Unsupervised segmentation by hidden Markov chain with bi-dimensional observed process

Abdelali Joumad and Abdelaziz Nasroallah

Abstract—In an unsupervised segmentation context, we propose a bi-dimensional hidden Markov chain model \((X, Y)\) that we adapt to the image segmentation problem. The bi-dimensional observed process \(Y = (Y^1, Y^2)\) is such that \(Y^1\) represents the noisy image and \(Y^2\) represents a noisy supplementary information on the image, for example a noisy proportion of pixels of the same type in a neighborhood of the current pixel. The proposed model can be seen as a competitive alternative to the Hilbert-Peano scan. We propose a bayesian algorithm to estimate parameters of the considered model. The performance of this algorithm is globally favorable, compared to the bi-dimensional EM algorithm through numerical and visual data.

Keywords—Image segmentation, Hidden Markov chain with a bi-dimensional observed process, Peano-Hilbert scan, Bayesian approach, MCMC methods, Bi-dimensional EM algorithm.

I. INTRODUCTION

HIDDEN Markov models (HMM) are frequently used to study one unobserved variable and uni-dimensional observed process such as hidden Markov fields [13], [22], [24], [23], [31], hidden Markov chain [7], [13], [5], [22], [30], [12], [36], pairwise hidden Markov model [32] or more unobserved variable and uni-dimensional observed process such as triple hidden Markov model [11]. The multivariate observed process in HMM is used to study unsupervised multi-component images segmentation. Such vectorial image can be obtained for example, from different channels, from several sensors or from image taken at various moments. There are several interesting publications, like Spherically Invariant Random Vector models (SIRV) and copula [15], [21], [24], [33], [34], that treat the multi-dimensional case [22], [24], [23], [31]. One of the main problems in statistical image segmentation using HMM is the adequacy between temporal and spatial context of pixels during the image modeling. This difficulty is usually solved by Markov fields, or by Hilbert-Peano scan in Markov chain case [7], [23]. We propose a hidden Markov chain with a bi-dimensional observed process as another alternative to study this problem. The cited works above concerned with multi-component images use an observed process \(Y = (Y^1, . . . , Y^M)\) where Hilbert-Peano scan is used to study each image \(Y^i\). Our proposed model consider one noisy image coupled with a noisy supplementary information on the same image (i.e. \(Y^1\) represents the noisy image and \(Y^i\) for \(i = 2, . . . , M\) represents a noisy supplementary information on \(Y^i\)). For simplicity we suppose that \(M = 2\) and \(Y^2\) is a noisy proportion of pixels of the same type in a neighborhood of current pixel (i.e. \(Y^2\) constructed after calculation of proportion on image and noisy them). The proposed model avoids the use of Hilbert-Peano scan. Also, we propose a bayesian algorithm to estimate parameters of bi-dimensional hidden Markov chain (BHMC) model. We denote this algorithm by (BBHMC). The performance of BBHMC is compared to a bi-dimensional EM algorithm (BEM) through a sample of simulation and visual data.

The paper is organized as follows:

In section II, the proposed hidden Markov model BHMC is briefly presented with some necessary hypothesis. Section III is devoted to the BEM algorithm. Section IV is concerned with the parameter estimation problem and the main steps of BBHMC algorithm. In section V, we give an interesting sample of simulation examples that show performance of BBHMC and BEM algorithms: parameter estimations, error rates and CPU times are computed for different scenarios. The efficiency of these two algorithms is examined on a sample of synthetic images in section VI. In the last section we propose a conclusion. A list of references is given in the end.

II. BI-DIMENSIONAL HIDDEN MARKOV MODEL

Consider a random process \((X, Y) = \{(X_t, Y_t) \in \Omega \times \mathbb{R}^2, t \in IN\}\) defined on a probability space \((\Omega, \mathcal{F}, P)\), where \(\Omega = \{\omega_1, . . . , \omega_K\}\) is the state space of Markov process \(X\) and \(K \geq 2\) is a number of classes. The process \(X\) is completely characterized by transition parameters \(c_{ij} = P(X_{t+1} = \omega_i | X_t = \omega_j)\) for \(i = 1, . . . , K\); the initial law \(\nu = (\nu_i, 1 \leq i \leq K)\) and the transition matrix \(A = (a_{ij})_{1 \leq i, j \leq K}\) are such that

\[
\nu_i = P(X_t = \omega_i) = \sum_{1 \leq j \leq K} c_{ij} \quad \text{and} \quad a_{ij} = \frac{c_{ij}}{\sum_{1 \leq j \leq K} c_{ij}},
\]

where \(t_1\) indicates the initial time of Markov chain \(X\) that is supposed homogenous. Furthermore we suppose the following hypothesis:

\((H_1): \forall (t, t') \in \mathbb{N}^2 \text{ such that } t \neq t', \text{ the conditional processes } Y_{t'}|X \text{ and } Y_t|X \text{ are independent.}\)

\((H_2): \forall t \in \mathbb{N}, Y_t|X_t = \omega_i \sim \mathcal{N}(\mu_i, \Sigma_i)\) for \(i = 1, . . . , K\), where the density function is given by

\[
f_i(y_t) = \frac{1}{2\pi|\Sigma_i|^{\frac{1}{2}}} \exp \left(-\frac{1}{2}(y_t - \mu_i)^T \Sigma_i^{-1} (y_t - \mu_i) \right),
\]
with $\mu_i = (\mu_{i1}, \mu_{i2})'$ and $\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & \sigma_{i1}\sigma_{i2} \\ \sigma_{i1}\sigma_{i2} & \sigma_{i2}^2 \end{pmatrix}$, $i = 1, \ldots, K$, are respectively the mean vector and variance-covariance matrix of bi-normal distribution, and $\psi$ is the transpose operator. The process $(X, Y)$ is called bi-dimensional hidden Markov chain (BHMC), where $X$ is the unobserved process and $Y = (Y^1, Y^2)$ is the observed one. For a realization of $(X, Y)$

