j-1},s_{i-1,j})P(s_{i,j}|s_{i,j-1},s_{i-1,j})P(O_{i,j}|s_{i,j}). \end{aligned} \tag{15}$$

If we now replace $P(s_{i,j}|s_{i,j-1},s_{i-1,j})$ in Equation (15) by the transition probability $a_{s_{i,j-1},s_{i-1,j},s_{i,j}}$ and consider two diagonal pixels to be independent we can write

$$\begin{aligned} & P(s_{i,j}|s_{i,j-1},s_{i-1,j},O_{i,j}) \propto \\ & P(s_{i,j-1})P(s_{i-1,j})a_{s_{i,j-1},s_{i-1,j},s_{i,j}}P(O_{i,j}|s_{i,j}). \end{aligned} \tag{16}$$

This is the main formula to calculate the state probabilities of pixel (i,j) given the observation and the two past states. Note, that the most restrictive assumption here is to suppose independence of the two past pixels. Nevertheless, PCVT has to make the same assumption when searching for the N superstates of each diagonal.

Now the main idea of CEP is to use Equation (16) to calculate $P(s_{i,j} = l)$ for $l = 1,2,\dots,M$ for all

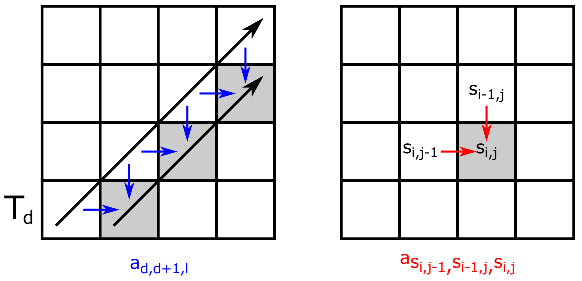


Fig. 2. While PCVT uses $a_{d,d+1,l}$ which represents the transition probability from one diagonal to another, CEP applies the transition probability $a_{s_{i,j-1},s_{i-1,j},s_{i,j}}$ for each pixel.

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 (17).

$$P(s_{i,j} = l|O_{i,j}) \propto \sum_{m=1}^{M}\sum_{n=1}^{M} a_{s_{i,j-1}=m,s_{i-1,j}=n,s_{i,j}=l} \\ P(s_{i,j-1} = m)P(s_{i-1,j} = n)P(O_{i,j}|s_{i,j} = l) \tag{17}$$

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

$$\sum_{l=1}^{M} P(s_{i,j} = l) = 1. \tag{18}$$

See Algorithm 2 for a schematic description of CEP and Figure 2 for a comparison of the state transitions applied by PCVT and CEP.

Algorithm 2. Complete Enumeration Propagation (CEP)

```
1. Initialize parameters $\mu_l$ and $\sigma_l$ for $l \in \mathcal{S}$.
2. Initialize state map using ML.
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 (16), (17) and
   (18).
5. Update parameters $a_{n,m,l}$, $\mu_l$ and $\sigma_l$ using Equations (12), (13) and (11).
6. Iterate steps 3), 4) and 5) until convergence.
```

When analyzing the computational complexity of CEP, only the calculation of step 3) is worth mentioning. This step is of order $\mathcal{O}((w * z)M^3)$ for an image of size $w \times z$ 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 (16). 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 the optimal hidden state map $\mathbf{s}^*$ as shown in Equation (10). From this point on we use the formulas (11), (12) and (13) to update the parameters of the emission probabilities. Then we iterate this procedure until convergence.

3.3. *Emission probabilities*

As we have shown in Section 3.1, our 2D-HMM framework allows different probability distributions as emission probabilities. On the one hand we study the performance of CEP using different probability functions. Therefore, we use the following five distributions: *Gamma*, *inverse Gaussian* (also known as *Wald distribution*), *Nakagami*, *Normal* and *Weibull* as defined by [26]. On the other hand we propose to select the appropriate distribution of each class online at the beginning of each iteration step.

Let's suppose CEP has just finished iteration i. At this point we have a hidden state map and the observations. So instead of adjusting the parameters of a given probability function to the corresponding observations, one can ask the question: Which distribution explains best the observations of state l? Once we have answered this question, we can switch to the chosen distribution, adjust its parameters and keep on running CEP.

In this section we present three ways to select a emission distribution for a given set of observations. In all three methods, the first step is to adjust the parameters of all candidate distributions to the corresponding observations using a maximum likelihood estimator. As possible distributions we use the ones named at the beginning of this section.

3.3.1. Bayes Theorem

The first method is based on Bayes Theorem. Suppose we have a set of distribution families $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$ where each family has the form $\mathcal{F}_i = \{f_{i,\theta} : \theta \in \Theta_i\}$ with Θ_i being the parameter space of $\mathcal{F}_i$. For example one $\mathcal{F}_i$ could be the family of Gaussian distributions and Θ_i is the set of possible parameters.

