



DETERMINISTIC SIMULATED ANNEALING APPLIED TO COMPUTER VISION TASKS

GONZALO PAJARES^{1*}, J. MIGUEL GUERRERO¹, JUAN ROMEO¹, DAVID SÁNCHEZ-BENÍTEZ¹, MARTÍN MONTALVO², P. JAVIER HERRERA², MARÍA GUIJARRO¹, JESÚS M. DE LA CRUZ² AND JOSÉ J. RUZ²

¹Department of Software Engineering and Artificial Intelligence, Universidad Complutense of Madrid, Spain

²Department of Computer Architecture and Automatic, Universidad Complutense of Madrid, Spain

*Corresponding Author: Email- pajares@fdi.ucm.es

Received: October 15, 2011; Accepted: February 21, 2012

Abstract- This paper makes a review about the application of the simulated annealing optimization approach for solving three different computer vision problems. They are image change detection, stereovision matching and texture image classification. The simulated annealing mechanism evaluates an entity in the image and considers the influence exerted by neighbor's entities about the first one. Based on this influence some feature of the first entity is conveniently modified. This allows to make better decisions based on such modification.

Key words- Simulated annealing, optimization, change detection, stereovision, texture classification, POLSAR image classification.

Citation: Gonzalo Pajares, et al (2012) Deterministic Simulated Annealing Applied to Computer Vision Tasks. Journal of High Performance Computing, ISSN: 2230-7192 & E-ISSN: 2230-7206, Volume 2, Issue 1, 2012, pp-20-23.

Copyright: Copyright©2012 Gonzalo Pajares, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Introduction

Computer vision is an emerging area which is demanding solutions for solving different problems. The basic data are bi-dimensional (2D) images captured from the three-dimensional (3D) real world.

A look to the 3D world allows observing objects in the scene which are formed by structures or elements grouped together. Fortunately when the objects are mapped from the 3D world to the 2D image they preserve this grouping.

From this observation, an important issue addressed in computer vision tasks based on image applications is referred to how sensory elements perceive the objects in a scene, i.e. how the scene analysis problem is addressed. To deal with real-world scenes some criterion for grouping elements in the scene is required. In the work of Wang [1] a list of major grouping principles is exhaustively studied. They are inspired in the Gestalt's principles [2]. The most important from the point of view of computer vision tasks are: proximity, labeled features that lie close in space tend to group; similarity, labeled features with similar values tend to group; connectedness, labeled features that lie inside the same connected region tend to group.

These principles allow defining a spatial neighborhood in the images from the 3D scene. Now the problem is to build some structure where the above principles are mapped. Several approaches can be used; one of them is the Deterministic Simulated Annealing (DSA) [3]. DSA is an optimization approach, which minimizes an energy function. The main contribution of DSA is its ability to avoid local minima during the optimization process thanks to the annealing scheme.

Because the spatial features are preserved in the 2D image, the first requirement is to define spatial relations between entities in the image and then the influences exerted by some entities over the others. This allows to define the parameters required by the DSA. Once spatial relations and influences are defined, the DSA can be applied. This paper describes how DSA can be successfully applied to three relevant tasks in computer vision: image change detection, stereovision matching and texture classification. The paper is organized as follows. Section 2 defines the spatial relations between entities and describes the DSA optimization process and. Sections 3 to 5 describe the application of the DSA to the three above mentioned computer vision tasks. Finally, in section 6 some conclusions are introduced.

Deterministic Simulated Annealing

Firstly we need to define the spatial configuration of entities interacting between them. This will allow configuring this spatiality as a network of nodes under a specific topology. Secondly, we define the DSA optimization process.

Network topology

The starting point is a 2D image. We focus the three applications at the image pixel-level and associate a pixel with a node. So, can build a network of q nodes where each pixel i located at the spatial image coordinates (x,y) is identified as the node ni. Each node has associated a state value si. Through the DSA these network states are reinforced or punished iteratively based on the influences exerted by their neighbor nodes. As the iterations progress these estates change their values trying to achieve the maximum network stability as possible. The main goal is to make better decisions based on this stability.

