Neural Networks 77 (2016) 1–6

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Image and geometry processing with Oriented and Scalable Map

Hao Hua *

Key Laboratory of Urban and Architectural Heritage Conservation (Southeast University), Ministry of Education, China
School of Architecture, Southeast University, 2 Sipailou, Nanjing 210096, China

ARTICLE INFO

Article history:
Received 16 June 2015
Received in revised form 1 December 2015
Accepted 20 January 2016
Available online 3 February 2016

Keywords:
Self-organizing map
Orientation
Scale
Computer graphics

ABSTRACT

We turn the Self-organizing Map (SOM) into an Oriented and Scalable Map (OS-Map) by generalizing the neighborhood function and the winner selection. The homogeneous Gaussian neighborhood function is replaced with the matrix exponential. Thus we can specify the orientation either in the map space or in the data space. Moreover, we associate the map's global scale with the locality of winner selection. Our model is suited for a number of graphical applications such as texture/image synthesis, surface parameterization, and solid texture synthesis. OS-Map is more generic and versatile than the task-specific algorithms for these applications. Our work reveals the overlooked strength of SOMs in processing images and geometries.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Generalizing SOM

The Oriented and Scalable Map (OS-Map) is motivated by a number of graphical applications that create a connection between a topological structure and a data space. For instance, solid texture synthesis maps an input image (pixels in a 2D space) to voxels (3D topology). There are intrinsic similarities between such applications and the Self-organizing Map (SOM). A closer look suggests that the orientation and scaling of the map are critical. Thus we turn the original SOM (Kohonen, 1982, 1998, 2013) into an oriented and scalable map by modifying the neighborhood function and the winner selection. As a result, the SOM becomes a special case of our generalized model. Our model highlights the SOM's overlooked potential in image and geometry processing. Although SOM usually maps high-dimensional data to a low-dimensional space, our OS-Map indicates that mapping low-dimensional data to a high-dimensional space is also promising especially in graphical applications.

The scale in OS-Map serves as a global parameter, which specifies how many times each input item should be presented in the resulting map. In traditional topographic maps, the scale is implicitly fixed to one. The orientation – more precisely, the orientation of the gradient of model vectors across the map – is a local feature. Our experiments indicate that local orientations have a significant impact on the global arrangement of the map, which is consistent with the principle of self-organization.

OS-Map inherits both the advantages and the drawbacks of SOM. The regular grid in SOM greatly facilitates the indexing of nodes, the learning process, and the visualization of results. The payoff is that the fixed topology is sometimes inadequate for learning other topologies. People have been continually improving both the structure (e.g., Fritzke, 1995, and Rauber, Merkl, & Dittenbach, 2002) and the learning algorithm (e.g., Bishop, Svensén, & Williams, 1998 and Heskes, 2001) of topographic maps. Especially, Piastra's (2013) Self-organizing adaptive map excels at learning surfaces from point samples, employing a growing-adapting process. However, we find it is problematic to integrate the notions of orientation and scale into the later models. And we commence with the original setting of SOM because of its simple form and illustrative strength.

1.2. Related applications in computer graphics

Our generalized model can perform texture synthesis, surface quadrangulation, and solid texture construction. Despite the enormous research focusing on these disparate subjects in the field of computer graphics, the similarities between them have not been fully exploited. This lack of attention to their related qualities is partly attributed to the greater interest in the performance of specific algorithms than in the generality of models. Our approach based on SOM asserts that the various applications share a common nature that can be formulated by a single model.

* Correspondence to: School of Architecture, Southeast University, 2 Sipailou, Nanjing 210096, China.
E-mail address: whitegreen@163.com.

http://dx.doi.org/10.1016/j.neunet.2016.01.009
0893-6080/© 2016 Elsevier Ltd. All rights reserved.