$$(x, y) = ((\omega_1, y_1), \ldots, (\omega_n, y_n)),$$

the likelihood is given by

$$L(x, y; \Theta) = \nu_1 \prod_{m=2}^{n} a_{m-1,n} \prod_{m=1}^{n} f_{im}(y_m),$$

where $\Theta = (\nu, \Lambda, (\mu_1, \Sigma_1), \ldots, (\mu_K, \Sigma_K))$. The size of $\Theta$ is $K^2 + 7K$ (i.e. $K$ for $\nu$, $K^2$ for $\Lambda$, $2K$ for $(\mu_1)_i = 1, \ldots, K$ and $4K$ for $(\Sigma_1)_{i=1, \ldots, K}$). Using the fact that each row of $A$ and $\sigma$ are probability distributions and the symmetry of $\Sigma$, this size can be reduced to $K^2 + 5K - 1$. We suppose that block components $\nu, \Lambda, (\mu_i, \Sigma_i), i = 1, \ldots, K$ of $\Theta$ are independent.

The classification problem is thus the estimation of the unobserved process realization $X = x$ from the observation $Y = y$. The Bayesian Maximum A Posteriori (MAP) and Maximum Posterior Mode (MPM) can be easily computed for the BHMC model. In this paper we will use the MPM method based on minimizing the mean ratio of wrongly classified points. Formally, for each $t$, the state $\omega_j$ that maximizes the a posteriori probability is such that

$$P(X_t = \omega_j \mid Y = y) = \max_{1 \leq i \leq K} P(X_t = \omega_i \mid Y = y).$$

The computation of the posterior marginal distribution is feasible thanks to the “Forward-Backward” method. Finally, the problem is: given the bi-dimensional observed process $Y$, we have to estimate the global parameter $\Theta$.

### III. BEM ALGORITHM

Generally, the maximum likelihood estimator has good statistical properties. The Expectation-Maximization (EM) algorithm [2], [26], [37] is one of the most popular method to approximate parameter $\Theta$ maximizing the likelihood $L(x, y; \Theta)$. Given an initial guess $\Theta^{(0)}$, the EM algorithm consists on generating iteratively a sequence $(\Theta^{(q)})_{q \in \mathbb{N}}$ that globally converges thanks to the theorem 10.5.2 in [25]. The two principal steps of EM are:

**E-step:** Determine $Q(\Theta, \Theta^{(q)}) = E_{\Theta^{(q)}}[\ln L(x, y; \Theta) \mid Y]$. 

**M-step:** Choose $\Theta^{(q+1)}$ to be the value that maximizes $Q(\Theta, \Theta^{(q)})$.

To adapt the EM algorithm to the hidden Markov chain with a bi-dimensional observed process, we need the following notations and definitions.

For $t \in \mathbb{N}$ and $\omega_i, \omega_j \in \Omega$, we define:

$$\psi_t(i, j) := P(X_t = \omega_i, X_{t+1} = \omega_j \mid Y = y)$$

and

$$\chi_t(i) := P(X_t = \omega_i \mid Y = y).$$

These probabilities can be expressed according to the forward-backward probabilities $\alpha^*_t(\cdot)$ and $\beta^*_t(\cdot)$ used in [26] as follows:

$$\psi_t(i, j) = \frac{\alpha^*_t(i) \alpha_{t+1} j f_{i}(y_{t+1}) \beta^*_{t+1}(j)}{\sum_{j=1}^{K} f_{i}(y_{t+1}) \sum_{j=1}^{K} \alpha^*_t(j) a_{ij}}$$

and

$$\chi_t(i) = \frac{\sum_{j=1}^{K} \psi_t(i, j)}{\sum_{i=1}^{K} \sum_{j=1}^{K} \psi_t(i, j)}. $$

where $\alpha^*_t(i) = P(X_t = \omega_i \mid Y_t = y_1, \ldots, Y_t = y_t)$ and $\beta^*_t(i) = \frac{P(Y_{t+1}=y_{t+1}, \ldots, Y_n=y_n \mid X_t=\omega_i)}{\sum_{i=1}^{K} P(Y_{t+1}=y_{t+1}, \ldots, Y_n=y_n \mid X_t=\omega_i)}$. The numerator and denominator in $\beta^*_t(i)$ are conditional joint density functions. These notations are usually used for simplicity.

Applying the two steps of EM algorithm to the proposed model BHMC (the obtained algorithm is noted BEM), we obtain the following equations: given estimations at iteration $q$, those at iteration $(q+1)$ are explicitly as follows:

For $i, j = 1, \ldots, K$,

$$\psi^{(q+1)}_{t}(i, j) = \frac{1}{n} \sum_{t=1}^{n} \chi^{(q)}_{t}(i) \psi^{(q+1)}_{t}(i, j), \quad \alpha^{(q+1)}_{ij} = \frac{\sum_{t=1}^{n} \psi^{(q)}_{t}(i, j)}{\sum_{t=1}^{n} \chi^{(q)}_{t}(i)},$$

and

$$\Sigma^{(q)}_{i} = \frac{\sum_{t=1}^{n} \chi^{(q)}_{t}(i) \chi^{(q)}_{t}(i) - \chi^{(q+1)}_{t}(i) \chi^{(q+1)}_{t}(i)}{\sum_{t=1}^{n} \chi^{(q)}_{t}(i)}.$$
A. Full conditional distribution for $A$

Using (1), for $i, j = 1, \ldots, K$ the estimation of $a_{ij}$ can be deduced from those of $c_{ij}$. So it's being sufficient to determine the full conditional distribution of $c_{ij}$.

