Pattern Anal Applic (2009) 12:137–150
DOI 10.1007/s10044-008-0110-5

THEORETICAL ADVANCES

Image change detection from difference image through deterministic simulated annealing

Gonzalo Pajares · José J. Ruz · Jesús M. de la Cruz

Received: 9 March 2007 / Accepted: 14 February 2008 / Published online: 14 March 2008

Abstract This paper proposes an automatic method based on the deterministic simulated annealing (DSA) approach for solving the image change detection problem between two images where one of them is the reference image. Each pixel in the reference image is considered as a node with a state value in a network of nodes. This state determines the magnitude of the change. The DSA optimization approach tries to achieve the most network stable configuration based on the minimization of an energy function. The DSA scheme allows the mapping of interpixel contextual dependencies which has been used favorably in some existing image change detection strategies. The main contribution of the DSA is exactly its ability for avoiding local minima during the optimization process thanks to the annealing scheme. Local minima have been detected when using some optimization strategies, such as Hopfield neural networks, in images with large amount of changes, greater than the 20%. The DSA performs better than other optimization strategies for images with a large amount of changes and obtain similar results for images where the changes are small. Hence, the DSA approach appears to be a general method for image change detection independently of the amount of changes. Its performance is compared against some recent image change detection methods.

G. Pajares (✉)
Dpto. Ingeniería del Software e Inteligencia Artificial,
Facultad de Informática, Universidad Complutense,
28040 Madrid, Spain
e-mail: pajares@fdi.ucm.es

J. J. Ruz · J. M. de la Cruz
Dpto. Arquitectura de Computadores y Automática,
Facultad de Informática, Universidad Complutense,
28040 Madrid, Spain

Keywords Image change detection · Difference images · Simulated annealing · Markov random fields

1 Introduction

The automatic image change detection methods are suitable for many vision computer applications [30]. For instance, video surveillance [7, 26, 31,36–38], analysis of multitemporal remote sensing images [3–5], tracking systems of moving objects [24], medical diagnosis [2] or driver assistance systems [12] among others.

The goal is to identify the changed pixels between two images captured at different time periods; one of the images is the reference. Different factors cause changes: appearance or disappearance of objects, motion of objects relative to the background, shape change of objects or environment modifications (buildings, fires, etc.) [30]. The techniques based on background modeling are out of the scope of this paper. The reader is referred to references [7, 8, 30].

According to the classification proposed by Radke et al. [30] there are several major types of change detection schemes for detecting the difference between two images, I_1 and I_2, of the same scene captured at different time periods. In Pajares [27] is included an introduction about some of the above schemes.

1. *Simple differencing*: Given a pixel location (x, y), the difference associated to this pixel is given as: $D(x, y) = |I_2(x, y) - I_1(x, y)|$. A change is detected at this location if $D(x, y)$ is large as compared to a threshold T. So, under this approach the goal is to compute the threshold T. A study about some histogram-based automatic thresholding algorithms for choosing T is provided in Rosin and Ioannidis [31]. In Wu et al. [38] and Lu and Suganthan [26] some

improvements are proposed in order to solve some drawbacks of some thresholding methods.

2. *Significance and hypothesis tests models*: A change is declared at a given pixel location (x, y) based on two competing hypotheses: the null hypothesis H_0 or the alternative hypothesis H_1, corresponding to *no-change* and *change* decision respectively. Aach and Kaup [1] assume that the difference image $D(x, y)$ is characterized by two zero-mean Gaussian distributions with variances σ_0^2 and σ_1^2 for H_0 and H_1 respectively. The goal of this approach is to estimate a changed binary mask Q given D. Bruzzone and Fernandez [5] estimate the parameters of the mixture distribution at each pixel location $p(D(x,y))$, which is modeled as

$$p(D(x,y)) = p(D(x,y)|H_0)P(H_0) + p(D(x,y)|H_1)P(H_1) \tag{1}$$

The conditional estimations $p(D(x,y)|H_0)$, $p(D(x,y)|H_1)$ and $P(H_0)$, $P(H_1)$ are carried out by using the iterative expectation maximization (EM) algorithm [9] until the convergence. They assume Gaussian distributions for the a priori probabilities, where the parameters to be estimated are the means μ_0, μ_1 and variances σ_1^2, σ_0^2, respectively. The initial values of the estimates are determined from D. In this approach an interpixel spatial relation is applied in order to achieve consistency during the estimation.

3. *Predictive models*: There are spatial and temporal models that exploit the close relationships between nearby pixels. In the spatial models, given two blocks of two intensity values, a polynomial function of the pixel coordinates is fitted to the intensity values. Hsu et al. [16] use likelihood ratio tests using constant, linear or quadratic models for the image blocks. A change is detected if the corresponding blocks are best fit by different polynomial coefficients. In temporal models Jain and Chau [17] assumed each pixel was identically and independently distributed according to the same Gaussian distribution related to the past by the same coefficient. They derive maximum likelihood estimates of the mean, variance and correlation coefficient at each point in time. A change is detected when the image intensities are independent.

4. *Shading models*: Based on linear dependencies was originally proposed by Durucam and Ebrahimi [11]. Given the pixel location (x, y) a vector of n neighboring intensity pixels, $\mathbf{x} = \{x_1,..., x_n\}$ is built. This pixel is a unchanged/changed pixel in the next frame if $\boldsymbol{x}$ and the new vector $\mathbf{y} = \{y_1,..., y_n\}$ built from the new frame are linearly dependent/independent. Assuming that $\boldsymbol{x}$, $\boldsymbol{y}$ are two linearly dependent vectors with no components zero, then the ratio of their components is constant, i.e., $x_1/y_1 = x_2/y_2 = \cdots = x_n/y_n =$ *constant*. In a 8-connected neighborhood this is expressed as: $I_1(x-1, y-1)/I_2(x-1, y-1) = I_1(x-1, y)/I_2(x-1, y) = \cdots = I_1(x+1, y+1)/I_2(x+1, y+1) =$ *constant*. Hence, in the 8-connected neighborhood, under the linear dependence, the variance is zero, as one can infer from the variance definition, $\sigma^2(x,y) = \frac{1}{n-1}\sum_{(x,y)\in A}\left(I_1(x,y)/I_2(x,y) - \mu(x,y)\right)^2$; where $\mu(x,y) = \frac{1}{n}\sum_{(x,y)\in A}\left(I_1(x,y)/I_2(x,y)\right) = constant$ and n is the number of members in the neighborhood A of (x, y), i.e., eight in this approach. Under linear independence, the variance should not be zero. Skifstad and Jain [32] and Liu et al. [24] apply this model and determine a threshold empirically. Considering that the intensity image is the product of the illumination $I_l(x, y)$ from the light source in the scene and the reflectance $I_r(x, y)$ of the object surface to which (x, y) belongs, the intensity can be expressed as $I(x, y) = I_l(x, y)I_r(x, y)$. The reflectance component depends only on the intrinsic properties of the object surface. Hence, given two images I_1, I_2 and assuming the illumination constant, a change is detected based on the reflectance components.

5. *Background modeling*: Given two consecutive frames $I_i(x, y)$ and $I_{i\text{-}1}(x, y)$, Chang et al. [7] determine that a pixel has changed if $|R_i\ (x, y)| > T$ otherwise it belongs to the background. R_i is computed as $R_i(x, y) = M_{i\text{-}1}(x, y)$ $D_i(x, y)$; D_i is the difference image between I_i and I_{i-1}. The first time M_{i-1} is set to zero, next during the following steps R_i is assigned to M_i. Carlotto [6] models the background as a Gaussian distribution computing its representative mean value and covariance matrix. A pixel is classified as changed if the Mahalanobis distance [9] between its difference value and the mean is large. Paragios and Deriche [29] and Stauffer and Grimson [35] assume that changed pixels belong to moving objects otherwise they are unchanged belonging to the background. A mixture probability density function is estimated assuming two classes according to the above consideration.