Example-based texture synthesis (Wei, Lefebvre, Kwatra, & Turk, 2009) creates new textures that resemble the texture of interest. There have been patch-based methods (Kwatra, Schödl, Essa, Turk, & Bobick, 2003; Liang, Liu, Xu, Guo, & Shum, 2001) and pixel-base approaches (Efros & Leung, 1999; Kwatra, Essa, Bobick, & Kwatra, 2005). In principle, texture synthesis creates a mapping from the pixels in the input image to the pixels in the synthesized image. The OS-Map can be easily configured as a synthesized image: each node serves as a pixel and its model vector refers to a specific position in the input image. Texture synthesis has been extended to solid texture synthesis (Dong, Lefebvre, Tong, & Drettakis, 2008; Kopf et al., 2007) that builds a mapping from pixels to a 3D array of voxels. We perform solid texture synthesis simply by setting the OS-Map three-dimensional.

Surface parameterization (Hormann, Lévy, & Sheffer, 2007; Ray, Li, Lévy, Sheffer, & Alliez, 2006; Sheffer, Praun, & Rose, 2006), surface quadrilateral remeshing (Dong, Bremer, Garland, Pascucci, & Hart, 2006; Peng, Barton, Jiang, & Wonka, 2014), and direction field (Crane, Desbrun, & Schröder, 2010; Knöppel, Crane, Pinkall, & Schröder, 2013) establish a relationship between a surface in 3D Euclidean space and a cross field (or a regular topological grid). Topographic maps can fulfill the task by learning the points sampled from the given surface with proper orientation control. An epistemological difference in our approach is that it anchors the 3D surface to a regular grid, while most approaches from computer graphics fit a cross field (or a grid) onto a 3D surface.

2. Oriented and scalable map

For consistency, the notations in this paper resemble Kohonen's (2013). OS-Map inherits the stepwise recursive algorithm from SOM:

---

t: current number of iteration.
φ: negative constant[1].
N: number of nodes of the map.
d: dimension of the data space.
$\mathbf{A}$: $d \times d$ orientation matrix.
$h_{ci}(t)$: $d \times d$ neighborhood matrix.
Initialize every model vector $\mathbf{m}_i$, $i \in [0, N]$, $\mathbf{m}_i \in \mathbb{R}^d$.
Initialize learning rate $\alpha(t)$ and neighborhood radius $\sigma(t)$.
FOR T iterations

1. retrieve an input item $\mathbf{x}(t)$, $\mathbf{x}(t) \in \mathbb{R}^d$.
2. select the winner $\mathbf{m}_c(t)$ by $c = \mathrm{argmin}_i \|\mathbf{x}(t) - \mathbf{m}_i(t)\|$, $i \in \{i|s(i) < 0\}$.
3. update the winner and its neighbors:
   FOR each node $\mathbf{m}_i$

$$\mathbf{m}_i(t+1) \leftarrow \mathbf{m}_i(t) + h_{ci}[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{1}$$

$$h_{ci}(t) = \alpha(t) exp\left(-\frac{\mathbf{A}}{2\sigma^2(t)}\right) \tag{2}$$

4. $\alpha(t) \leftarrow (1-s)min_\alpha + s\ max_\alpha$
   $\sigma(t) \leftarrow (1-s)min_\sigma + s\ max_\sigma$
   where $s = \frac{exp(\varphi t/T) - exp(\varphi)}{1-exp(\varphi)}$

There are two differences between the above algorithm and the original SOM. First, the winner is selected from a contingent subset rather than from all of the nodes. Such local selection directly contributes to the scaling of the map, which will be elaborated in Section 2.2. Second, the neighborhood function, $h_{ci}(t)$, is interpreted as a matrix (4) instead of a scalar. Such neighborhood function allows anisotropic mapping—more precisely, adapting to the desirable directions of the gradient of the model values across the map.

2.1. Orientation

The term $\mathbf{x}(t) - \mathbf{m}_i(t)$ in the regression formula (1) is a d-dimensional vector.[2] So it is natural to view $h_{ci}(t)$ as a matrix (instead of a scalar in SOM), which we find beneficial for orientation control. In cases where the input term $\mathbf{x}(t)$ is three-dimensional (denoted with x, y, z as subscript), the increment is represented by three bivariate Gaussian functions:

$$h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] = \alpha(t)\begin{bmatrix} \exp\left(-\frac{c_x(\mathbf{P}_x\cdot\mathbf{D})^2 + c'_x(\mathbf{P}'_x\cdot\mathbf{D})^2}{2\sigma^2(t)}\right)[\mathbf{x}(t)-\mathbf{m}_i(t)]_x \\ \exp\left(-\frac{c_y(\mathbf{P}_y\cdot\mathbf{D})^2 + c'_y(\mathbf{P}'_y\cdot\mathbf{D})^2}{2\sigma^2(t)}\right)[\mathbf{x}(t)-\mathbf{m}_i(t)]_y \\ \exp\left(-\frac{c_z(\mathbf{P}_z\cdot\mathbf{D})^2 + c'_z(\mathbf{P}'_z\cdot\mathbf{D})^2}{2\sigma^2(t)}\right)[\mathbf{x}(t)-\mathbf{m}_i(t)]_z \end{bmatrix} \tag{3}$$

where $\mathbf{D}$ is the vector starting from the winner c pointing to the ith node on the map. The horizontal component $\mathbf{D}_u$ equals the distance between the two nodes in the horizontal direction, likewise for $\mathbf{D}_v$ in vertical direction (Fig. 1, left). $\mathbf{P}_x$ specifies the desirable direction of the gradient for the x-component of the current winner $\mathbf{m}_c$. For instance, if it is required that the x-component of model vectors varies along the vertical direction twice as fast as along the horizontal, then one can set $\mathbf{P}_x = (\pm 1, 0)$ (modulo π) and $c_x/c'_x = 1/2$. Formally, $\mathbf{P}_x$ presents the first eigenvector of the desired distribution (gradient) for the x-component of model $\mathbf{m}_c$. The second eigenvector $\mathbf{P}'_x$ is perpendicular to the first. Hence the neighborhood function in (3) is the matrix exponential:

$$h_{ci}(t) = \alpha(t)\exp\left(-\frac{1}{2\sigma^2(t)}\begin{bmatrix} \mathbf{P}^2_{xu} & \mathbf{P}^2_{xv} & 2\mathbf{P}_{xu}\mathbf{P}_{xv} \\ \mathbf{P}^2_{yu} & \mathbf{P}^2_{yv} & 2\mathbf{P}_{yu}\mathbf{P}_{yv} \\ \mathbf{P}^2_{zu} & \mathbf{P}^2_{zv} & 2\mathbf{P}_{zu}\mathbf{P}_{zv} \end{bmatrix} \times \begin{bmatrix} \mathbf{D}^2_u & \mathbf{D}^2_v \\ \mathbf{D}^2_v & \mathbf{D}^2_u \\ \mathbf{D}_u\mathbf{D}_v & -\mathbf{D}_u\mathbf{D}_v \end{bmatrix}\begin{bmatrix} c_x & c_y & c_z \\ c'_x & c'_y & c'_z \end{bmatrix} \odot \mathbf{I}\right). \tag{4}$$

$\mathbf{I}$ is the identity matrix. The Hadamard product $\odot$ leads to a diagonal matrix whose exponential is again a diagonal matrix. As illustrated in Fig. 1 and in (4), multiple heterogeneous kernels generalize the notion of neighborhood function. When $\mathbf{P}_x = \mathbf{P}_y = \mathbf{P}_z = (\sqrt{1/2}, \sqrt{1/2})$ and $c_x = c'_x = c_y = c'_y = c_z = c'_z = 1$, formula (4) is reduced to a single homogeneous Gaussian. Therefore, SOM is a special case of OS-Map. The orientation of the cortical maps has been studied by Obermayer, Ritter, and Schulten (1990) and others. However, their maps are still isotropic with respect to each dimension of the input space (Fig. 1 left). By contrast, our model allows anisotropic mapping in each dimension of the input $\mathbf{x}(t)$ respectively (Fig. 1 right).

Orientation control in learning geometric data is critical. In the following examples (Figs. 2 and 3), SOM fails to learn the geometry, while OS-Map catches the geometric features correctly. Local orientations collectively contribute to the global arrangement of the map. The input item $\mathbf{x}(t)$, a point in 3D Euclidean space, is randomly and uniformly sampled from the roll/torus surface. The

[1] $\varphi = -2.5$ in our programs.

[2] d: the dimension of the input data space.

Download English Version:

https://daneshyari.com/en/article/404050

Download Persian Version:

https://daneshyari.com/article/404050

Daneshyari.com