For $u, v = 1, \ldots, K$,

$$\pi(c_{uv} | \cdot) \propto \pi(c_{uv}) \pi(Y | X = x, \nu, A, \mu, \Sigma)$$

$$\propto \pi(c_{uv}) \pi(X = x | \nu, A, \mu, \Sigma)$$

where $\mu = (\mu_i; 1 \leq i \leq K)$, $\Sigma = (\Sigma_i; 1 \leq i \leq K)$ and the symbol $\propto$ indicates proportionality up to a constant that does not depend on the considered variable.

On the other hand, including the fact that $X$ is an homogenous Markov chain and $x = (x_{ik})_{1 \leq i \leq r, 1 \leq k \leq r_i}$ is a realization of $r$ independent trajectories of $X$, we have

$$P(X = x | \nu, A) = \prod_{i=1}^{r} P(X_{it1} = x_{i1} | \nu)$$

$$= \prod_{k=2}^{r_i} P(X_{itk} = x_{ik} | X_{itk-1} = x_{i(k-1)}, \nu, A)$$

$$= \prod_{i=1}^{r} \prod_{k=2}^{r_i} P(X_{itk} = x_{ik} | \nu) \pi(X = x | \nu, A)$$

$$= \prod_{i=1}^{r} \prod_{k=2}^{r_i} \prod_{j=1}^{r_j} P(X_{itj} = x_{ij} | \nu) \pi(X = x | \nu, A)$$

So

$$\pi(c_{uv} | \cdot) \propto \pi(c_{uv}) \prod_{i=1}^{r} \prod_{k=2}^{r_i} \prod_{j=1}^{r_j} P(X_{itk} = x_{ik} | \nu) \pi(X = x | \nu, A)$$

To select the prior distribution $\pi(c_{uv})$, $Gamma(\alpha_{uv}, 1)$ is a natural choice since it leads to a Dirichlet distribution supposed for each row of the transition matrix $A$. The generation of the univariate full conditional distribution for $c_{uv}$ can be obtained by a Metropolis accept-reject algorithm.

**Remark 1.** Simulation from a Dirichlet distribution $D_K(\delta_1, \ldots, \delta_K)$ can be drawn thanks to the following result: if $\xi_1, \ldots, \xi_K$ are independent with $\xi_i$ having a $Gamma(\delta_i, 1)$ distribution, then

$$(\frac{\xi_1}{K}, \frac{\xi_2}{K}, \ldots, \frac{\xi_K}{K}) \sim D_K(\delta_1, \ldots, \delta_K)$$

which can be written as

$$\pi(\nu | \cdot) \propto \pi(\nu) \prod_{k=1}^{K} r_k^{\nu(k)}$$

where $r_k(1) := \sum_{i=1}^{r_i} \mathbb{1}_{\{x_{ik} = \omega_j\}}$ is the number of visits to state $\omega_j$ at initial time $t_1$ (i.e. $\mathbb{1}$ is the indicator function of set $E$). A conjugate prior for $\nu = (\nu_1, \ldots, \nu_K)$ is a Dirichlet distribution $D_K(\delta_1, \ldots, \delta_K)$ with known hyper-parameters $\delta_k > 0$, $k = 1, \ldots, K$ (i.e. $\pi(\nu) \propto \prod_{k=1}^{K} r_k^{\delta_k-1}$).

This choice is justified by the fact that $\nu_1, \ldots, \nu_K$ are parameters of a multinomial distribution. So we obtain

$$\pi(\nu | \cdot) \propto \prod_{k=1}^{K} (\nu_k^{r_k(1) + \delta_k-1})$$

which leads to

$$\pi(\nu | \cdot) \sim D_K(\delta_1 + r_1(1), \ldots, \delta_K + r_K(1)).$$

C. Full conditional distribution for $\mu_j$ and $\Sigma_j$

We suppose that $\Sigma_j, j = 1, \ldots, K$, are independent and $\mu_j$ depends only on $\Sigma_j, j = 1, \ldots, K$, the corresponding prior distributions are chosen to be

$$\mu_j | \Sigma_j \sim N_2(\tau_j, \Sigma_j) \quad \text{and} \quad \Sigma_j \sim Inv - Wishart(\beta_j, W_j^{-1})$$

where the hyper-parameters $\tau_j, \beta_j$ and $W_j^{-1}$ are supposed known.

Let’s consider the following definitions and notations: $N_i^j := \{k = 1, \ldots, r_i, x_{ik} = \omega_j\}$ is the set of times of visits to state $\omega_j$ in trajectory $i$, $N_j = \sum_{i=1}^{r} | N_i^j |$, $y_j = \frac{1}{N_j} \sum_{i=1}^{r} \sum_{k \in N_i^j} y_{ik}$ and $S_j = \sum_{k \in N_i^j} (y_{ik} - y_j)(y_{ik} - y_j)^T$, where $| N_i^j |$ is the cardinality of the finite set $N_i^j$ and $y = (y_{ik})_{1 \leq i \leq r, 1 \leq k \leq r_i}$ is the given realization of $Y$.

So, for $j = 1, \ldots, K$,

$$\pi(\Sigma_j, \mu_j | \cdot) \propto \pi(\Sigma_j, \mu_j) \pi(Y | X = x, \Sigma_j, \mu)$$

$$\propto \pi(\Sigma_j) \pi(\mu_j | \Sigma_j) \pi(Y | X = x, \Sigma_j, \mu).$$

On the other hand, by hypothesis $(H_1)$ and $(H_2)$ in section II, we have

$$\pi(Y | X = x, \Sigma, \mu)$$

$$\propto \prod_{i=1}^{r} \prod_{k=1}^{r_i} \pi(Y_{itk} | X_{itk} = x_{ik}, \Sigma x_{ik}, \mu x_{ik})$$

$$\propto \prod_{i=1}^{r} \prod_{k=1}^{r_i} |\Sigma x_{ik}|^{-\frac{1}{2}} e^{\frac{1}{2} (Y_{itk} - \mu x_{ik})^T \Sigma x_{ik}^{-1} (Y_{itk} - \mu x_{ik})}$$

Therefore, the full conditional distribution of $(\Sigma_j, \mu_j)$ is

$$\pi(\Sigma_j, \mu_j | \cdot) \propto \pi(\Sigma_j) \pi(\mu_j | \Sigma_j) \pi(Y | X = x, \Sigma_j, \mu_j).$$
given by
\[
\pi(y_j) = \frac{\pi(y_j | y_j)}{\int \pi(y_j | y_j) \, dy_j}
\]

This joint prior distribution is obtained using independence of block components of \( \Theta \) supposed in section II.