6. *Change mask consistency*: As reported in Radke et al. [30] the methods using thresholding give noisy results, isolated changed/unchanged pixels, holes or jagged boundaries. To overcome this drawback most methods try to apply a consistency criterion. One important framework used by these methods to enforce the contextual information is Markov Random Fields (MRF). Under the MRF framework, a probability density function (PDF) is defined for each pixel location in the difference image D. The PDF generally follows the Gibbs distribution and embeds an energy term. The goal of the approaches based on change mask consistency is to maximize the PDF at each pixel location or equivalently minimize the energy. Bruzzone and Fernandez-Prieto [4], Aach and Kaup [1], Kasetkasen and Varshney [19], Liu [23] and Liu et al. [25] basically compute two energy terms. Previously each pixel is labeled

as changed/unchanged according to a thresholding technique applied to the difference image D. A first energy term is computed from the joint density function of the pixel values in D given the label. This term takes into account a kind of self-information for each pixel location. A second energy term is computed through the interactions between a pixel and its neighbors taking into account their labels. This term in the above approaches is termed the consistency term. Assuming a random field defined by the PDFs, we can see in the above approaches that the energy for each pixel is computed based only on the own pixel and its neighbors, out of the neighborhood the contribution to the energy of the pixels is null. This implies that the random fields fulfil the so called Markov condition for spatial descriptions as the images, i.e., the field is termed a MRF. The MRF is an approach which has been broadly used in image analysis [10]. Liu et al. [25] compute the consistency by applying the mean field theory (MFT) which assumes that the impacts from the neighbors can be approximated by an average field. In Pajares [27] the change detection problem is focused minimizing an energy function through the analog hopfield neural network (HNN) paradigm. Under this paradigm, the energy function assumes a trade-off between the self-information and the consistency. Also, under the HNN approach the consistency is extended so that the interactions in a neighborhood around a pixel location are based not only in the labels but also in the joint density function values of the neighbors, i.e., this implicitly assumes the Markovian condition. This extension and the analog properties of the HNN paradigm make of this method a valid approach for the set of images tested as compared with other existing image change detection strategies.

Unfortunately, through additional experiments we have verified that for images where the amount of changes surpasses the 20% the performance of the HNN approach decreases (see Fig. 5 and related comments in Sect. 3.3). This is because there is an important number of these difference images in which the energy falls in local minima that are not global optimum. This behavior of the Hopfield neural network is reported in Haykin [15]. The change mask consistency approaches, involving both contextual and self information, perform favorably. The deterministic simmulated annealing (DSA) is also an energy optimization based approach which can embed contextual and self information with the advantage that it can avoid local minima. Indeed, according to Geman and Geman [13] and reproduced in Haykin [15] when the temperature involved in the simulated annealing process satisfies some constraints (explained in the Sect. 2.5) the system converges to the minimum global energy which is controlled by the annealing scheduling instead of the nonlinear first-order differential equation used in HNN. This is the main difference of the proposed DSA technique with respect to the HNN approach.

In Kasetkasen and Varshney [19] the stochastic simulated annealing (SSA) is used to minimize the energy under the MRF framework. The results are binary labels indicating only changed/unchanged pixels. In Duda et al. [9] it is reported that SSA is slow due to its discrete nature as compared to the analog nature of the DSA.

In summary, we focus on the DSA approach, making the main contribution of this paper, because of the following set of advantages: (a) the contextual and self information can be mapped under an energy function; (b) the annealing scheduling allows avoiding local minima; (c) the optimization process corrects a posteriori the initial errors derived from a thresholding approach and (d) the analog nature allows to obtain the strength of the change for each pixel location.

The paper is organized as follows. In Sect. 2, the DSA process is described including the mapping of the self-information and consistencies. The performance of the method is illustrated in Sect. 3, where a comparative analysis against other existing image change detection strategies is carried out. Finally, in Sect. 4, there is a discussion of some related topics.

2 Deterministic simulated annealing process

We build a network of nodes, so that each pixel location (x, y) in the reference image or equivalently in the difference image D is associated to a node i. The node i is interconnected to the node j through a symmetric synaptic weight w_{ij} which is to be defined later. Moreover, each node i has associated a state value s_i which will be set through an activation function. A commonly activation function is the hyperbolic tangent one, which is bounded by -1 and +1 values [15]. Because of the analog nature of the DSA approach and taking into account the above limits, the s_i values range in the continuous interval $[-1, +1]$, where -1 and +1 indicate, from our point of view, a secure unchanged and changed pixels respectively. Other values in $[-1, +1]$ measure the degree of the change. The states are updated after each step t. Based on the change mask consistency strategies described in the point 6 of the introduction, we assume that our method falls under the MRF framework [10]; where instead of maximizing a PDF we minimize an energy function, which is equivalent. The updating process for each node i, is carried out through the DSA optimization approach assuming that the energy must be minimum. This is carried out through a *regularization* coefficient which computes the consistency between the states of the nodes in a given neighborhood and a *data* coefficient which computes the consistency between the difference image data values in D also in the same neighborhood. This is the neighborhood which characterizes the MRF as stated in the previous section. Both, regularization and data coefficients represent a trade-off between them, so that they can be mutually compensated. The own state value (i.e., the self-information) is mapped during the DSA updating process. A network initialization process is required before the updating process is triggered. These issues are addressed in the following subsections.

2.1 Network initialization

The network initialization is carried out by exploiting the characteristics of the difference image, D. We use the initialization strategy, described in Bruzzone and Fernandez-Prieto [5]. From the histogram $h(D)$ of the difference image, we compute the minimum and maximum values $m = \min\{D\}$ and $M = \max\{D\}$ and then two thresholds T_0 and T_1 as $T_0 = M_D\ (1 - \alpha) + \alpha m$ and $T_1 = M_D\ (1 + \alpha) - \alpha m$, where M_D is the middle value of $h(D)$, i.e., $M_D = \frac{1}{2}(m + M); \alpha \in (0, 1)$, set to 0.5 in this paper. Generally, m is zero because the most typical case is that some pixel is unchanged.

The thresholds T_0 and T_1 divide the histogram into three zones, Z_1, Z_2 and Z_3 as follows: $Z_1 = \{D(x,y)|m \leq D(x,y) \leq T_0\}$; $Z_2 = \{D(x, y)|\ T_0 < D(x, y)\ \leq T_1\ \}$ and $Z_3 = \{D(x, y)|T_1 < D(x, y) \leq M\}$. Z_1/Z_3 define two ranges where pixels belonging to D can be identified as unchanged/changed respectively. Z_2 defines the ambiguous zone, in which pixels cannot be identified as either changed or unchanged.

Based on the above, each node i is initialized with a state value as follows:

$$s_i^{t=0} = \begin{cases} -1 & \text{if} \quad D(x,y) \in Z_1 \\ +1 & \text{if} \quad D(x,y) \in Z_3 \\ -1 \text{ or } +1(\text{randomly}) & \text{if} \quad D(x,y) \in Z_2 \end{cases} \tag{2}$$

A node with its state value equal to −1/+1 is clearly identified as unchanged/changed. Nodes with values ranging in (−1, +1] are changed nodes with different strengths. The states associated to Z_2 are randomly initialized through a uniform distribution.

2.2 Statement of problem

The goal of the proposed method is to determine the magnitude of the change of each node from the initial state values and by applying consistency through the data and regularization coefficients and also through the own state value. This is achieved basically through the DSA optimization process. The process must evolve by increasing the state value of a changed node towards +1 and decreasing the state of any unchanged node towards −1. This implies that any node (changed or unchanged) can modify its state, but also they could stay with stable states even if they are changed nodes.