DSA Optimization process

Suppose the network with the q nodes. The simulated annealing optimization problem is: modify the analogue values si so as to minimize the energy [3, 4],

$$E(t) = -\frac{1}{2} \sum_{i=1}^q \sum_{k=1}^q r_{ik}(t) s_i(t) s_k(t) \tag{1}$$

where rik(t) is the symmetric weight interconnecting two nodes, i and k, in the network at the iteration t and can be positive or negative ranging in [-1,+1]; sk(t) is the state of the neighbouring node k. Each rik(t) determines the influence that the node k exerts on i trying to modify the state si(t). According to [3] the self-feedback weights must be null (i.e. rii = 0). The DSA approach tries to achieve the most network stable configuration based on the energy minimization.

The term rik(t) is a consistency coefficient which computes the degree of coincidence between the states of the nodes in a given neighbourhood, defined as the m-connected spatial region, , where m is set normally to 8 and allows the implementation of the proximity and connectedness Gestalt's principles [1,2]. This coefficient is computed at the iteration t as follows,

$$r_{ik}(t) = \begin{cases} 1 - |s_i(t) - s_k(t)| & k \in N_i^m, i \neq k \\ 0 & k \notin N_i^m, i = k \end{cases} \tag{2}$$

From (2) we can see that rik(t) ranges in [-1,+1]. The influence exerted by the node k over the node i will be positive (reward) or negative (punishment). Hence, a positive data consistency will contribute towards the network stability. Table 1 shows the behavior of the energy, equation (1), against consistencies and state values. As one can see, the energy decreases as the data and the state values are both simultaneously consistent (rows 1 and 4 in the left part of Table 1); otherwise under any inconsistency the energy increases.

Table 1- Behaviour of the energy term against the consistency coefficient and state values

r _{ik} (t)	s _i (t)	s _k (t)	E(t)	r _{ik} (t)	s _i (t)	s _k (t)	E(t)
+	+	+	-	-	+	+	+
+	+	-	+	-	+	-	+
+	-	+	+	-	-	+	+
+	-	-	-	-	-	-	+

The simulated annealing process was originally developed in [5,6] under a stochastic approach. In this paper we have implemented the deterministic one described in [3,7], because, as reported in [3], the stochastic is slow due to its discrete nature as compared to the analogue nature of the deterministic. Following the notation in

[3], let $u_i(t) = \sum_k r_{ik}(t) s_k(t)$ be the force exerted on node i by

the other nodes $k \in N_i^m$ at the iteration t; then the new state si(t+1) is obtained by adding the fraction f(.,.) to the previous one according to equation (3)

$$s_i(t+1) = \frac{1}{2} [f(u_i(t), T(t)) + s_i(t)] \\ = \frac{1}{2} [\tanh(u_i(t)/T(t)) + s_i(t)] \tag{3}$$

where, as always, t represents the iteration index. The fraction f(.,.) depends upon u(t) and the temperature T, at the iteration t.

The equation (3) differs from the updating process in [3] because we have added the term si(t) to the fraction. This modification represents the contribution of the self-support 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. The introduction of the self-support tries to minimize the impact of an excessive neighboring influence. Hence, the updating process establishes a trade-off between its own influence and the influence exerted by the nodes k by averaging both values.

One can see from equation (2) that if a node i is surrounded by nodes with similar state values, rik(t) should be high. This implies that the si(t) value should be reinforced through the equation (3) and the energy given by the equation (1) is minimum and vice

versa. Moreover, at high T, the value of f(.,.) is lower for a given value of the forces ui(t). Details about the behavior of T are

given in [3]. We have verified that the fraction $u_i(t)/T(t)$ must be small as compared to si(t) in order to avoid that the updating is controlled only by ui(t). Under the above considerations and based on [5,6,8], the following annealing schedule suffices to obtain a

global minimum: $T(t) = T_0 / \log(t+1)$, with T0 being a sufficiently high initial temperature. T0 is computed as follows [9]: a) we select four images in change detection and image classification or four pairs of stereo images in stereovision matching containing the nodes to be updated and compute the energy in equation (1); b) we choose an initial temperature that permits about 80% of all transitions to be accepted (i.e. transitions that decrease the energy function), and the temperature value is changed until this percentage is achieved; c) 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, $T = 8 \langle \Delta E_k \rangle$, where $\langle \cdot \rangle$ is the mean value. We

have also verified that a value of $t_{max} = 200$ suffices, although the expected condition $T(t) = 0, t \rightarrow +\infty$ 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 all our experiments.