D. A bayesian algorithm for BHMC

We propose a bayesian algorithm for bi-dimensional hidden Markov chain model, principally based on the classical Gibb’s sampler and Metropolis accept-reject algorithms. For given hyper-parameters \( \alpha_j, \beta_j, \delta_j, \eta_j, \) and \( W_j^{-1} ; i, j = 1, \ldots, K, \) the main steps of BBHMC are the following:

The BBHMC Algorithm

E0. Select starting values (iteration \( q = 0 \)),
\( \Theta^{(q)} = (\nu^{(q)}, \eta^{(q)}, \mu^{(q)}, \Sigma^{(q)}, i, j = 1, \ldots, K) \)
and compute \( c_{ij}^{(q)} \) using (1).

E1. Iteration \( (q + 1) \)

E1.1 Generate a sample
\( x^{(q+1)} = \{ x^{(q+1)}_{ik}, i = 1, \ldots, r, k = 1, \ldots, r_i \} \)

E1.2 Generate a transition matrix
\( A^{(q+1)} = (a_{ij}^{(q+1)})_{1 \leq i, j \leq K} \)

E1.2.1 According to the following accept-reject algorithm:

E1.2.1.1 Generate a candidate
\( b_{ij} \sim N(c_{ij}^{(q)}, 1) \) and \( u \sim \mathcal{U}(0, 1) \)
such that \( b_{ij} \) and \( u \) are independent.

E1.2.2 Compute \( \prod(b_{ij}) \) and \( \prod(c_{ij}^{(q)}) \) by
\[
\prod(b_{ij}) = \prod_{l=1}^{r_i} \prod_{k=1}^{r_j} \{ \text{if } (x^{(q)}_{ik}, x^{(q)}_{jk}) \neq (i,j) \} + \prod_{l=1}^{r_i} b_{ij} \prod_{k=1}^{r_j} \{ (x^{(q)}_{ik}, x^{(q)}_{jk}) = (i,j) \}
\]
\[
\prod(c_{ij}^{(q)}) = \prod_{l=1}^{r_i} \prod_{k=1}^{r_j} c_{ij}^{(q)} \{ (x^{(q)}_{ik}, x^{(q)}_{jk}) \neq (i,j) \}
\]

E1.2.3 Compute
\( \rho_{ij}^{(q+1)} = \min \left\{ 1, \frac{q(b_{ij}) \prod(b_{ij})}{q(c_{ij}^{(q)}) \prod(c_{ij}^{(q)})} \right\} \)

E1.2.4 Compute
\( c = c_{ij}^{(q)} \{ u \leq \rho_{ij}^{(q+1)} \} + b_{ij} \{ u > \rho_{ij}^{(q+1)} \} \)

If \( c = c_{ij}^{(q)} \), we reject \( b_{ij} \) and return to E1.2.1.

E1.2.5 Compute \( a_{ij}^{(q+1)} \) (by formula in (1))

E2. Compute \( N_j, S_j, \delta_j \) and \( W(S_j, W_j) \) for \( j = 1, \ldots, K \)

E3. For \( j = 1, \ldots, K, \)

E3.1 Generate
\( \pi(\mu_j^{(q+1)}) \sim \mathcal{N}_2 (\tau_j, N_j, S_j, \delta_j, \eta_j, W(S_j, W_j)) \)

E3.2 Generate
\( \pi(\Sigma_j^{(q+1)}) \sim \text{Inverse - Wishart}(\delta_j + N_j, W(S_j, W_j)) \)

E3.3 Generate
\( \pi(\nu^{(q+1)}) \sim D_K (\delta_1 + r_1(1), \ldots, \delta_K + r_K(1)) \)

V. NUMERICAL SIMULATION

We interest to the performance of BEM and BBHMC algorithms by simulating different noisy Markov chains, and we compare the estimations obtained from the noisy data to those computed from the complete data. These last will be called
Empirical Values and will be denoted (EV). The computation of the empirical values \(c_{ij}^*, \mu_j^*\) and \(\Sigma_j^*\) for \(i, j = 1, \ldots, K\) makes itself by the following formulas:

\[
c_{ij} = \frac{1}{r_{ij}} \left| \sum_{t=1}^{r_{ij}} \ni_{ij} \right|
\]

\[
\mu_j^* = \frac{1}{N_j} \sum_{i \in N_j} y_{itk},
\]

\[
\Sigma_j^* = \frac{1}{N_j} \sum_{i \in N_j} (y_{itk} - \mu_j^*)(y_{itk} - \mu_j^*)^T
\]

In the following, we present the parameter initializations and the data scenarios, that has been used by many authors [7], [14], [29], after adapting them to the bi-dimensional case.

A. Parameter initializations

For \(i, j = 1, \ldots, K\) and \(d, d' = 1, 2\)

- Markov chain parameters:

  \[
  \nu_i = \frac{1}{K}, \quad \alpha_i = 0.5 \quad \text{and} \quad \alpha_{ij} = \frac{1}{2(K-1)} \quad \text{for} \quad i \neq j
  \]

- Noisy parameters:

  Mean vector:

  If \(K = 2p\), for \(1 \leq k \leq p\)

  \[
  \mu_{k+1}^d = m_d - 0.5(p-k)\sqrt{\sigma_{dd}}
  \]

  \[
  \mu_{K-k}^d = m_d + 0.5(p-k)\sqrt{\sigma_{dd}}
  \]

  If \(K = 2p + 1\), for \(1 \leq k \leq K\)

  \[
  \mu_{k+1}^d = m_d - 0.5(0.5K - k)\sqrt{\sigma_{dd}}.
  \]

  Variance-covariance matrix:

  \[
  \sigma_{dd} = \sigma_{dd}, \quad \sigma_{dd'} = c_{ij}\sigma_{dd'} \quad \text{for} \quad d \neq d'
  \]

where \(c_i = 0.1\) for numerical simulation and \(c_i = 0.9\) for image segmentation. The mean vector \((m^1, m^2)\) and the variance-covariance matrix \((\sigma_{11}, \sigma_{12}, \sigma_{21}, \sigma_{22})\) are computed using a generated trajectory of \(X\).