Suppose a network with N nodes. The simulated annealing optimization problem is: modify the analog values s_i so as to minimize the energy [9, 15]

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} s_i s_j \tag{3}$$

where w_{ij} is the symmetric weight interconnecting two nodes i and j and can be positive or negative ranging in [−1, +1]. Each w_{ij} determines the influence that the node j exerts on i trying to modify the state s_i, Fig. 1 displays how this influence is exerted through the consistencies mapped as data and regularization coefficients. According to Duda et al. [9] the self-feedback weights must be null (i.e., $w_{ii} = 0$). The DSA approach tries to achieve the most network stable configuration based on the energy minimization. From (3) one can see that this expression requires the

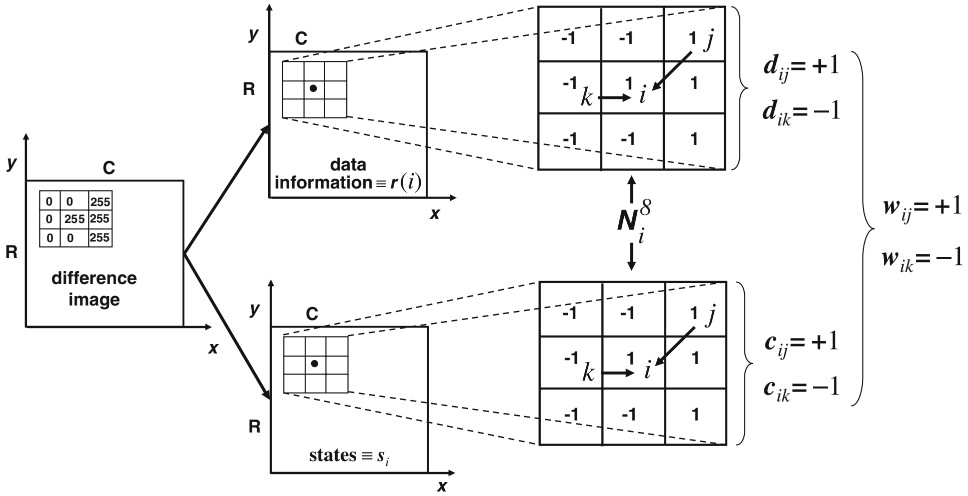

Fig. 1 Pedagogical example displaying the influence exerted by nodes j and k on node i

computation of w_{ij} and the states of the nodes s_i and s_j;w_{ij} is obtained as a weighted sum of the regularization and the data coefficients, it involves two nodes based on a neighborhood relation; s_i and s_j are obtained after the corresponding updating process.

2.3 Data coefficient

In Pajares [27] also a relation among the pixels is described by defining a kind of data consistency. We map the data information following Bruzzone and Fernandez-Prieto [5] according to the description in the *significance and hypothesis tests models*. Given the difference image D we formulate the Bayes rule for each pixel location (x, y) considering the H_0 and H_1 hypotheses as follows:

$$P(H_k|D(x,y)) = \frac{p(D(x,y)|H_k)P(H_k)}{p(D(x,y))} \tag{4}$$

where $p(D(x,y))$ is given in the Eq. (1) and the conditional probabilities $p(D(x,y)|H_k)$ and the a priori $P(H_k)$ are estimated through the EM algorithm. The initialization is carried out through the process described in the Sect. 2.1.

A node in the network is declared as changed or unchanged if it is associated to H_1 or H_0, respectively. This membership assignment is carried out according to the following equation,

$$\begin{aligned} H_n &= \arg \max_{H_n \in \{H_0,H_1\}} \{P(H_n|D(x,y))\} \\ &= \arg \max_{H_n \in \{H_0,H_1\}} \{P(H_n)p(D(x,y)|H_n)\} \end{aligned} \tag{5}$$

From (5) a *data map* is built for each pixel location, i.e., for each node (pixel) $i \equiv (x, y)$ in the network. According to the maximum value in (5), H_0 or H_1 is selected, i.e., $n = 0$ or 1 consequently. So, for each node i we compute the data information $r(i)$ as follows:

$$r(i) = \begin{cases} -P(H_0|D(i)) & \text{if} \quad H_n = H_0 \\ +P(H_1|D(i)) & \text{if} \quad H_n = H_1 \end{cases} \tag{6}$$

Each node i has associated its m-connected neighborhood, N_i^m, m is set to 8 in this paper. The data consistency between nodes i and j in the network is measured through a similarity measurement by the data coefficient d_{ij} as follows:

$$d_{ij} = \begin{cases} 1 - |r(i) - r(j)| & j \in N_i^m, j \neq i \\ 0 & j \notin N_i^m \end{cases} \tag{7}$$

From (7) we can see that d_{ij} ranges in [−1, +1] where the lower/higher limit means minimum/maximum data consistency, respectively. Hence, according to the Eq. (4) the data coefficient measures how similar are the probabilities of change for the nodes i and j. Indeed, assume that i and j are both changed or unchanged nodes simultaneously with identical magnitude in the difference image D and $j \in N_i^m$, $j \neq i$; hence both are associated to the same hypothesis H_0 or H_1 respectively and $r(i) = r(j)$. This implies that $d_{ij} = 1$ when i and j are of the same category and magnitude. Now consider that they belong to different categories, i.e., i is changed and j unchanged or viceversa with $j \in N_i^m$, $j \neq i$. Assume that i is with H_0 and j with H_1 and they both take the maximum values, i.e., $P(H_0|D(i)) = 1$ and $P(H_1|D(j)) = 1$; according to (6) $r(i) = -1$ and $r(j) = +1$. The same is applicable if i is with H_1 and j with H_0. This leads to $d_{ij} = -1$ which indicates maximum inconsistency. Any similarity measurement and norm could be used in the Eq. (7).

2.4 Regularization coefficient

In Bruzzone and Fernandez-Prieto [5], Aach and Kaup [1], Kasetkasen and Varshney [19] and Pajares [27] a relation among the pixels is described by defining a kind of contextual consistency. Given the node i with its state value s_i^t and a set of nodes j, where $j \in N_i^m$ with state values s_j^t. The node i achieves a high consistency when s_i^t and s_j^t have both similar values. This is mapped into the regularization coefficient c_{ij} through a similarity measurement as follows:

$$c_{ij} = \begin{cases} 1 - \left| s_i^{t-1} - s_j^{t-1} \right| & j \in N_i^m \\ 0 & j \notin N_i^m \end{cases} \tag{8}$$

From (8) we can see that c_{ij} varies with the iteration and ranges in [−1, +1] where the lower/higher limit means minimum/maximum consistency, respectively. As in d_{ij}, any similarity measurement and norm could be used in the Eq. (8). As one can see from the Eq. (8), the c_{ij} values are computed taking into account only the previous state values.

2.5 Simulated annealing: updating process

From (7) and (8) we combine d_{ij} and c_{ij} as the averaged sum, taking into account the signs,

$$\begin{aligned} &W_{ij} = \gamma d_{ij} + (1-\gamma)c_{ij}; \quad w_{ij} = \left[\mathrm{sgn}\left(W_{ij}\right)\right]^{v+1} W_{ij}; \\ &\mathrm{sgn}\left(W_{ij}\right) = \begin{cases} -1 & W_{ij} \leq 0 \\ +1 & W_{ij} > 0 \end{cases} \end{aligned} \tag{9}$$

$\gamma \in [0,1]$ represents the trade-off between both coefficients. After a set of experiments we have chosen $\gamma = 0.65$ because the state values are already involved directly in the energy computation through the Eq. (3). This avoids the over contribution of the state values in the energy value; sgn is the *signum function* and v is the number of negative values in the set $C \equiv \{W_{ij}, s_i, s_j\}$, i.e., given $S \equiv \{q \in C/q < 0\} \subseteq C$, $v = \mathrm{card}(S)$. The expressions in (9) take into account that the energy must achieve its minimum value for stable states.

Figure 1 shows a pedagogical example about the influence exerted, through the weights w_{ij}, w_{ik}, on node i by the nodes j and k according to the Eq. (9) from the data and regularization coefficients d_{ij}, d_{ik}. This is achieved from the difference image of size RxC based on the neighborhood N_i^8. As one can see the influence exerted by the nodes j and k is +1 and -1 respectively, this is because of the similarity of j and dissimilarity of k with respect the node i.