The approach based on Bayes Theorem assigns to each state the probability density

$$f_l = \underset{1\leq i\leq n, \theta\in\Theta_i}{\arg\max} \prod_{s_{i,j}=l} f_{i,\theta}(O_{i,j}) \tag{19}$$

where $f_{i,\theta}(O_{i,j})$ is the probability function of observing $O_{i,j}$ given that the emission density corresponds to family $\mathcal{F}_i$ with parameter $\theta \in \Theta_i$.

3.3.2. Kolmogorov-Smirnov Test

The key idea behind the second method is to use the p-value of the two-sample Kolmogorov-Smirnov test (KS) [21] in order to determine which is most suitable distribution for state l. Therefore we draw samples from the adjusted distributions and compare them with the observations. After calculating the p-value for every possible distribution we choose the one with the highest p-value.

3.3.3. Kullback-Leibler Divergence

Finally we propose a method that is based on the Kullback-Leibler Divergence (KL). This time we look at the histogram of the observations and evaluate the fitted probability function at each point of the histogram. Then we calculate the Kullback-Leibler Divergence of the probability densities and the histogram. The smaller the Kullback-Leibler Divergence, the more likely that the observations come from the tested distribution.

In the next section we present the experimental results for each emission probability as well as for each distribution-adaptation-strategy.

4. Experimental results: Image segmentation

In this section we present the experimental results of different image segmentation algorithms. To evaluate the algorithms, we use Cohen's $\widehat{\kappa}$ coefficient [3] which is a widely accepted statistical measure of agreement [11,27]. It is defined as

Algorithm 3. Maximum Likelihood Classification (ML)

```
1. Initialize parameters $\mu_l$ and $\sigma_l$ 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 $s_{i,j} =$
   $\arg\max_{l\in\mathcal{S}} P(I_{i,j}|s_{i,j} = l, \theta)$.
```

$$\widehat{\kappa} = \frac{P_O - P_E}{1 - P_E}. \tag{20}$$

In Equation (20), P_O is the relative observed agreement between the segmented image and the ground truth and P_E is the hypothetical probability of chance agreement.

As mentioned in Section 3, there is no theoretical guarantee that our 2D-HMM converges. Still, we observed that in more than 97% of the experiments PCVT and CEP converged within 150 iterations. Only very few times the 2D-HMM got stuck in an endless loop where two to five pixels were constantly changing state forward and backward. The reason for this behavior is that once a pixel changes its state the transition probabilities and the emission probabilities slightly change. So in the next iteration the pixel might switch back to its previous state and so on.

To stop PCVT and CEP in the case of some endlessly switching pixels, we set a maximum of 150 iterations. Nevertheless, the state-switching pixels have almost no effect on the $\widehat{\kappa}$ coefficient.

In Figure 3, we compare how many iterations it takes for PCVT and CEP to converge. Therefore we gather the number of iterations of all experiments in two histograms – one for PCVT and one for CEP. Finally, we place smooth curves over these histograms in order to obtain a figure that is easy to interpret.

We observe that in general PCVT needs less iterations than CEP. On the other hand the computational complexity of a single iteration is higher in the case of PCVT. As a result, the computing times of PCVT and CEP are very similar as shown in Figure 6.

Another issue of iterative, statistical models is the influence of the initial parameters on the final result. Of course it would be advantageous to have a theoretical guarantee for the invariance of the final result, but, as we pointed out in Section 2, Equation (10) is infeasible. Therefore we cannot expect to obtain the same segmentation for different initializations.

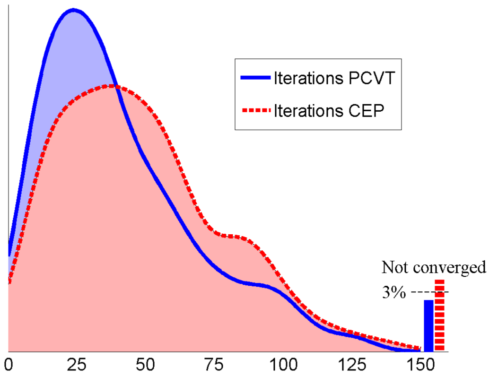


Fig. 3. Probability density estimation of iterations necessary until convergence. In some cases (around 3% of all experiments) PCVT and CEP did not converge.

To evaluate the influence of the initial parameters, we started the experiments in this section from different points and observed the resulting $\widehat{\kappa}$ values. In Figures 5, 6 and 8 we show the mean $\widehat{\kappa}$ coefficients as well as the standard deviations for different initial parameters. In most cases, the standard deviations of $\widehat{\kappa}$ are below 0.05. This indicates that the initial parameters have very little influence on the final segmentation.

Before we present the experimental results, we like to point out, that on the one hand we want to evaluate the performance of different emission probabilities plugged into our 2D-HMM framework while on the other hand we like to compare our approach to other classification algorithms. As benchmark classifiers we use PCVT as described in Algorithm 1, Maximum Likelihood Classification (ML) as described in Algorithm 3 and Potts Iterated Conditional Modes (ICM) as described in Algorithm 4.