- Hyper-parameters:

  For \(1 \leq i, j \leq K\), \(\alpha_{ij} = 2\), \(\tau_i = (2, 2)'\), \(\beta_i = 1\), \(\delta_i = 1\), and \(W_i^{-1} = \left( \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right)\)

B. Markov chain and noisy data scenarios

While considering a sample of size \(n = 1000\) and a number of classes \(K = 2\), there are two factors of wish we study the influence on the performance of BEM and BBHMC algorithms: the noise and the Markov chain.

- For the noise factor, we consider the following three scenarios:

  (MD): means discriminating noise \(\mu_1 \neq \mu_2\) and \(\Sigma_1 = \Sigma_2\)

  (VD): variances discriminating noise \(\mu_1 = \mu_2\) and \(\Sigma_1 \neq \Sigma_2\)

  (MVD): means and variances discriminating noise \(\mu_1 \neq \mu_2\) and \(\Sigma_1 \neq \Sigma_2\).

The noise level depends strongly on the following two parameters:

\[
\Delta = (\mu_1^2 - \mu_2^2)^2 + (\mu_2^2 - \mu_1^2)^2 \quad \text{and} \quad \rho = \rho_2^2 / \rho_1^2,
\]

where

\[
\rho_2 = \sigma_2^2 / \sqrt{\sigma_1^2 \sigma_2^2} \quad \text{and} \quad \rho_1 = \sigma_1^2 / \sqrt{\sigma_1^2 \sigma_2^2}.
\]

We fix the parameters values for each scenario as follows:

- **MD**: \(\mu_1 = (1, 1)', \mu_2 = (1, 2)'\) and \(\Sigma_1 = \Sigma_2 = \left( \begin{array}{cc} 2 & 0.9 \\ 0.9 & 3 \end{array} \right)\).

- **VD**: \(\mu_1 = \mu_2 = (1, 1)', \Sigma_1 = \left( \begin{array}{cc} 0.25 & 0.2 \\ 0.2 & 4 \end{array} \right)\) and \(\Sigma_2 = \left( \begin{array}{cc} 5 & 5.3 \\ 5.3 & 9 \end{array} \right)\).

- **MVD**: \(\mu_1 = (1, 1.5)', \mu_2 = (2, 1)', \Sigma_1 = \left( \begin{array}{cc} 0.25 & 0.9 \\ 0.9 & 4 \end{array} \right)\) and \(\Sigma_2 = \left( \begin{array}{cc} 5 & 3.5 \\ 3.5 & 9 \end{array} \right)\).

C. Implementation parameters

There are several parameters which arise when implementing MCMC algorithms [39]. The algorithm must be run for a heating period, called burn-in, which depends on initial values of parameters. The aim of this period is to reduce the effect of initial values on posterior inference. Visual inspection of plots of the algorithm output \((\Theta^{(q)}, q = 0, 1, \ldots, T)\) with \(T\) is the global length of iterations, is the most obvious and commonly used method for determining the burn-in period. For the burn-in period length \(T_0\) of BBHMC algorithm, we take \(T_0 = 1000\) for numerical simulations and \(T_0 = 7000\) for image segmentation. After this period, we start the collection of estimations for \(T_1 = 1000\) supplementary iterations. So, the total number of iterations is \(T = T_0 + T_1\) and the estimation \(\hat{\Theta}\) of \(\Theta\) is given by the MMSE estimator \(\hat{\Theta} = \frac{1}{T} \sum_{q=T_0}^{T} \Theta^{(q)}\).

For BEM algorithm, we use an automatic stopping criteria based on the following distances (inspired to [7])

\[
D^2(\mu^{(q)}) = \frac{1}{K} \sum_{k=1}^{K} c_k \left[ (\mu_k^{(q)} - \mu_k^{(0)})^2 + \frac{1}{q+1} \sum_{l=0}^{q} \mu_l^{(q)} \right] + \frac{1}{q+1} \sum_{l=0}^{q} \mu_l^{(q)}
\]

\[
D^2(\sigma^{(q)}_{ij}) = \frac{1}{K} \sum_{k=1}^{K} c_k \left[ (\sigma_{ij}^{(q)} - \sigma_{ij}^{(0)})^2 + \frac{1}{q+1} \sum_{l=0}^{q} \sigma_{ij}^{(q)} \right] \quad 1 \leq i, j \leq 2
\]

The stopping of iterations is a function of quality estimation of parameters and stoutness of MPM method. The BEM
algorithm is stopped when \( D^2(\mu(q)) < \varepsilon \) and \( D^2(\sigma(q)) < \varepsilon \) for \( i,j = 1,2 \). We take \( (\varepsilon, c_k) = (10^{-4}, 1) \) for numerical simulations and \( (\varepsilon, c_k) = (10^{-2}, 10^{-3}) \) for image segmentation.

**D. Numerical results**

This subsection deals with numerical estimation for parameters and misclassification rates obtained by BEM and BBHMC, compared with those obtained by EV. These algorithms are tested on combinations of C1 and C2 with MD, VD and MVD. The BEM and BBHMC algorithms use the same initializations. The obtained estimations are presented in Tables II to V. In Table I, we present the misclassification rates of pixels computed by MPM method.