The simulated annealing process was originally developed in Kirkpatrick et al. [21] and Kirkpatrick [20]. In this paper we have implemented the approach described in Duda et al. [9] and Haykin [15]. According to Duda et al. [9] the DSA process is computationally faster than the SSA process. We have verified this assertion by implementing both versions (deterministic and stochastic), obtaining very similar solutions and identical performance in terms of correct changed or unchanged nodes. Nevertheless, the deterministic version has been faster than the stochastic, by exactly two orders of magnitude. This agrees with Duda et al. [9]. Moreover, we have not found problems to reach the global minimum under the deterministic version; this is because the DSA is initially guided (not randomly) during the initialization.

Following the notation in Duda et al. [9], let $l_i^t = \sum_j w_{ij} s_j^t$ be the force exerted on node i by the other nodes $j \in N_i^m$ at the iteration t; then the new state s_i^{t+1} is obtained by adding the fraction $f(\cdot, \cdot)$ to the previous one as follows,

$$s_i^{t+1} = \frac{1}{2}\left[f\left(l_i^t, T(t-1)\right) + s_i^t\right] = \frac{1}{2}\left[\tanh\left(l_i^t / T(t)\right) + s_i^t\right] \tag{10}$$

where t represents the iteration index. The fraction $f(\cdot, \cdot)$ depends upon l_i and T at the iteration t.

This equation differs from the updating process in Duda et al. [9] because we have added the term s_i^t to the fraction $f(\cdot, \cdot)$. This modification represents the contribution of the self-information from node i to its updating process. This implies that the updated value for each node i is obtained by taking into account its own previous state value and also the previous state values of its neighbors. This tries to minimize the impact of an excessive neighboring influence. Hence, the updating process tries to achieve a trade-off between its own influence and the influence exerted by the nodes j by averaging both values.

One can see from Eqs. (3) to (10) that if a node i is surrounded by nodes with similar image difference values and similar labels, w_{ij} should be high. This implies that the s_i value should be reinforced through the Eq. (10) and the energy given by the Eq. (3) minimum and vice versa. Moreover, at high T, the value of $f(\cdot, \cdot)$ is lower for a given value of the forces l_i^t. Details about the behavior of T are given in Duda et al. [9]. We have verified that this fraction must be small as compared to s_i^t in order to avoid that the updating is controlled only by the data and regularization terms embedded in w_{ij}, i.e., through the data and contextual consistencies. Under the above considerations and based on Hajek [14], Geman and Geman [13] and Haykin [15], the following annealing schedule suffices to obtain a global minimum: $T(t) = T_0 / \log(t+1)$, with T_0 being a sufficiently high initial temperature. T_0 is computed as follows [Laarhoven and Aarts 22]: (1) we select four pairs of images, computing the energy in (3) for each pair after the network initialization; (2) we choose an initial temperature that permits about 80% of all transitions to be accepted (i.e., transitions that decrease the energy function), and this value is changed until this percentage is achieved; (3) we compute the M transitions ΔE_k and we look for a value for T for which $\frac{1}{M}\sum_{k=1}^{M} \exp(-\Delta E_k / T) = 0.8$, after rejecting the higher order terms of the Taylor expansion of the exponential, $T = 5\langle \Delta E_k \rangle$, where $\langle \cdot \rangle$ is the mean value. In our experiments, we have obtained $\langle \Delta E_k \rangle = 9.2$, giving $T_0 = 46.0$ (with a similar order of magnitude as that reported in Hajek [14]). We have also verified that a value of $t_{\max} = 100$ suffices, although the expected condition $T(t) = 0$, $t \to +\infty$ in the original algorithm is not fully fulfilled. The assertion that it suffices is based on the fact that this limit was never reached in our experiments as shown later in the Sect. 3.3, hence this value does not affect the results.

In order to avoid strong intensity variations between the two images under processing, we perform, when required, radiometric adjustment through homomorphic filtering according to the results obtained in Pajares et al. [28]. As mentioned before, each pixel in the reference image creates a node in the network. The DSA process is as follows [9]:

1. *Initialize*: load each node with $s_i^{t=0}$ as given by the equation (2); set $\alpha = 0.5$, $\gamma = 0.65$, $\varepsilon = 0.05$ (constant to accelerate the convergence); $t_{\max} = 100$. Define *nc* as the number of nodes that change their state values at each iteration.
2. *DSA process*:

```
t = 0
while $t < t_{\max}$ or nc $\neq 0$
    $t = t + 1$; nc = 0;
    for each node $i$
        update $s_i^t$ according to the Eq. (10) from Eqs.
(6) to (9)
if $|s_i^t - s_i^{t-1}|$    $\varepsilon$ then
      $nc = nc + 1$
    end if
end for
end while
```

3. *Outputs*: the states s_i for all nodes updated.

At each iteration t, the energy is computed according to the Eq. (3) which is rewritten in the Eq. (11),

Table 1 Data sets properties and description

Type of data	Nodes (frames size)	Number of pairs analyzed	Description
Outdoor	1,392 × 1,040	22	8 video sequences acquired at 15 fps during 60 consecutive seconds and captured during 15 different days, i.e., under different atmospheric conditions affecting the illumination
real video sequence		16:2 per sequence	5 of the above sequences captured 5 different scenes and 3 captured duplicated scenes, obviously with different objects under motion.
		6:2 for each duplicated scene	The camera remains static during the acquisition (i.e., all frames are registered)
Indoor	840 × 760	24	8 video sequences acquired at 15 fps during 90 consecutive seconds (intervals A: 0–45s and B: 46–90s)
real video sequence		8: only interval A	Interval A: the illumination remain unchanged
		8: only interval B	Interval B: the illumination is manually modified
		8: intervals A, B	The camera remains static during the acquisition (i.e., all frames are registered)
			If any image belongs to the interval B, then homomorphic filtering is applied
Synthetic	400 × 400	30	Given 10 pairs of remote sensing images, from each pair (I_1,I_2) we make synthetic images from I_2 by introducing manual changes (I_3), Gaussian noise (I_4) and "salt and pepper" noise (I_5). All with an amount of changes greater than the 20%.
remote sensing		10: I_1 with I_3	
		10: I_1 with I_6	
		10: I_1 with I_5	Always homomorphic filtering is applied

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\operatorname{sgn}\left(W_{ij}\right)\right]^{v+1} W_{ij} s_i^t s_j^t \tag{11}$$

where W_{ij}, sgn and v are defined in the Eq. (9) and s_i^t, s_j^t are updated according to the above procedure.

3 Validation, comparative analysis and performance evaluation

3.1 Description of the data sets

All data sets used for testing purposes have been selected taking into account that the amount of change surpasses the 20% with respect to the full difference image. They are exclusively used for the experiments carried out in this work.

As mentioned before, the DSA approach is proposed because of its better performance in image change detection when the amount of changes surpasses the 20% with regard to the full image difference. Hence the images used for testing purposes fulfill this requirement. We used real video sequences from outdoor and indoor environments. We have also prepared synthetic images from remote sensing scenes, because we have not available real remote sensing images with an amount of change greater than the 20%. The synthetic images are created by introducing changes that overpass such amount of changes through two procedures: (a) manually and (b) adding noise (Gaussian and salt and pepper). In Table 1 we can find a summary of the data sets used. In order to assess the validity of the results for the video sequences, we prepare a ground-truth for each pair of images to be analyzed as follows: (a) we have selected 30 frames of the real video sequences; (b) each frame is manipulated by introducing synthetic changes, under our control, i.e., we know the amount of changes introduced; (c) based on the analysis of thresholding methods for image change detection reported in Rosin and Ioannidis [31] we have verified that the best performance is achieved by the methods described in Kapur et al. [18] (KA) and Wu et al. [38] (WU). So, given two real images the ground truth is established by computing the intersection and the union of the change results provided by KA and WU, we choose either the intersection or the union according to our observation and refine it manually if required. The subjective observation is suggested in some approaches [30].