While ML is a classical non-contextual classification method, ICM is an iterative algorithm that rapidly converges to the local maximum of the function $P(\mathrm{s}|O,\theta)$ closest to the initial segmentation provided by the user. ICM is a well studied contextual algorithm that was used as a benchmark function in many works [7,10]. The algorithm goes back to a work of Geman and Geman [8], where they 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.

In this work, the initial segmentation for ICM is provided by ML and the parameter β is estimated according to [9]. Note, that ICM is an isotropic method whereas PCVT and CEP are causal 2D-HMM [13].

Algorithm 4. Iterated Conditional Modes (ICM)

```
1. Choose a pixel's visit scheme for the image.
2. Initialize parameters $\mu_l$ and $\sigma_l$ for $l \in \mathcal{S}$.
3. ML segmentation of $O$.
4. Estimate parameter $\beta$.
5. For each pixel $(i,j)$, change the label given in the
   previous iteration for the label $l \in \mathcal{S}$ that maximizes
   $$g(l) = \ln p(O_{i,j}|l,\mu_l,\sigma_l) + \beta U_{i,j}(l) \qquad (21)$$
   where $U_{i,j}(l)$ is the number of pixels in the neighborhood of $(i,j)$ with hidden state $l$.
6. Iterate step 4) and 5) until convergence.
```

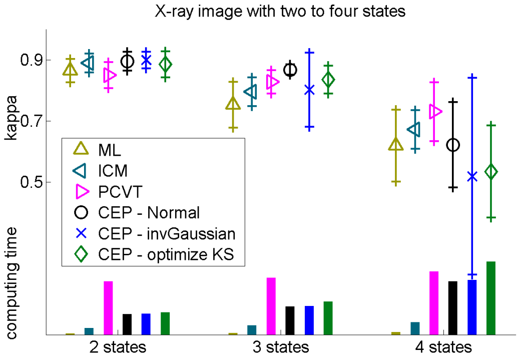


Fig. 5. Evaluation of the X-ray image for different setups. The four states of the image can be merged to two or three superstates. All algorithms were started from different initial conditions. The mean $\widehat{\kappa}$ values and its standard deviations are shown, as well as the average computing times. Note, that the computational complexity of CEP rises with the number of states.

We propose five different scenarios to evaluate and compare ML, ICM, PCVT and CEP with its different emission probabilities. The experiments consist of three real images – an inverse digitized X-ray image, a multispectral optical Landsat image and a standard benchmark image – as well as an artificial image and a database of 200 synthetic images that were observed through different probability distributions.

4.1. Multimodal X-ray image

The first image comes from the field of diagnostic radiography. Due to the sensing method and the posterior digitization process, this type of imagery has a