From Table I, we remark that BBHMC has globally a small advantage with respect to BEM mainly for C2+MD and C2+MVD scenarios which are considered as worst cases. Concerning estimations of \( \nu \) and \( \mu \), resumed in Tables II and III, the BEM algorithm takes advantage with respect to BBHMC algorithm with a stability advantage of estimations for the last. It seems that this stability is due to the estimations form (i.e. estimators obtained by the law number large).

Regarding Tables IV and V, the estimations of \( \mu \) and \( \Sigma \) obtained by BEM and BBHMC are very closed to those obtained by EV.

### Table I

**ERROR RATES FOR EACH CLASS (ER1 AND ER2) GIVEN BY EV, BEM AND BBHMC FOR DIFFERENT DATA SCENARIOS, WHERE ERi IS THE MISCLASSIFICATION RATE OF CLASS i. THE MEAN CPU TIME FOR BEM AND BBHMC IS 7.5**

<table>
<thead>
<tr>
<th>Data</th>
<th>Error rate(%)</th>
<th>EV</th>
<th>BEM</th>
<th>BBHMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1+MD</td>
<td>( E_R1 )</td>
<td>2.4</td>
<td>2.4</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>3.3</td>
<td>3.3</td>
<td>3.5</td>
</tr>
<tr>
<td>C1+VD</td>
<td>( E_R1 )</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>1.8</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>C1+MVD</td>
<td>( E_R1 )</td>
<td>0.1</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>C2+MD</td>
<td>( E_R1 )</td>
<td>22.6</td>
<td>23.7</td>
<td>22.6</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>18.5</td>
<td>19.0</td>
<td>18.5</td>
</tr>
<tr>
<td>C2+VD</td>
<td>( E_R1 )</td>
<td>6.7</td>
<td>7.1</td>
<td>6.9</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>12.7</td>
<td>12.9</td>
<td>12.7</td>
</tr>
<tr>
<td>C2+MVD</td>
<td>( E_R1 )</td>
<td>4.8</td>
<td>4.2</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>( E_R2 )</td>
<td>12.9</td>
<td>16.0</td>
<td>12.9</td>
</tr>
</tbody>
</table>

### Table II

**ESTIMATIONS OF COMPONENT \( \nu_1 \) BY EV, BEM AND BBHMC FOR DIFFERENT DATA SCENARIOS. THE STANDARD DEVIATION IS GIVEN BETWEEN BRACKETS FOR BBHMC ESTIMATIONS**

<table>
<thead>
<tr>
<th>Data</th>
<th>EV</th>
<th>BEM</th>
<th>BBHMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1+MD</td>
<td>0.522</td>
<td>0.523</td>
<td>0.490 (0.410×10^{-2})</td>
</tr>
<tr>
<td>C1+VD</td>
<td>0.522</td>
<td>0.523</td>
<td>0.490 (0.410×10^{-2})</td>
</tr>
<tr>
<td>C1+MVD</td>
<td>0.522</td>
<td>0.519</td>
<td>0.490 (0.410×10^{-2})</td>
</tr>
<tr>
<td>C2+MD</td>
<td>0.475</td>
<td>0.477</td>
<td>0.493 (0.021×10^{-2})</td>
</tr>
<tr>
<td>C2+MVD</td>
<td>0.475</td>
<td>0.501</td>
<td>0.493 (0.021×10^{-2})</td>
</tr>
</tbody>
</table>

**E. Level noise and misclassification error**

We consider different levels (\( \Delta, \rho \)) of MD, VD and MVD noises. These noises are combined with C1 and C2. For these combinations, the behavior of the misclassification rate, computed by MPM method, using EV is given in Table VI. The performance of BEM and BBHMC through the following three cases (\( \Delta, \rho \) = \( \{5,1\}, \{0,1\}, \{0,0.63\} \)) is presented in Fig. 1 to Fig. 6.

In each figure, we give the behavior of error rate (for each class) with respect to the iterations. In comparison with EV, we constat that BEM and BBHMC give good results for (\( \Delta, \rho \) = \( \{5,1\} \)) (see Fig. 1 and Fig.4): the error rates for BEM and BBHMC are nearer to those given by EV after \( T_I = 1000 \) iterations for BBHMC and after 50 iterations for BEM. So BEM outperforms BBHMC in this easiest case (i.e. lower level
TABLE V
ESTIMATIONS OF $\Sigma_1$ AND $\Sigma_2$ BY EV, BEM AND BBHMC FOR DIFFERENT DATA SCENARIOS. THE STANDARD DEVIATION IS GIVEN BETWEEN BRACKETS FOR BBHMC ESTIMATIONS.

<table>
<thead>
<tr>
<th>Data</th>
<th>$\Sigma_1$</th>
<th>$\Sigma_2$</th>
<th>$\Sigma_{BBHMC}$</th>
<th>EV</th>
<th>BEM</th>
<th>BBHMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1+MD</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{12}$</td>
<td>$\sigma_{21}$</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
</tr>
<tr>
<td></td>
<td>3.362</td>
<td>3.409</td>
<td>0.279</td>
<td>0.111</td>
<td>0.112</td>
<td>0.112 (0.001 × 10^{-5})</td>
</tr>
<tr>
<td>C1+VD</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{12}$</td>
<td>$\sigma_{21}$</td>
<td>0.248</td>
<td>0.345</td>
</tr>
<tr>
<td></td>
<td>4.194</td>
<td>4.147</td>
<td>0.244</td>
<td>0.208</td>
<td>0.214</td>
<td>0.227 (0.003 × 10^{-5})</td>
</tr>
<tr>
<td>C1+MVD</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{12}$</td>
<td>$\sigma_{21}$</td>
<td>2.484</td>
<td>2.244</td>
</tr>
<tr>
<td></td>
<td>6.784</td>
<td>6.792</td>
<td>0.223</td>
<td>0.123</td>
<td>0.111</td>
<td>0.112 (0.001 × 10^{-5})</td>
</tr>
<tr>
<td>C2+MD</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{12}$</td>
<td>$\sigma_{21}$</td>
<td>0.250</td>
<td>0.239</td>
</tr>
<tr>
<td></td>
<td>4.309</td>
<td>4.677</td>
<td>0.192</td>
<td>0.104</td>
<td>0.095</td>
<td>0.094 (0.004 × 10^{-5})</td>
</tr>
<tr>
<td>C2+VD</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{12}$</td>
<td>$\sigma_{21}$</td>
<td>4.554</td>
<td>3.017</td>
</tr>
<tr>
<td></td>
<td>11.583</td>
<td>11.501</td>
<td>1.045</td>
<td>0.145</td>
<td>0.033</td>
<td>0.038 (0.000 × 10^{-5})</td>
</tr>
</tbody>
</table>