The DSA process is synthesized as follows [3]:

1. Initialization: load each node with $s_i(t=0)$; set $\varepsilon = 0.01$ (constant to accelerate the convergence); $t_{max} = 200$. 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 equation (3)
 if $|s_i(t) - s_i(t-1)| > \varepsilon$ then
 $nc = nc + 1$; else $nc = nc$
 end if
 end for
 end while
3. Outputs: the states for all nodes updated.

DSA in Image Change Detection

Given a set of images of the same scene taken at several different times, the goal is to identify the set of pixels that are significantly different between the last image (I_L) and a previous reference image (I_R); changed pixels may result from a combination of underlying factors, including appearance or disappearance of objects, motion of objects relative to the background, shape change of objects or environment modifications (buildings, fires, etc.) [10]. To identify the pixels which have changed a difference image, $D = |I_L - I_R|$, is computed between both involved images. The values in D are mapped to range in $[-1, +1]$ and represent changed/unchanged pixels. Indeed, a value of -1 indicates the pixel is unchanged and a value of $+1$ means maximum change. Thus, each pixel location i in D determines the node n_i in the network which is built for such purpose. The state value for each node is exactly the corresponding normalized value in D . Once the network is built and the nodes loaded with their state values, the DSA process determines consistencies among all nodes in the network and establishes pixels which have changed or not. The full procedure is described in [11]. Figure 1 (a)-(b) displays two images from the same sequence, where (a) is the reference one; (c) is the difference image D , i.e. the network initialization and (d) displays only consistent changes according to the DSA procedure.

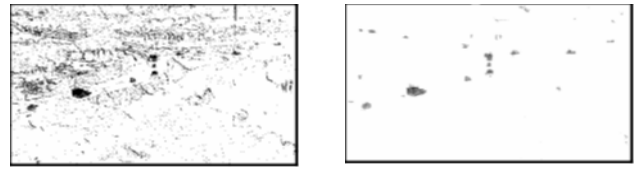


Fig. 1- (a) and (b) two images of the same video sequence; (c) network initialization; (d) changes detected with the DSA approach

DSA in Stereovision Matching

The main problem in stereovision is the image matching, that is, the process of identifying the corresponding points in two images (left and right) that are cast by the same physical point in 3-D space. This can be carried out by applying a similarity criterion between pixels which are to be matched. Two pixels are identified as matched, following a horizontal direction, between both images if they display identical intensity values. For each pixel in the left image its corresponding pixel in the right one is searched based on the similarity criterion and the difference between the horizontal x positions is computed. These differences are known as disparities and they are mapped to range in the interval $[-1, +1]$. Once we have computed all disparities, a network of nodes is built, where each node n_i represents a pixel on the left image. The state values of these nodes are the normalized disparity ones. Once the disparity values are obtained by similarity, the DSA process is triggered to achieve stable disparities. Two approaches, based on DSA, have been applied in [12] with edge segments as features and also in [13] in stereo images obtained with fish-eye optical lenses.

Figure 2 (a)-(b) displays a pair of stereo images (courtesy of Carnegie-Mellon University); (c) represents the disparity values, i.e. the network initialization and (d) displays stable disparities according to the DSA procedure.

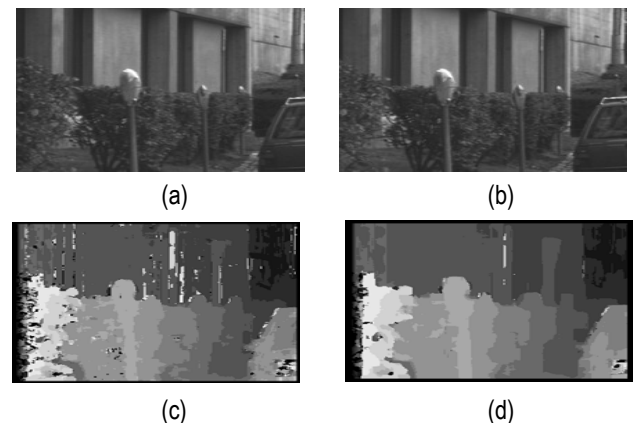


Fig. 2- (a)-(b) stereo image pair courtesy of Carnegie-Mellon University; (c) disparity values obtained by similarity; (d) corrected disparity values through the DSA approach

DSA in Texture Classification

Given an original image and a set of c possible classes w_j , ($j = 1, \dots, c$) the goal is to determine the class where a pixel belongs to. Different strategies have been proposed in the literature, including supervised ones [14]. The probabilistic parametric Bayesian (BP) which assigns pixels to a class based on the maximum probability

values. All probability values range in $[0, +1]$ and they are linearly mapped to range in $[-1,+1]$.