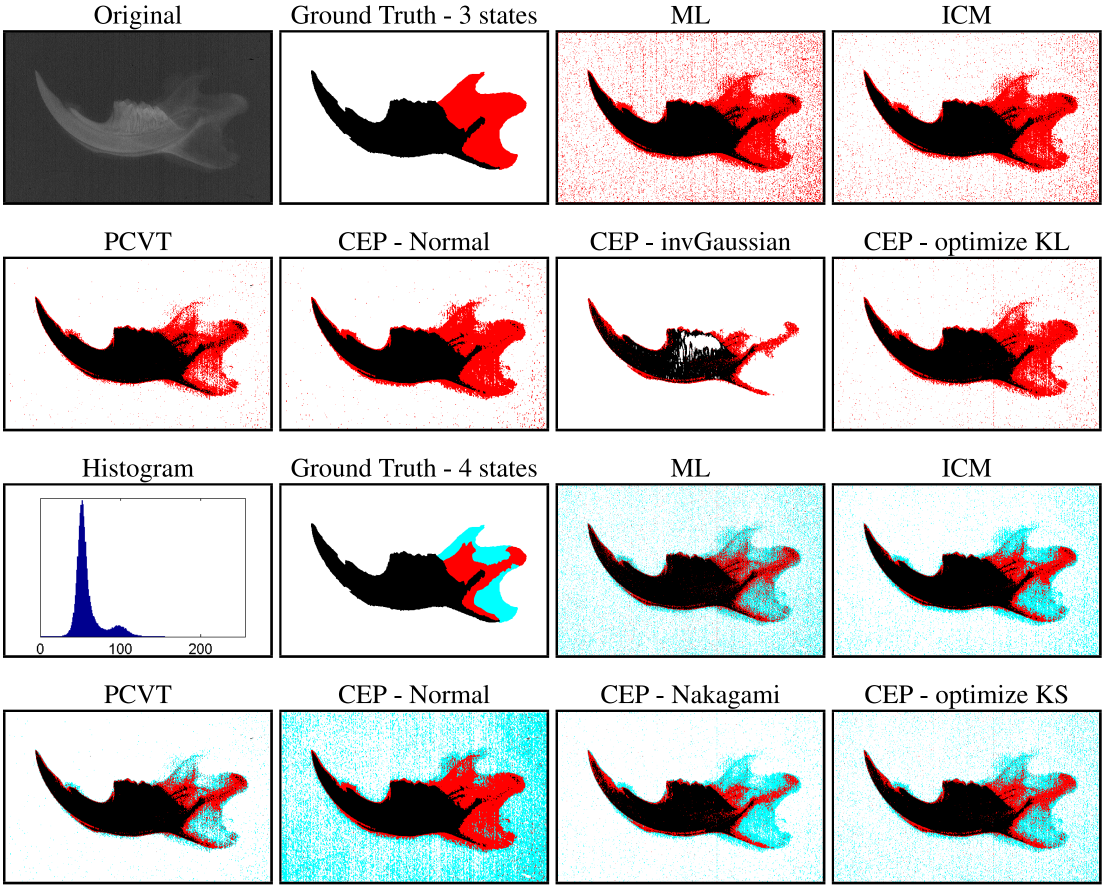


Fig. 4. Segmentations of an X-ray image with three and four states. The emission probability has great influence on the final segmentation. Interestingly, the segmentation of the rat tooth is almost the same for three and four states when using CEP with a normal distribution. In this case, CEP uses the extra state to model the background.

very low signal-to-noise ratio [5]. It shows a Wistar rat's jaw and forms part of a growth study of rats [6]. As ground truth we use a manual segmentation made by a biologist.

The interesting aspect about this image is that it contains four states (bone and tooth, tissue and flesh, cartilage and background) which can also be grouped into two or three superstates, in accordance to the smooth modes of the intensity histogram.

We run ML, ICM, PCVT and CEP for all three cases (four states, two or three superstates) and evaluate the performance using $\widehat{\kappa}$. Some of the segmentation results of this scenario are shown in Figure 4. The evaluation of this experiment with respect to different emission probabilities is shown in Figure 5 and in Figure 6 we present a comparison of all segmentation algorithms.

4.2. *Multispectral, multimodal satellite image*

The second experiment is a multispectral Landsat TM image of an agricultural area in the humid pampa of Argentina. It 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 7, since we only have ground truth labels for these regions. The ground truth data consists of three states, but we have no information of how many states are present in whole the image, so we evaluate all algorithms for three to five states and show the results in Figure 8.

4.3. *Synthetic imagery*

In Section 3.3, we proposed three algorithms to find the optimal emission probability "online" after each

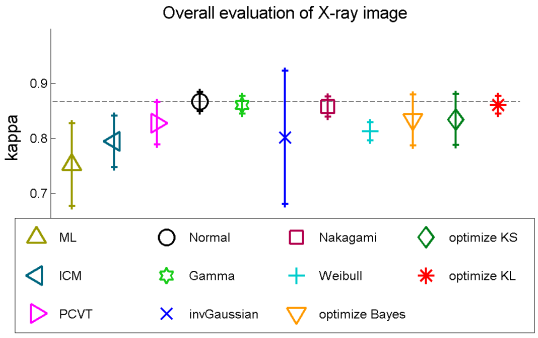


Fig. 6. Overall evaluation of the X-ray image. CEP outperforms the other algorithms for all emission probabilities – except the inverse Gaussian and the Weibull distribution.

Algorithm 5. Synthetic images
1. Design ground truth.
2. Choose order of states randomly (e.g. the background could be state two, etc.).
3. Choose the emission probability for each state randomly.
4. Choose parameters of each probability function randomly within certain limits.
5. Observe the ground truth according to the given distributions.
6. Repeat steps 2), 3) and 4) 200 times.

iteration step. In this section, we evaluate the performance of these algorithms for a database of 200 synthetic images. The idea is to find out, whether the presented algorithms can correctly identify the underlying emission probabilities or not. Besides that, we want to verify that our distribution-updating model achieves better results in terms of $\widehat{\kappa}$ than ML or a common CEP-model that uses only normal distributions as emission probabilities.

First of all, we designed a synthetic image with four states, as shown in Figure 10. This image serves as the ground truth for our experiment. Starting from that ground truth, a probability distribution is chosen at random for every state and its parameters are varied within certain limits. The applied probability functions are: *Gamma*, *inverse Gaussian*, *Nakagami* and *Weibull*. Once we have picked four distributions for the four ground truth states, the observations are made according to the density functions. The exact procedure to obtain the synthetic images is described in Algorithm 5.