noise). In all other cases, BBHMC works better than BEM: the error rates, after stabilization, are nearer to those given by EV for BBHMC than for BEM (see Fig. 2, Fig. 3, Fig. 4 and Fig. 6).

TABLE VI
THEORETICAL ERROR RATES FOR EACH CLASS ($ER_1$ AND $ER_2$) FOR C1 AND C2. $ER_{12}$ IS THE MISCLASSIFICATION ERROR OF CLASS 1.

<table>
<thead>
<tr>
<th>Markov chain</th>
<th>$\Delta = 5$</th>
<th>$\Delta = 2$</th>
<th>$\Delta = 0$</th>
<th>$\Delta = 5$</th>
<th>$\Delta = 2$</th>
<th>$\Delta = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\rho = 1$</td>
<td>$\rho = 1$</td>
<td>$\rho = 1$</td>
<td>$\rho = 1$</td>
<td>$\rho = 1$</td>
<td>$\rho = 1$</td>
</tr>
<tr>
<td>C1 $ER_1$</td>
<td>0.5</td>
<td>1.3</td>
<td>2.4</td>
<td>9.3</td>
<td>6.5</td>
<td>2.4</td>
</tr>
<tr>
<td>C2 $ER_2$</td>
<td>7.3</td>
<td>12.18</td>
<td>22.6</td>
<td>10.9</td>
<td>38.8</td>
<td>17.8</td>
</tr>
<tr>
<td>C1 $ER_{12}$</td>
<td>9.7</td>
<td>15.8</td>
<td>18.5</td>
<td>27.09</td>
<td>31.6</td>
<td>18.7</td>
</tr>
</tbody>
</table>

VI. UNSUPERVISED IMAGE SEGMENTATION

In the following, we compare BBHMC using Hilbert-Peano scan and BHMC using a noisy supplementary information (for example: proportion of pixels in the same class as the current pixel in a neighborhood of the last) on image. We consider that a neighborhood of a pixel is composed of 3, 5 or 8 pixels around it according to the following three positions: corner, border or inside. The comparison is made through a sample of four synthetic images given in Fig. 7 (Alphabet, Gibbs,
International Science Index, Mathematical and Computational Sciences Vol:5, No:12, 2011

The problem of this method is that the past and the future of a pixel doesn’t correspond always to its spatial context. Thus, when one considers the method “line by line”, two neighboring pixels and belonging to the same column are near spatially and distant in sense of Markov chain. To avoid the problem of the temporal and spatial contexts, a lot of authors use the Hilbert-Peano scan [6], [16], [19]. In this method, the noisy image is firstly transformed into one dimension chain using Hilbert-Peano scan and the segmented image is reconstructed using an inverse Hilbert-Peano scan. The proposed model BHMC using noisy supplementary information overcomes the problem of adequacy between temporal and spatial context without using the Hilbert-Peano scan.

**Letter B and Ring.** These standard images are used for tests in many image segmentation procedures [3], [7], [12], [23], [35]. Also, we compare performance (error rates) of BEM and BBHMC by applying them on the same synthetic images. These images are corrupted by the following noises:

**MD:** $\mu_1 = (112, 112), \mu_2 = (115, 116)$ and $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 300 & 105 \\ 105 & 360 \end{pmatrix}$. These values correspond to $(\Delta, \rho) = (0, 1)$.

**VD:** $\mu_1 = \mu_2 = (112, 112)'$, $\Sigma_1 = \begin{pmatrix} 300 & 325 \\ 325 & 360 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 500 & 350 \\ 350 & 900 \end{pmatrix}$. These values correspond to $(\Delta, \rho) = (25, 1)$.

**MVD:** $\mu_1 = (112, 112)'$, $\mu_2 = (115, 116)'$, $\Sigma_1 = \begin{pmatrix} 300 & 325 \\ 325 & 360 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 500 & 350 \\ 350 & 900 \end{pmatrix}$. These values correspond to $(\Delta, \rho) = (0, 0.27)$.

To appreciate visually the level of the considered noises, some examples are given in Fig.8. The segmentation result of these examples by BBHMC with MPM method is presented in Fig. 9.

### A. BHMC without and with Hilbert-Peano scan

The simplest idea for modeling an image by Markov chain is to consider this image “line by line” or “column by column.”

In Tables VII and VIII, we give the error rates for each class ($ER_1$ and $ER_2$) obtained by BHMC with noisy supplementary information and BHMC with Hilbert-Peano scan, where images used are resized to $32 \times 32$. The estimation parameter problem of the hidden Markov chain is not landed here; the noise and chain parameters are estimated from EV and are used by MPM method for the segmentation. From Table VII, we constat that the two approaches doesn’t give satisfactory response for class 2 in all corrupted Alphabet cases. May be this is due to the image size, witch is small. In all other cases, BHMC with noisy supplementary information gives a reduced mean error rate with respect to the BHMC using Hilbert-Peano scan. The mean reduction factor is 2.14, 1.33 and 1.71 for Gibbs, Letter B and Ring respectively.