Because the number of classes is known, we can build a network of nodes net_j for each class w_j , where each node i in the net_j is identified as a pixel location i in the image which is to be classified. The normalized probability values are identified as the initial states at each network net_j which are to be updated during the DSA process. Under this approach we have c networks and the DSA process is applied to each network. The energy function defined in equation (1) is extended to cover the total number of classes, i.e. networks, as follows,

$$E(t) = -\frac{1}{2} \sum_{j=1}^c \sum_{i=1}^q \sum_{k=1}^q r_{ik}^j(t) s_i^j(t) s_k^j(t) \quad (4)$$

where the state values are initially the probabilities provided by the BP classifier referred to each network net_j and the consistency coefficient $r_{ik}(t)$ is also related to each network net_j . This procedure is exhaustively explained with some extensions including fuzzy clustering in [14]. Figure 3 displays an original image (a) and the classes assigned to each pixel in (b) after applying the DSA procedure according to labels in (c).

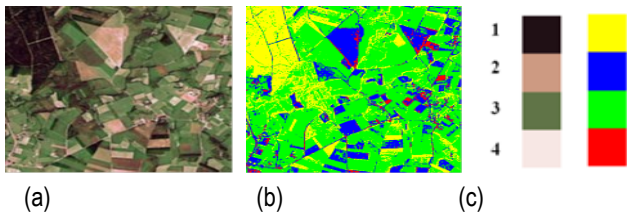


Fig. 3- (a) original image; (b) labeled image; (c) correspondence between labels and classes

Conclusions

In this paper, we propose a framework based on simulated annealing which is applied to three relevant Computer Vision tasks. Because 3D spatial relations are preserved in 2D images, we can map them by computing some parameters which allows determining mutual influences. Some properties of the entities are modified based on these influences. Better decisions are possible thanks to this mapping.

This idea can be extended to other computing vision tasks involving entities that can be mutually influenced. The unique requirement is to identify these relations and its mapping.

Acknowledgement

This paper has been funded under project DPI2009-14552-C02-01 from the Ministerio de Educación y Ciencia of Spain within the Plan Nacional of I+D+i.

References

- [1] Wang D. (2005) *IEEE Transactions on Neural Networks*, 16 (6), 1401-1426.
- [2] Koffka K. (1935) *Principles of Gestalt Psychology*. Harcourt, New York, NY, USA.
- [3] Duda R.O., Hart P.E. and Stork D.S. (2000) *Pattern Classification*.
- [4] Haykin S. (1994) *Neural Networks: A comprehensive Foundation*.

- [5] Kirkpatrick S., Gelatt C.D. and Vecchi M.P. (1983) *Science*, 220, 671-680.
- [6] Kirkpatrick S. (1984) *J. Statist. Phys.*, 34, 975-984.
- [7] Hajek B. (1988) *Math. Oper. Res.*, 13, 311-329.
- [8] Geman S. and Geman G. (1984) *IEEE Trans. Patt. Anal. Mach. Int.*, 6, 721-741.
- [9] Laarhoven P.M.J. and Aarts E.H.L. (1989) *Simulated Annealing: Theory and Applications* Kluwer Academic: Norwell, MA, USA.
- [10] Radke R.J., Andra S., Al-Kofahi O. and Roysam B. (2005) *IEEE Transactions on Image Processing*, 14(3), 294-307.
- [11] Pajares G., Ruz J.J. and Cruz J.M. (2009) *Pattern Analysis and Applications*, 12, 137-150.
- [12] Pajares G. and Cruz J.M., *IEEE Trans. Systems, Man, Cybernetics, Part B: Cybernetics*, 34(4), 1646-1657.
- [13] Herrera P.J., Pajares G., Guijarro M., Ruz J.J. and Cruz J.M. (2011) *Expert systems with Applications*, 38, 8622-8631
- [14] Guijarro M., Pajares G. and Herrera P.J. (2009) *Sensors*, 9, 7132-7149.