Figures 2, 3 and 4 show representative pairs of images belonging to the type of data described in the Table 1. Figure 2 a, b show frames I_{200} and I_{800} (the sub index indicates the number of frame in the sequence) of the same outdoor sequence; (Fig. 2c) shows the ground truth map; (Fig. 2d) displays a raw image difference between Fig. 2a and b; Fig. 2e shows the results of the network initialization and (Fig. 2e) the results obtained by the DSA process after eight iterations. The maximum degree of change is

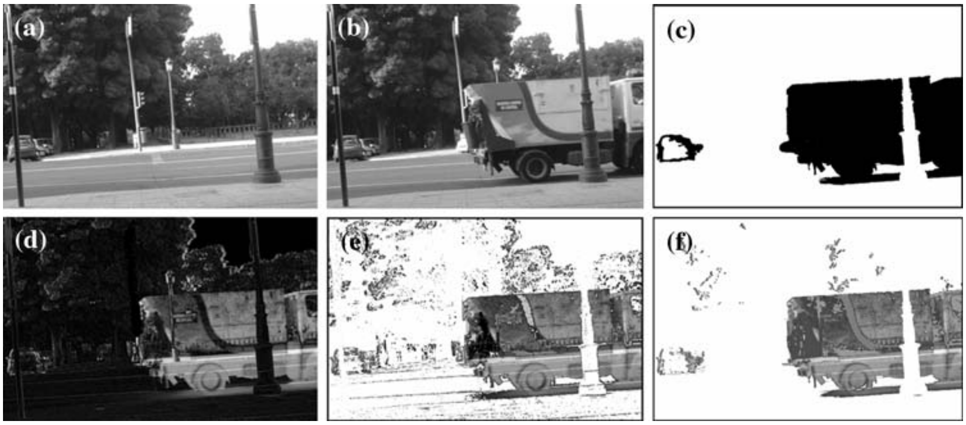

Fig. 2 Outdoor environment (**a**, **b**), two images of the same sequence (**c**), ground truth map (**d**), difference image (**e**), network initialization (**f**), changes detected with the DSA approach between (**a**) and (**b**)

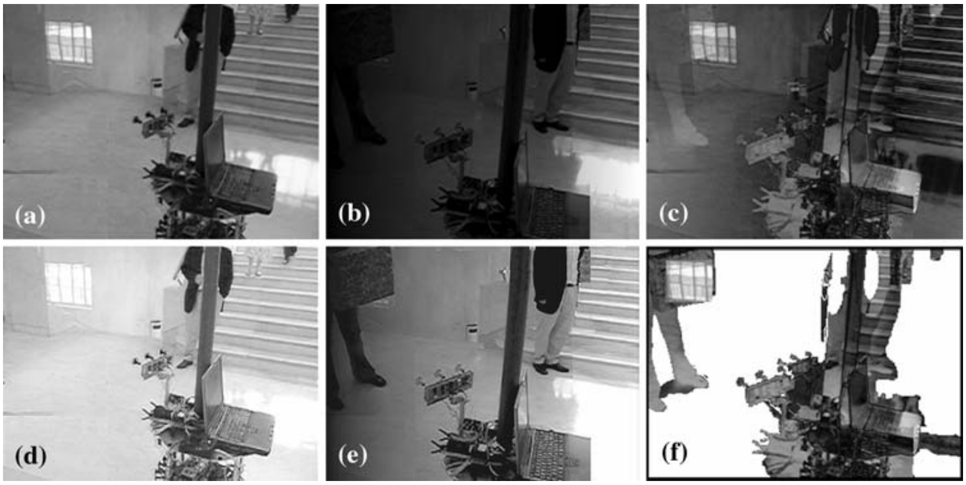

Fig. 3 Indoor environment (**a**) and (**b**) two images of the same sequence (intervals A and B, respectively); **c** difference image from **a** and **b**, **d** and **e** homomorphic filtering results for (**a**) and (**b**), respectively; **f** changes detected with the DSA approach from (**d**) and (**e**)

represented as a black pixel while unchanged areas are white. Figure 3 (a) and (b) show two frames of the same indoor scene. Each indoor video sequence is acquired at 15 frames per second during 90 consecutive seconds. During the first 45 s (interval A) the illumination remains invariable. During the remaining 45 s (interval B), we intentionally vary the illumination. This is achieved by closing blind windows and switching off artificial lights. This means that two images belonging to the same interval (A or B) will have similar illumination; on the contrary if they belong to different intervals they will have different illumination. We apply homomorphic filtering for images belonging to different intervals. The images (a) and (b) in the Fig. 3 belong to the intervals A and B, respectively. Figure 3c shows the difference image between the raw images in (a) and (b). The results of the homomorphic filtering applied to (a) and (b) are displayed in (d) and (e), respectively. Finally, Fig. 3f shows the results obtained by the DSA method for the filtered images (d) and (e). Figure 4 a and b show two images of the same urban area acquired during different days; Fig. 4c shows a synthetic image obtained from Fig. 4b by introducing changes manually surpassing the 20%; Fig. 4a shows a noisy image corrupted from Fig. 4b with zero-mean Gaussian noise of variance 5, i.e., the 26.4% has changed. The addition of noise is a common practice in image change detection [4, 5, 7, 19] in order to verify the robustness of the methods. The remote sensing images are geo-referenced by selecting twelve control points and applying a 10-parameter quadratic model, which allows mapping the pixel coordinates between both images [34]. Initially, we have the original images I_1 and I_2 and the synthetic one I_3 (made from I_2 by introducing known artificial changes). We obtain D_{23} as the difference between I_2 and I_3, i.e., D_{23} contains exactly the artificial changes. Then we register I_2 and I_3 against I_1 obtaining $I_{\mathrm{R}21}$ and I_{R31}, respectively. D_{R23} is the difference image obtained from I_{R21} and I_{R31}. The error between D_{23} and D_{R23} is the registration error, which on average is quantified as the 0.5%.

3.2 Description of the experiments

Different experiments have been carried out to assess the validity and robustness of the proposed DSA approach. The

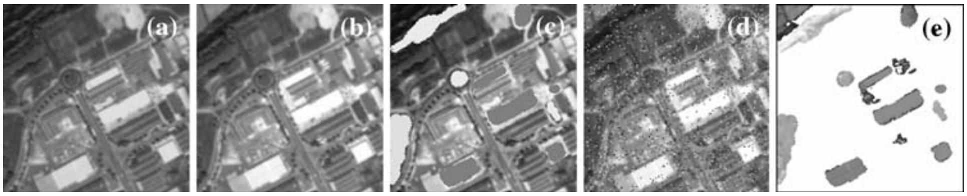

Fig. 4 **a**, **b** Urban area acquired by the IKONOS satellite at different times, **c**, **d** synthetic images obtained from **b** with artificial changes and noise with zero mean and variance 5.0; **e** changes obtained for the DSA method between the images (**a**) and (**c**)

effectiveness of our DSA method is verified against the following six strategies, described in the Sect. 1: MTD [7], Liu [24], MAP [1], SKI [32], CAR [6], BRU [5] and HNN [27]; Table 2 shows the thresholds and window sizes used for each method. The window size is the neighborhood in DSA, HNN, BRU and MAP required for mapping the contextual information.

The results obtained for each method are compared against the ground truth based on the following measures [31]: TP: True positives, i.e., number of change pixels correctly detected; FP: False positives, i.e., number of no-change pixels incorrectly labelled as change; TN: True negatives, i.e., number of no-change pixels correctly detected; FN: False negatives, i.e., number of change pixels incorrectly labelled as no-change. From these quantities the following two measures are used [31]: the percentage of correct classification computed as $PCC = (\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN})$ and the Yule coefficient like in Sneath and Sokal [33] as $|(\mathrm{TP}/(\mathrm{TP}+\mathrm{FP}))+(\mathrm{TN}/(\mathrm{TN}+\mathrm{FN}))-1|$. These measures have been chosen because of the reasons provided in Rosin and Ioannidis [31]. In such reference it is reported that the most obvious approach is to combine all four values as in the PCC. The PCC measure is broadly used in computer vision for assessing a classifier's performance, it is also reported that the PCC performs unfavourably when the amount of change is small compared to the overall image (less than 4%), our testing images have relative high ratings (greater than 20%). Hence, the PCC is acceptable; nevertheless we also use the Yule coefficient in order to overcome possible limitations of the PCC.