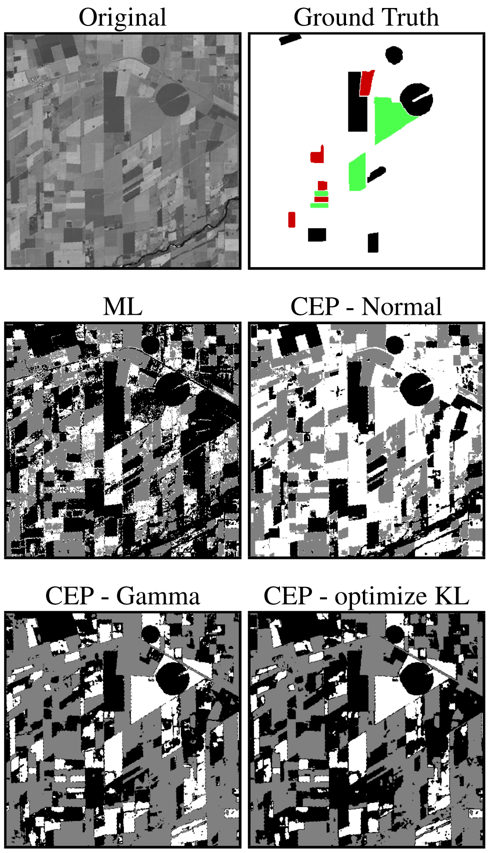


Fig. 7. Segmentations of band five of a Landsat image.

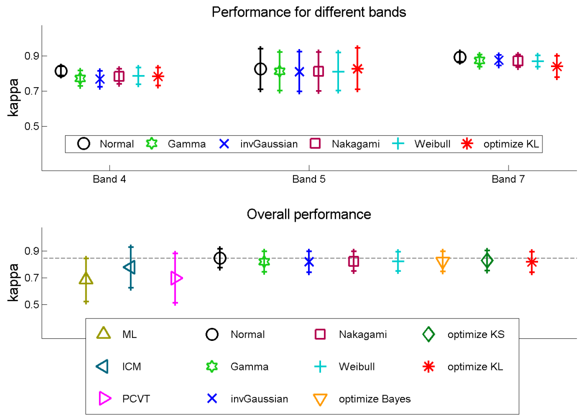


Fig. 8. Evaluation of the Landsat image. Above: Comparison of different emission probabilities in the CEP-framework. Below: Overall performance comparing CEP with the benchmark classifiers. Note, that independently of the emission probability CEP achieves the highest $\widehat{\kappa}$ values.

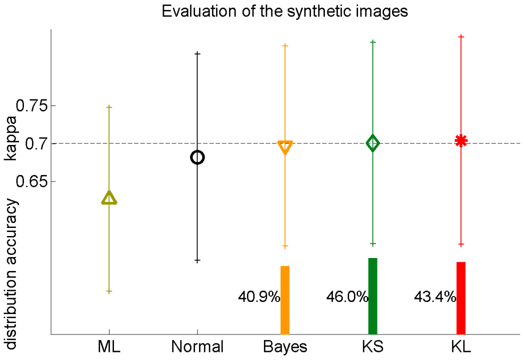


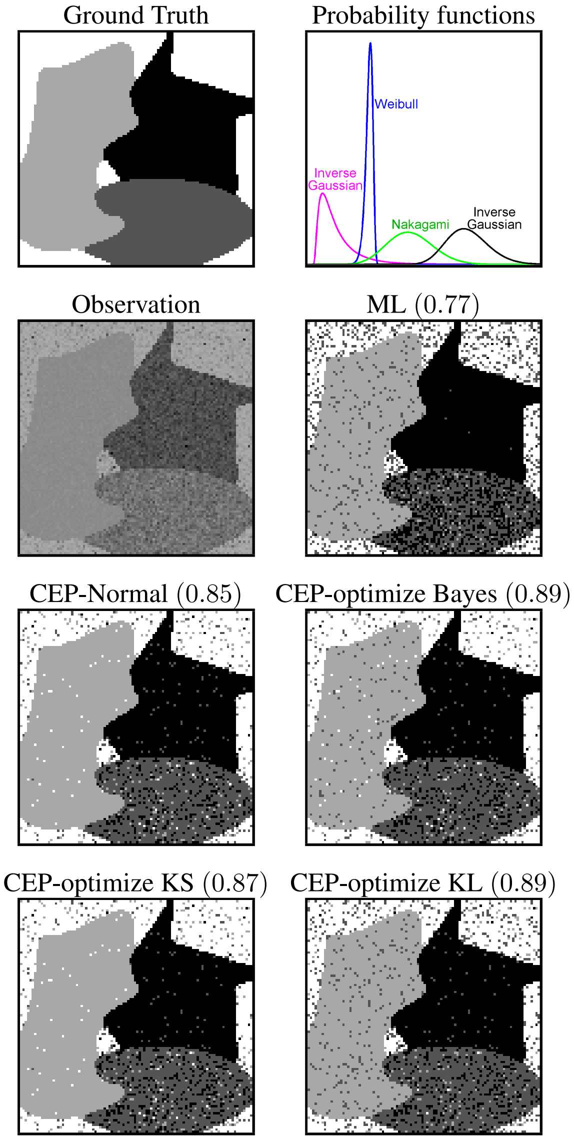
Fig. 9. Evaluation of the segmentation results of the synthetic images. Above: Average $\widehat{\kappa}$ for each algorithm. Below: Percentage of correctly identified emission probabilities.