### B. BEM and BBHMC performances

Before analyzing the results, let us note that for each statistical image segmentation method, there exists a theoretical error, obtained with the true parameter values [3]. In our case this error is estimated from empirical values EV (i.e. obtained from real and noisy images). For a simplification reason, we just interest to the error rates (wrongly classified pixels) and we do not give the parameters estimation. Thus we present,
in Table IX, the error rates for each class (\(E_{R1}\) and \(E_{R2}\)) obtained by EV, BEM, and BBHMC on corrupted images. We constat that BEM and BBHMC, compared to EV, prove successful in image Letter B. This can be explained by the fact that the two pixel classes are well separated. Also we constat that BEM is better than EV for BBHMC in all cases, but the mean error rates given by BBHMC are significantly reduced than those obtained by BEM and they are nearer to those given by EV. The mean reduction factors between BBHMC and BEM are 2.65, 1.39, 1.79 and 1.53 for Alphabet, Gibbs, Letter B and Ring respectively. All the estimations, for each data scenario, are obtained by BEM and BBHMC for a mean CPU time equal to 36.5.

![Image](image-url)

**Fig. 9.** Alphabet+VD, Gibbs+MVD, Letter B+MD and Ring+VD segmented by BBHMC with MPM method

### VII. CONCLUSION

We have proposed a bi-dimensional Markov chain model, BBHMC, mathematically designed by \((X,Y)\) where \(X\) is the unobserved Markov chain and \(Y = (Y^1, Y^2)\) is the observed process. In previous works \(Y^1\) and \(Y^2\) represent two images when, in our case \(Y^1\) represent noisy image and \(Y^2\) represent noisy supplementary information on the studied image (we have taken \(Y^2\) as proportion of pixels in the same class as the current pixel in a neighborhood of the last). For the proposed model, we have considered a bayesian algorithm BBHMC to estimate the model parameters. We have compared BBHMC to the classical Expectation-Maximization algorithm in bi-dimensional case, BEM. The results of these two algorithms are evaluated by empirical values, EV, computed from complete data. Our propositions are favorably tested on simulation and real data. We have considered different data scenarios and we have obtained satisfactory results for our approach: the obtained parameters estimation are nearer to those given by EV and the mean factor reduction of error rates with respect to BEM is between 1.33 and 2.65 for image segmentation. Also the proposed model BBHMC outperforms (see Tables VII and VIII) the same model considered with Hilbert-Peano scan.

Actually, we interest to the generalization of this work to multi-dimensional case by integrating supplementary information to each image (i.e. \(Y = (Y^1, Y^{11}, Y^2, Y^{22}, \ldots, Y^M, Y^{MM})\), where \(Y^i\) is an image and \(Y^{ij}\) is a supplementary information on \(Y^j\), for \(i = 1, \ldots, M\) with \(M > 2\).

### REFERENCES


---

**TABLE VII**

<table>
<thead>
<tr>
<th>Model</th>
<th>Alphabet</th>
<th>Gibbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MD</td>
<td>YD</td>
</tr>
<tr>
<td>BHMC</td>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>(E_{R1})</td>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>(E_{R2})</td>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>BHMC with (E_{R1})</td>
<td>0.19</td>
<td>0.09</td>
</tr>
<tr>
<td>Hilbert-Peano (E_{R2})</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**TABLE VIII**

<table>
<thead>
<tr>
<th>Model</th>
<th>Alphabet</th>
<th>Gibbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MD</td>
<td>YD</td>
</tr>
<tr>
<td>BHMC</td>
<td>7.3</td>
<td>11.05</td>
</tr>
<tr>
<td>(E_{R1})</td>
<td>45.3</td>
<td>47.3</td>
</tr>
<tr>
<td>(E_{R2})</td>
<td>7.3</td>
<td>6.1</td>
</tr>
<tr>
<td>BHMC with (E_{R1})</td>
<td>66.1</td>
<td>70.8</td>
</tr>
<tr>
<td>Hilbert-Peano (E_{R2})</td>
<td>7.3</td>
<td>6.1</td>
</tr>
</tbody>
</table>

**TABLE IX**

<table>
<thead>
<tr>
<th>Data</th>
<th>Error rates (%)</th>
<th>EV</th>
<th>BEM</th>
<th>BBHMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.08</td>
<td>7.89</td>
<td>2.77</td>
</tr>
<tr>
<td>+ MD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.08</td>
<td>7.89</td>
<td>2.77</td>
</tr>
<tr>
<td>+ VD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.87</td>
<td>7.89</td>
<td>3.05</td>
</tr>
<tr>
<td>+ MVD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>+ MD</td>
<td>(E_{R2})</td>
<td>1.70</td>
<td>1.00</td>
<td>1.29</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>8.10</td>
<td>12.34</td>
<td>8.29</td>
</tr>
<tr>
<td>+ VD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>8.27</td>
<td>12.75</td>
<td>8.57</td>
</tr>
<tr>
<td>+ MVD</td>
<td>(E_{R2})</td>
<td>0.27</td>
<td>0.00</td>
<td>0.24</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>1.95</td>
<td>2.55</td>
<td>1.93</td>
</tr>
<tr>
<td>+ MD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.04</td>
<td>2.76</td>
<td>2.16</td>
</tr>
<tr>
<td>+ VD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.71</td>
<td>2.76</td>
<td>2.13</td>
</tr>
<tr>
<td>+ MVD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.81</td>
<td>4.53</td>
<td>2.87</td>
</tr>
<tr>
<td>+ MD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>3.05</td>
<td>4.70</td>
<td>3.15</td>
</tr>
<tr>
<td>+ VD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Alphabet</td>
<td>(E_{R1})</td>
<td>2.99</td>
<td>4.73</td>
<td>3.11</td>
</tr>
<tr>
<td>+ MVD</td>
<td>(E_{R2})</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>
non supervisée d’images, Actes de Quatorzième Colloque GRETSI 93, Juan-les-Pins, France, pp. 105-108, 1993.