DSA, HNN and BRU are iterative automatic change detection methods; the thresholds specified are required during the network initialization process (Sect. 2.1). Bruzzone and Fernandez-Prieto [5] conclude that the initialization threshold for the BRU approach is a non-critical parameter because of the iterative process assumes and corrects initial possible errors. This is also applicable in our DSA strategy, i.e., the automatic condition is preserved. Table 3 describes the set of experiments carried out from the data sets shown in the Table 1. It indicates the number of pairs and the characteristics of the images selected for each pair (I_a, I_b). MTD requires a sequence of consecutive frames which is not available for the experiments E2, E6, E7 and E8, i.e., it is not tested for these experiments.

3.3 Experimental results

Before we explain the results obtained for the above experiments, we give the details that motivated the choice of the DSA. We selected the three pairs of images with the greatest number of changes from the pairs of the eight experiments described in the Table 3, i.e., 24 pairs of images where the amount of changes, on average, is about the 37%. We also selected 24 pairs of images with an amount of change less than the 20% similar to the pairs described in Pajares [27]. We applied the strategy defined in this reference for both sets of images. The set of images with changes less than the 20% is termed HNN1 and the set of images greater than the 20% is termed HNN2. In the Fig. 5 is displayed the PCC and Yule coefficient values for both sets of changed images. We can see that HNN gets worse for HNN2 where local minima have been detected. This justifies the choice of DSA as indicated in the introduction.

Now we give details about the results for the eight experiments described in the Table 3. The comparative performance of the proposed DSA approach is analyzed in terms of the correct classification, computation time and number of iterations. Table 4 shows the results in terms of the correct classification for the eight experiments. For each pair of images, we compute the PCC and Yule scores; the final result for each experiment is averaged by the

Table 2 Window size and threshold values for the change detection methods

Methods	MTD	LIU	SKI	MAP	CAR	BRU	HNN	DSA
Window size	1 × 1	3 × 3	3 × 3	3 × 3	N/A	3 × 3	3 × 3	3 × 3
Threshold	52.5	0.05	0.06	78	N/A	0.5	0.5	0.5

Table 3 Set of experiments and pairs of images selected

Experiment	Environment	Number of pairs	Characteristics of the pairs of images selected (I_a, I_b) t_{ab}: time elapsed between the acquisition of I_a and I_b
E1	Outdoor sequence	16	8 sequences, 2 pairs per sequence. Frames I_a and I_b of each pair belong to the same sequence; $t_{ab} > 20$ secs.
E2	Outdoor sequence	6	There are 3 duplicated scenes. I_a and I_b belong to different sequences of duplicated scenes. Two pairs are selected for each two duplicated sequences. I_a and I_b represent the same order number for the frame in the sequence. Homomorphic filtering is applied.
E3	Indoor sequence	8	8 sequences, 1 pair per sequence; I_a and I_b belong both to the interval A; $t_{ab} > 20$ seconds.
E4	Indoor sequence	8	8 sequences, 1 pair per sequence; I_a and I_b belong both to the interval B; $t_{ab} > 20$ s. Homomorphic filtering is applied.
E5	Indoor sequence	8	8 sequences, 1 pair per sequence; I_a and I_b belong to the intervals A and B respectively; $t_{ab} > 40$ s. Homomorphic filtering is applied.
E6	Synthetic remote sensing	10	10 pairs with manual changes. Homomorphic filtering is applied.
E7	Synthetic remote sensing	10	10 pairs with Gaussian noise of zero-mean and variance 5. Homomorphic filtering is applied.
E8	Synthetic remote sensing	10	10 pairs with salt and pepper noise of densities greater than 20%. Homomorphic filtering is applied.

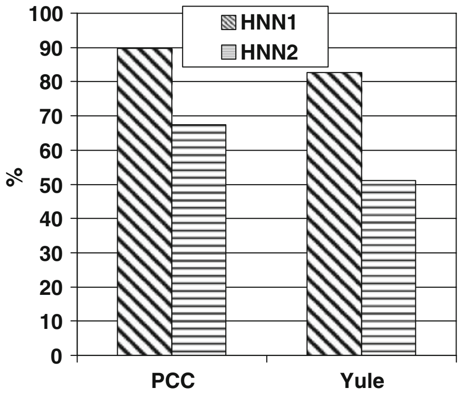

Fig. 5 Behavior of the HNN for different amounts of change: HNN1 and HNN2 less and greater than 20%, respectively

number of pairs used. Larger values indicate better performance. We have averaged the number of iterations for each experiment, resulting: E1 = 8.2, E2 = 12.3, E3 = 7.9, E4 = 11.8, E5 = 14.2, E6 = 11.8, E7 = 10.1, E8 = 10.3. So, the nearest integer is used as the number of iterations for comparing the iterative BRU and HNN approaches. The standard deviation is also displayed. The best results for each experiment are obtained by the relaxation approaches (DSA, HNN, BRU) and especially for the proposed DSA approach (in bold).

Table 5 shows the results in terms of execution time for each method. All tests have been developed in MATLAB and then compiled under MicroSoft Visual C++ 6.0 and executed on a P4 1.4 GHz with 512-MB RAM. The time depends on the size of each frame (the sizes are given in the Table 1). As already mentioned, this size determines the number of nodes in the DSA. Hence, the time is computed for this size according to the type of environment. The time for MTD is computed for the processing of two consecutive frames without the labeling and closing operations used for removing noises and shadows, as described in the original work. The results shown in Table 5 do not include the time spent during homomorphic filtering when applied. This time is proportional to the frame size; it has been quantified as 42.20 ms for remote sensing (400 × 400), 151.30 ms for indoor frames (840 × 760) and 381.8 ms for outdoor frames (1,392 × 1,040). One can see that the computational load of BRU, HNN and DSA, which apply a relaxation procedure, is greater than the cost required for the remaining methods.

The time for MTD, LIU, SKI, CAR and MAP only depends on the size of the images processed. In BRU, HNN and DSA depends on both, the number of iterations and the amount of changes, because each change is stored at each iteration to be recovered during the next one.

Figures 6, 7 and 8 show the behavior of the averaged PCC coefficient for simulated annealing (DSA), Hopfield neural network (HNN) and Bayes decision theory (BRU) for each experiment against the number of iterations. For each experiment this number is computed by averaging the number of iterations of the different pairs and rounding the result to the nearest integer. A similar tendency is shown by the Yule coefficient. Hence, for simplicity, we have omitted its graphical behavior. As one can see, DSA and HNN start its convergence process with lower values than

Table 4 Averaged (*Av*) PCC and Yule scores and standard deviation (σ) for each method against the set of experiments