The performance of the three online algorithms was very similar for the 200 test images. All methods correctly identified between 40% and 46% of the emission probabilities and achieved average $\widehat{\kappa}$ values between 0.697 and 0.704. Still, the online methods have a relative improvement of 3% with respect to the normal distribution and a 12% improvement compared to ML. The average $\widehat{\kappa}$ values of this experiment are presented in Figure 9 and in Figure 10 some of the segmentations are shown.

4.4. Binary image

In this experiment we try to segment the logo of the Technological National University (UTN). The ground truth and some segmentations of this two-class problem are shown in Figure 11. In this experiment we observe the two states "logo" and "background" through normal distributions with varying means. As a result, we get images that are hard to segment for close means and easy to segment when the means of the two states are far from each other.

We observe, that the classical CEP with normal distributions has problems separating the two states when their means are close to each other but shows excellent results for more separated states. The same holds for CEP when it is used together with the distribution-updating algorithms. Like in the case of the synthetic image, KL shows slightly better results than Bayes and KS. The evaluation of this experiment is presented in Figure 12.



Ground Truth

Probability functions

Observation

ML (0.77)

CEP-Normal (0.85)

CEP-optimize Bayes (0.89)

CEP-optimize KS (0.87)

CEP-optimize KL (0.89)

Fig. 10. Synthetic image: ground truth, one set of probability functions, the corresponding observation and the segmentations with $\widehat{\kappa}$ values in brackets.

4.5. 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 decolorization in the sky behind the cameraman. We run the segmentation algorithms with five states and show the results in Figure 13. Note, that CEP is the only method that finds the building in the background, but its sen-

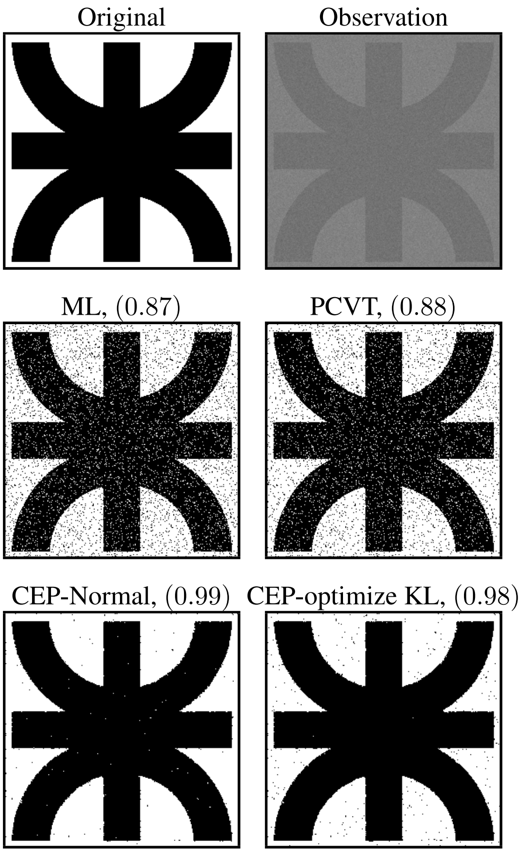


Fig. 11. Segmentation of the logo of the National University of Córdoba. In this case the two states are observed according to $Logo \sim N(50, 25)$ and $Background \sim N(65, 25)$. $\widehat{\kappa}$ values in brackets.

sitivity also makes it highlight shadows in the sky behind the cameraman. 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 run all algorithms 50 times with random initial conditions and present the average number of pixels that changed state from one iteration to another in Figure 14. Besides that we show the intensity histogram of the original picture and the means and standard deviations found by the segmentation algorithms in Figure 15. One can see that ICM does not end up far away from the initial states 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.

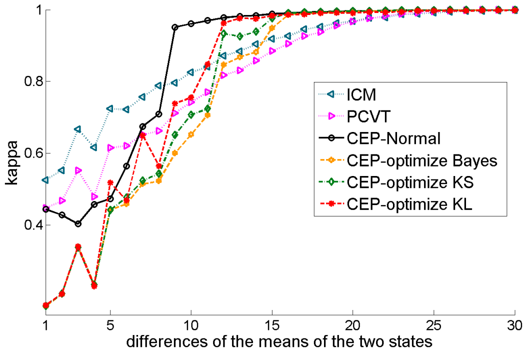


Fig. 12. Segmentations of a binary logo image observed for different normal distributions. The standard deviation of the two distributions is fixed at 5 while the difference of the two means varies between 1 and 30 (x-axis). Hence, the closer the Gaussians, the harder the segmentation task.

5. Conclusions

In this work, we presented a new image segmentation framework. Unlike most of the 2D-HMM algorithms, our approach leads to a feasible 2D-HMM without discarding any hidden states. Therefore we have to assume that two diagonal pixels are independent.