$\times 10^{-3}$		E1		E2		E3		E4		E5		E6		E7		E8	
		Av	σ	Av	σ	Av	σ	Av	σ	Av	σ	Av	σ	Av	σ	Av	σ
MTD	PCC	563	23.1	Not tested		651	14.5	224	8.8	247	13.2	Not tested		Not tested		Not tested	
	Yule	311	14.2			446	11.9	114	7.9	122	9.4						
CAR	PCC	448	12.4	366	12.1	444	12.8	231	8.3	274	7.9	353	12.4	221	11.2	212	13.2
	Yule	287	11.7	243	11.9	253	10.1	105	8.4	107	7.8	147	11.6	112	10.7	113	11.8
LIU	PCC	601	20.3	578	9.8	764	16.0	412	10.2	447	10.2	566	13.2	450	9.2	433	12.6
	Yule	334	10.8	282	10.2	303	9.9	144	11.1	175	8.9	227	12.1	201	9.4	181	12.9
SKI	PCC	663	13.9	510	10.2	773	14.2	455	12.3	481	10.1	447	10.2	442	11.5	423	9.8
	Yule	401	10.4	366	9.3	375	8.9	153	9.9	164	9.9	222	10.3	187	11.9	166	9.9
MAP	PCC	871	18.6	680	15.0	889	12.2	522	10.3	557	12.3	812	12.8	606	12.8	671	11.7
	Yule	600	19.6	435	14.8	540	13.2	201	12.2	232	9.8	567	10.3	344	12.7	398	11.2
BRU	PCC	918	14.5	813	13.0	954	13.4	692	12.0	704	10.4	832	13.2	689	11.2	828	12.8
	Yule	683	12.1	455	10.2	708	11.8	245	11.2	373	12.3	549	12.3	430	10.3	396	11.9
HNN	PCC	931	11.9	851	11.7	957	12.2	714	11.1	751	12.3	860	13.4	710	12.4	848	12.4
	Yule	667	12.8	549	10.3	710	11.3	278	13.2	451	11.8	573	9.9	444	10.8	413	10.6
DSA	PCC	**945**	12.5	**867**	11.2	**969**	12.4	**738**	10.9	**766**	10.7	**881**	10.1	**751**	11.5	**852**	12.1
	Yule	**710**	11.3	**581**	10.3	**742**	12.6	**346**	12.9	**498**	12.1	**598**	11.2	**469**	12.1	**434**	10.9

Table 5 Comparative results in terms of execution time

Unit: ms	MTD	LIU	SKI	CAR	MAP	BRU	HNN	DSA
Outdoor environment	44.8	186.7	100.9	199.3	225.9	3464.1	3634.9	3492.8
Indoor environment	1.1	79.1	39.8	79.5	99.0	1413.6	1643.0	1576.3
Remote sensing	4.4	26.8	15.6	32.2	46.3	612.7	748.9	701.3

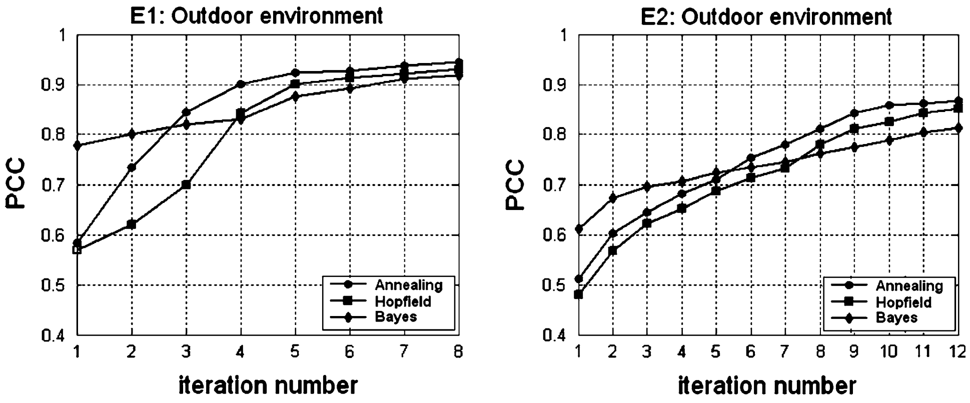

Fig. 6 Averaged PCC values for DSA (annealing), HNN (Hopfield) and BRU (Bayes) methods against the number of iterations: experiments E1 and E2

BRU and then, after a small number of iterations, overpass the values of BRU. This means that the initialization process in BRU is better than the initialization in DSA and HNN. On the contrary, they converge faster than BRU. We have verified that BRU requires a greater number of iterations than HNN or DSA to achieve similar performances. DSA performs better than HNN with a similar tendency.

In the Fig. 2f we can see how an important number of false changed and unchanged areas have been removed as compared with the initialization image in the Fig. 2e. This is also applicable for the remainder sets of images tested.

3.4 Analysis of the results

1. The best performance for the set of images tested, in terms of correct classifications, is achieved by the proposed DSA approach. Because of DSA and HNN

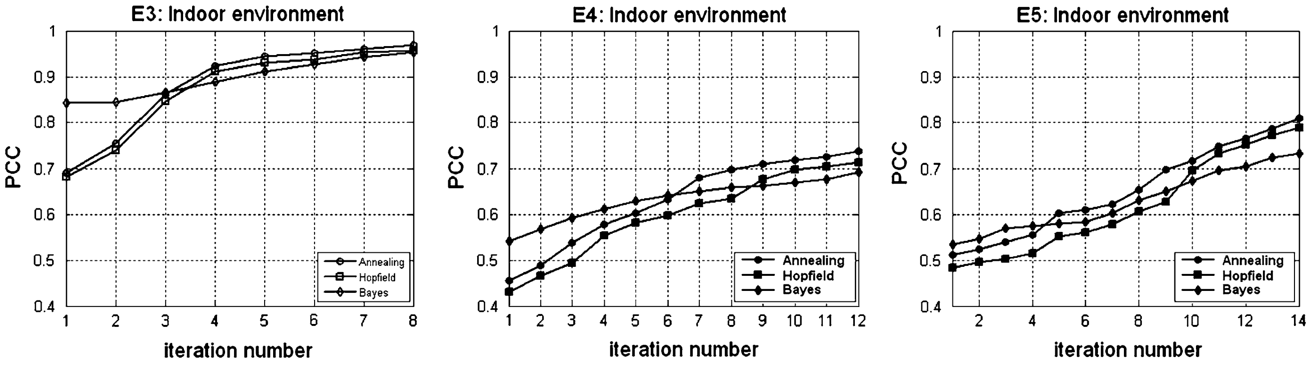

Fig. 7 Averaged PCC values for DSA (annealing), HNN (Hopfield) and BRU (Bayes) methods against the number of iterations: experiments E3, E4 and E5

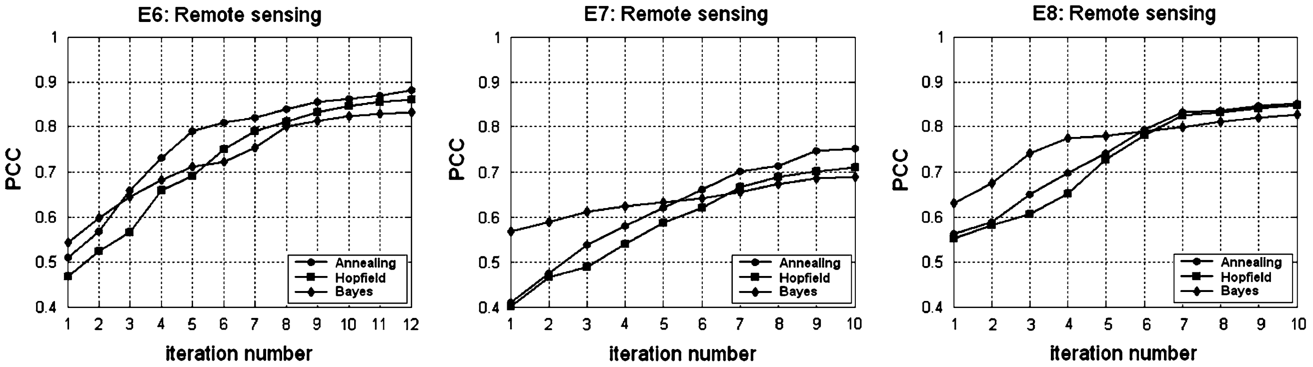

Fig. 8 Averaged PCC values for DSA (annealing), HNN (Hopfield) and BRU (Bayes) methods against the number of iterations: experiments E6, E7 and E8

map both contextual and self information under a similar scheme and apply an energy-based optimization process this improvement is achieved thanks to the simulated annealing scheduling involved in DSA.

2. The iterative approaches perform better than non iterative in terms of correct classifications and worst in terms of computational cost. The iterative have also the capability for correcting initial misclassifications.
3. The iterative approaches considering both contextual and self information achieve better results than those using only the contextual one.
4. The DSA performs well for images with significant changes in the illumination (E2, E4, E5) and also for noisy images (E5, E6 and E7).

4 Conclusions

In this paper we have developed a new automatic strategy for image change detection based on the well-founded DSA paradigm. The mapping of the contextual information is a suitable mechanism for image change detection methods. The Simulated Annealing framework under its Deterministic version is able to avoid the drawback of the Hopfield neural network related to the local minima during the energy minimization process. The performance of the DSA against the stochastic simulated annealing version is also exploited. The proposed DSA method has proven to be robust against noise. Its accuracy performance against other existing strategies has been compared favorably. The main drawback of the DSA method comes from its high execution time. So, for real-time requirements under surveillance tasks it should be implemented under parallel architectures.

Acknowledgments Part of the work has been performed under project no. 143/2004 Fundación General UCM. The authors are also grateful to the referees for their constructive criticism and suggestions on the original version of this paper.

References

1. Aach T, Kaup A (1995) Bayesian algorithms for adaptive change detection in image sequences using Markov Random fields. Signal Process Image Commun 7:147–160
2. Bosc M, Heitz F, Armspach JP, Namer I, Gounot D, Rumbach L (2003) Automatic change detection in multimodal serial MRI: application to multiple sclerosis lesion evolution. Neuroimage 20:643–656

3. Bruzzone L, Fernández-Prieto D (2000a) An adaptive parcel-based technique for unsupervised change detection. Int J Remote Sensing 21(4):817–822
4. Bruzzone L, Fernández-Prieto D (2000b) Automatic analysis of the difference Image for unsupervised change detection. IEEE Trans. Geosci Remote Sensing 38(3):1171–1182
5. Bruzzone L, Fernández-Prieto D (2002) An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. IEEE Trans Image Process 11(4):452–466
6. Carlotto MJ (2005) A cluster-based approach for detecting man-made objects and changes in imagery. IEEE Trans Geosci Remote Sensing 43(2):374–387
7. Chang CC, Chia TL, Yang CK (2005) Modified temporal difference method for change detection. Opt Eng 44(2):1–10
8. Desurmont M, Bastide M, Chaudy C, Parisot D, Delaigle JF, Macq B (2005) Image analysis architectures and techniques for intelligent surveillance systems. IEE Proc Vis Image Signal Process 152(2):224–231
9. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York
10. Dunmur AP, Titterington DM (1998) Mean fields and two-dimensional Markov random fields in image analysis. Pattern Anal Appl 1(4):248–260
11. Durucam E, Ebrahimi T (2001) Change detection and background extraction by linear algebra. Proc IEEE 89(10):1368–1381
12. Fang CY, Cheng SW, Fuh CS (2003) Automatic change detection of driving environments in a vision-based driver assistance system. IEEE Trans Neural Netw 14(3):646–657
13. Geman S, Geman G (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6:721–741
14. Hajek B (1988) Cooling schedules for optimal annealing. Math Oper Res 13:311–329
15. Haykin S (1994) Neural networks: a comprehensive foundation. Macmillan College Publishing Co, New York
16. Hsu YZ, Nagel HH, Reckers G (1984) New likelihood test methods for change detection in image sequences. Comput Vis Graph Image Process 26:73–106
17. Jain Z and Chau Y (1995) Optimum multisensor data fusion for image change detection. IEEE Trans Syst Man Cybern 25(9):1340–1347
18. Kapur J, Sahoo P, Wong A (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285
19. Kasetkasem T, Varshney PK (2002) An image change detection algorithm based on Markov random field models. IEEE Trans Geosci Remote Sensing 40(8):1815–1823
20. Kirkpatrick S (1984) Optimization by simulated annealing: quantitative studies. J Stat Phys 34:975–984
21. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
22. Laarhoven van PMJ, Aarts EHL (1989) Simulated annealing: theory and applications. Kluwer, Holland
23. Liu Q. (2005) New change detection models for object based encoding of patient monitoring video. PhD Thesis, School of Engineering. University of Pitsburg
24. Liu SC, Fu CV, Chang S (1998) Statistical change detection with moments under time-varying illumination. IEEE Trans Image Process 7(9):1258–1268
25. Liu Q, Sclabassi RJ, Li CC, Sun M (2005) An application of MAP-MRF to change detection in image sequence based on mean field theory. EURASIP J Appl Signal Process 2005:1956–1968
26. Lu T, Suganthan PN (2004) An accumulation algorithm for video shot boundary detection. Multimedia Tools and Appl 22:89–106
27. Pajares G (2006) A Hopfield neural network for image change detection. IEEE Trans Neural Netw (in press)
28. Pajares G, Ruz JJ, Cruz JM (2005) Performance analysis of homomorphic systems for image change detection. In: Marques JS, Pérez de la Blanca N, Pina P (eds) Pattern recognition and image analysis. Lecture Notes in Computer Science, vol 3522. Springer, Berlin, pp 563–570
29. Paragios N, Deriche R (2000) Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Trans Pattern Anal Mach Intell 22(3):266–280
30. Radke RJ, Andra S, Al-Kofahi O, Roysam B (2005) Image change detection algorithms: a systematic survey. IEEE Trans Image Process 14(3):294–307
31. Rosin PL, Ioannidis E (2003) Evaluation of global image thresholding for change detection. Pattern Recogn Lett 24: 2345–2356
32. Skifstad K, Jain R (1989) Illumination independent change detection from real world images sequences. Comput Vis Graph Image Process 46(9):387–399
33. Sneath P, Sokal R (1973) Numerical taxonomy: the principle and practice of numerical classification. W.H. Freeman, San Francisco
34. Starck JL, Murtagh F, Bijaoui A (2000) Image processing and data analysis: the multiscale approach. Cambridge University Press, Cambridge
35. Stauffer C, Grimson WEL (2000) Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell 22(8):747–757
36. Stringa E, Regazzoni CS (2000) Real-time video shot detection for scene surveillance applications. IEEE Trans Image Process 9:69–79
37. Valera M, Velastin SA (2005) Intelligent distributed surveillance systems: a review. IEE Proc Vis Image Signal Process 152(2):192–204
38. Wu QZ, Cheng HY, Jeng BS (2005) Motion detection via change-point detection for cumulative histograms of ratio images. Pattern Recogn Lett 26:555–563

Author Biographies

G. Pajares Received M.Sc and Ph.D. degrees in Physics from UNED (distance University from Spain) (1987, 1995) discussing a thesis on the application of pattern recognition techniques to stereovision. He was working in Indra Space and INTA developing remote sensing applications. He joined the Complutense University in 1995 as an associated professor and from 2004 as a professor at full time on the Faculty of Informatics in the Department of Software Engineering and Artificial Intelligence. The areas covered are: Computer vision, Artificial Intelligence. His current research interests include machine visual perception, pattern recognition and neural networks.

José. J. Ruz received M.Sc degree in Physics from the Complutense University of Madrid in 1974, and the Ph.D degree in computer science in 1980 from the same University. He joined the Department of Computer Architecture and Automatic Control of the Complutense University in 1981 where he is a Professor. His current research interest includes parallel architectures for distributed optimization, high performance computing for pattern recognition and optimal path planning for unmanned aerial vehicles.

J. M. de la Cruz received M.Sc degree in Physics and Ph.D. from the Complutense University in 19790 and 1984, respectively. From 1985 to 1990 he was with the Department of Automatic Control, UNED (Distance University of Spain), and from October 1990 to 1992 with the Department of Electronic, University of Santander. In October 1992, he joined the Department of Computer Science and Automatic Control of the Complutense University where he is a Professor. His current research interest includes robotics vision systems, fusion sensors and applications of automatic control to robotics and flight control.