The experimental results show that CEP outperforms the benchmark algorithms when the segmentation task is well defined. As a rule of thumb, CEP should be the first choice when the $\widehat{\kappa}$ coefficient is approximately 0.7 or higher. For this class of segmentation problems, CEP achieved an average relative improvement of 8% with respect to ICM and PCVT.

On the other hand, ICM and PCVT led to better results for images with very low signal-to-noise ratios. Besides that, ICM and the non-contextual ML algorithm demand significantly less computational resources than PCVT and CEP. Note in this context, that the complexity of CEP depends above all on the number of states and that for more than four states CEP is generally slower than PCVT.

When it comes to the emission probability, we observed that the applied probability function can have great influence on the final segmentation. Surprisingly, the normal distribution is appropriate for a wide range of segmentation problems – although it is often crit-

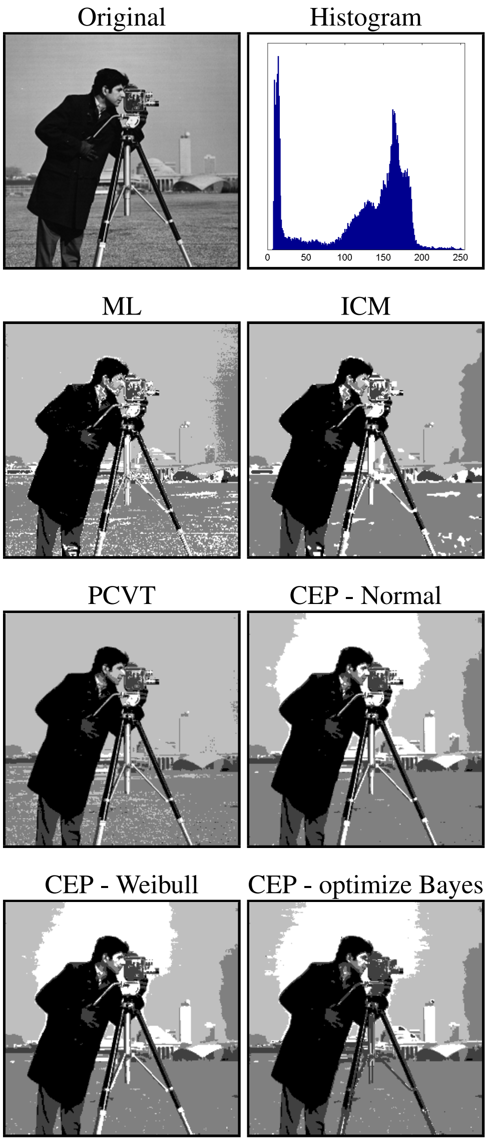


Fig. 13. Segmentations of a standard test image “Cameraman”.

icized for being too narrow. In addition to that, we sometimes obtained very unexpected results for certain probability functions. An example therefore is the segmentation of the X-ray image with an inverse Gaussian distribution.

In order to tackle the problem of which emission probability to use, we proposed three methods to update the emission probabilities online after each iteration. The methods are based on Bayes theorem, the Kolmogorov-Smirnov test and the Kullback-Leibler Divergence. All three methods showed very similar results despite that they are very different approaches. Even though the results of the three methods are not bad, further research has to be done in this area to improve the presented methods.

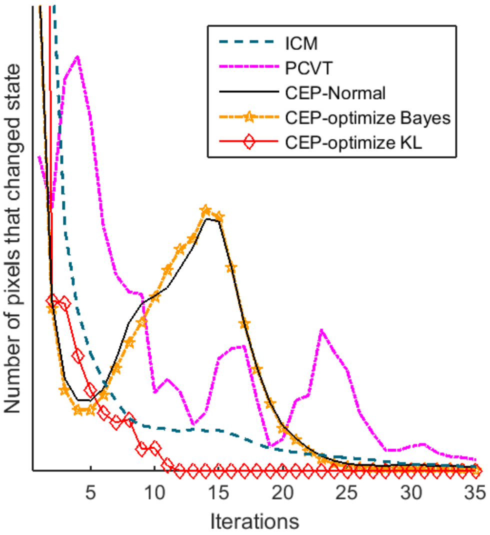


Fig. 14. “Cameraman” test image: Average convergence of ICM, PCVT and CEP for 50 runs with random initial conditions.

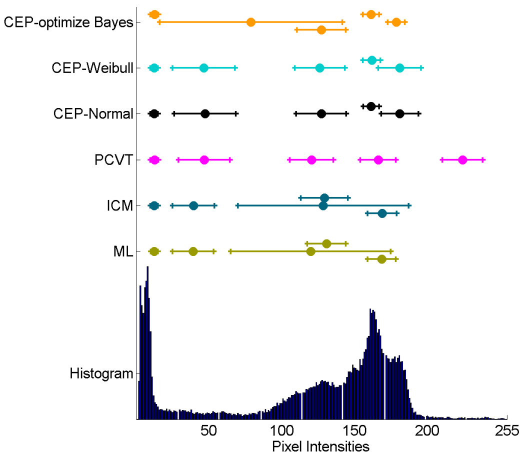


Fig. 15. Histogram of the “Cameraman” test image and the means and standard deviations of every segmentation method.

In conclusion, the CEP is not only a challenge to other HMM algorithms like ICM or PCVT but also a complementary. Especially for images with not too low signal-to-noise ratios CEP should be preferred to the benchmark algorithms. In future works, the 2D-

CEP-framework could be extended to higher dimensions.

Acknowledgment

This work has been partially supported by grants from Secyt-UNC and CONICET. The authors would like to thank the National University of Córdoba.

References

[1] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. In *Annals of Mathematical Statistics 1* (1970), pp. 164–171.

[2] Chen, W.-B., and Zhang, C. A hybrid framework for protein sequence clustering and classification using signature motif information. *Integrated Computer-Aided Engineering 16*, 4 (2009), 353–365.

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

[4] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39*, 1 (1977), 1–38.

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

[6] Fontanetti, P. A., Mandalunis, P. M., and Vermouth, N. T. Response of bone tissue associated with eruption of rat first mandibular molar of pups born from mothers subjected to constant light during pregnancy. *Bone 49*, 6 (2011), 1380–1385.

[7] Fu, X.-Y., You, H.-J., and Fu, K. Building segmentation from high-resolution sar images based on improved markov random field. *Acta Electronica Sinica 40*, 6 (2012), 1141–1147.

[8] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 6 (1984), 721–741.

[9] Gimenez, J., Frery, A. C., and Flesia, A. G. Inference strategies for the smoothness parameter in the potts model. *Proc. of the IEEE International Geoscience and Remote Sensing Symposium 10*, 1 (2013), 1–17.

[10] Jing, J., Li, Y., Li, P., and Jiao, Y. Textile printing pattern image segmentation based on algorithm of mrf. *Journal of Information and Computational Science 10*, 13 (2013), 4007–4015.

[11] Johnson, B., and Xie, Z. Classifying a high resolution image of an urban area using super-object information. *ISPRS Journal of Photogrammetry and Remote Sensing 83* (2013), 40–49.

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

[13] Kato, Z., and Zerubia, J. *Markov Random Fields in Image Segmentation*. Now Publishers Inc, 2012.

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

[15] Lattari, L., Montenegro, A., Conci, A., Clua, E., Mota, V., Vieira, M. B., and Lizarraga, G. Using graph cuts in gpus for color based human skin segmentation. *Integrated Computer-Aided Engineering 18*, 1 (2011), 41–59.

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

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

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

[19] Ma, X., Schonfeld, D., and Khokhar, A. Video event classification and image segmentation based on noncausal multidimensional hidden markov models. *Pattern Recognition Letters 18*, 6 (2010), 1304–1313.

[20] Machado, D. A., Giraldi, G., and Novotny, A. A. Multi-object segmentation approach based on topological derivative and level-set method. *Integrated Computer-Aided Engineering 18*, 4 (2011), 301–311.

[21] Marsaglia, G., Tsang, W., and Wang, J. Evaluating kolmogorov's distribution. *Journal of Statistical Software 8*, 18 (2003), 1–4.

[22] Meraoumia, A., Chitroub, S., and Bouridane, A. 2d and 3d palmprint information, pca and hmm for an improved person recognition performance. *Integrated Computer-Aided Engineering 20*, 3 (2013), 303–319.

[23] Mesquita, R. G., Mello, C. A. B., and Almeida, L. H. E. V. A new thresholding algorithm for document images based on the perception of objects by distance. *Integrated Computer-Aided Engineering 21*, 2 (2014), 133–146.

[24] Olmeda, D., Premebida, C., Nunes, U., Armingol, J. M., and de la Escalera, A. Pedestrian detection in far infrared images. *Integrated Computer-Aided Engineering 20*, 4 (2013), 347–360.

[25] Pedrino, E. C., Roda, V. O., Kato, E. R. R., Saito, J. H., Tronco, M. L., Tsunaki, R. H., Morandin, O., and Nicoletti, M. C. A genetic programming based system for the automatic construction of image filters. *Integrated Computer-Aided Engineering 20*, 3 (2013), 275–287.

[26] Rice, J. *Mathematical Statistics and Data Analysis*. Brooks Cole, 2007.

[27] Sun, L., Wu, Z., Liu, J., Xiao, L., and Wei, Z. Supervised spectral-spatial hyperspectral image classification with weighted markov random fields. *IEEE Transactions on Geoscience and Remote Sensing 53*, 3 (2014), 1490–1503.

[28] Tan, L., and Xu, S. A model checking based approach to risk analysis in supply chain consolidations. *Integrated Computer-Aided Engineering 16*, 3 (2009), 243–257.

[29] Yang, Y. B., Li, Y. N., Gao, Y., Yin, H. J., and Tang, Y. Structurally enhanced incremental neural learning for image classification with subgraph extraction. *International Journal of Neural Systems 24*, 7 (2014). 1450024 (13 pages).