Supplementary Data

Supplementary Data S1

In this supplementary data we describe with mathematical details four descriptors used to classify the dendritic spines.

S1. 2D SPINE ANALYSIS (2D DESCRIPTOR)

S1.1. Basic theory

In this 2D spine analysis (2dSpAn) descriptor, a grayscale image (I) is used as input. Dendrites with spines are defined as the object region ($\mathcal{O}$) and the remaining region is considered as the background ($B = I - \mathcal{O}$). For segmentation of dendritic spines, first, two points are marked on a dendrite (by a researcher) to identify it. Then, a region of interest (ROI) is estimated around these two points, and denoted as R_i where $R \subseteq I, i = 1, 2, \ldots, n$ if n is the total number of ROIs identified from the image. Subsequently an adaptive binarization algorithm (Otsu, 1975) is used to binarize R_i. After binarization, a set of convolution kernels proposed by Basu et al. (2007) is applied to segment the dendritic portion of R_i. The convolution kernels K_α are represented as mutually exclusive set of three tuples, such that $K_\alpha : \{(\alpha, k_{11}, k_{12}), (\alpha, k_{21}, k_{22}) \ldots (\alpha, k_{M1}, k_{M2})\}$, where α represents the angular spread (in degree) of the kernel and $M = \frac{360}{\alpha}$. Designed convolution kernels can be applied with different values $\alpha = 90$, $\alpha = 45$, etc. for better segmentation result. During the segmentation, first kernel k_{i1} from the first marked point is applied until second point or a boundary condition is reached. Let the expanded region in this step be $R_1 \subset \mathcal{O}$. Next, second kernel k_{i2} is applied from the second marked point, until the first point or a boundary condition is reached. Let the region expanded after applying k_{i2} be $R_2 \subset \mathcal{O}$. Then a region $R_3 = R_1 \cap R_2$ is estimated as the intersection of the two expanded regions generated by two kernels k_{i1} and k_{i2}. Finally, a spine seed is selected manually as a point inside a potential dendritic spine region bounded by B and R_3. The bounded region, expanded from S_k using a region growing algorithm (Adams and Bischof, 1994), is marked as segmented spine. The segmentation result may change with variation of α and binarization threshold.

S1.2. Quantitative assessment of dendrite spine morphology

Here we describe mathematical equations written in 2dSpAn that help us in dendritic spine classification into one of four categories:

1. Stubby
2. Mushroom
3. Filipodia
4. Spine-head protrusion (in our study we call thin)

The formal definitions of the base, neck, and head of the spine are used by the standard mathematical notations of digital topology and geometry. First, the base B_k of the dendritic spine S_k is identified as the set of points (S_{Base}) in R_3 such that S_{Base} and S_k are adjacent. The central base point (CBP) of spine S_k, $CBP_k = \frac{1}{B_k} \sum ip_i$, where $|B_k|$ is the cardinality of the set of points in B_k, $CBP_k \in R_3 \cup S_k$ and $p_i \in B_k$. Distance transform (DT) (Borgefors, 1986; Saha et al., 2015) in R is calculated by considering S_k as the object and $(R - S_k)$ as background. Let the DT values (distance of each point from the background) for any point $p \in R$ be $\Omega(p)$. The local scale of an object pixel is defined as the DT value (depth) at the closest locally deepest pixel. Specifically, a pixel $p \in S_k$ is said to be locally deepest if $\forall q \in N_l(p), \Omega(q) \leq \Omega(p)$, where $N_l(p)$ the $(2l + 1)^3$ neighborhood of p, $l = 2$, is used to avoid noisy local maxima. To find the length of the spine, from the farthest point to CBP_k in S_k, DT is computed in S_k by considering CBP_k as background and represent it as $\hat{\Omega}$. The central head point (CHP) is defined as $CHP_k = \frac{1}{HP_k} \sum ip_i$, where $HP_k \subset S_k$ is the set of locally deepest points in S_k, $|HP_k|$ is the cardinality of the set of points in HP_k, $CHP_k \in S_k$, and $p_i \in HP_k$. The farthest point (FP), $p \in S_k$, is defined such that, $\forall q \in S_k$, $\hat{\Omega}(q) \leq \hat{\Omega}(q)$ CBP, CHP, and FP play key roles in estimation of the spine attributes such as length of the spine, neck length, neck width, and head width. A geodesic path is defined from base to head BH_k of the spine S_k, joining the two central points CHP_k and CBP_k such that $\sum_{p_i \in BH_k} \Omega(p_i)$ is minimized (in other words BH_k is the minimum distance path between CBP_k and CHP_k). Likewise, the central path from head to farthest point HF_k of the spine S_k is computed by joining CHP_k and FP_k such that $\sum_{p_i \in BH_k} \Omega(p_i)$ is minimized (in other words HF_k is the minimum distance path between CHP_k and FP_k). The neck length is estimated of S_k as $NL_k = BH_k - \Omega(CHP_k)$. Minimum neck width is estimated as $MNW = \min_{\forall p_i \in BH_k} \Omega(p_i)$. Average head width of the spine S_k is estimated as $AHW_k = \mathrm{avg}_{\forall p_i \in BH_k} \Omega(p_i)$ and the length of the spine is estimated as $L_k = |BH_k| + |HF_k|$. Hereunder we show the algorithm that classifies the spines into one of four categories:

```
for (each segmented spine $S_K \in S$) do
if ($NL_k = = 0$) then $S_k =$ Stubby
if (($NL_k > 0$) && $\frac{D_{max}(HP_k)}{L_k} > \gamma$ then $S_k =$ filipodia
if ($NL_k > 0$) && ($S_k \neq Filipodia$) && $\frac{BH_k}{L_k} < \delta$ then $S_k =$ Mushroom
if ($NL_k > 0$) && ($S_k \neq Filipodia$) && ($S_k \neq Mushroom$) then $S_k =$ Spine-Head Protrusions (in our study we call it as thin)
```

where HP_k are locally deepest points, $D_{max}(HP_k)$ is the distance between the two farthest locally deepest points, and δ and γ are thresholds.

For more in-depth information about 2dSpAn, descriptor see Basu et al. (2016).

S2. SPINE TOOLS (2D DESCRIPTOR)

This descriptor was written in Python 3 language and is based on virtual skeleton and cluster analysis. For take into consideration purpose of shape analysis and to classify spines into classes, the images of individual spines were first straightened, which is transformed so that the virtual skeleton of each spine formed a straight line. Next, the images were rescaled to normalize the spine area, and for each spine $d(h)$, diameter of the spine as a function of distance from the dendrite, was found. To classify the spines according to shape, a two-step procedure was used: first, all $d(h)$ functions were clustered, next, the clusters were manually sorted into three groups (mushroom, stubby, and thin spines) based on average images of the clusters and visual inspection of spines comprising the clusters. For details about this method see Jasińska et al. (2016).

S3. NEUROLUCIDA 360 (3D DESCRIPTOR)

This method is based, in part, on the laboratory version of NeuronStudio, originally developed at the Icahn School of Medicine at Mount Sinai by Wearne and colleagues (Rodriguez et al., 2003, 2006, 2008; Wearne et al., 2005). Unfortunately no more information about the algorithm was presented in the article, only the protocols and instructions how to use different tools in this descriptor (Dickstein et al., 2016).

S4.3D SPINE ANALYSIS (3D DESCRIPTOR)

S4.1. Basic theory

This method allows users to mark a specific dendritic spine to segment the spine as three-dimensional (3D) volume and extract relevant morphometric features with high accuracy and minimal user intervention. This method is also based on digital topology and geometry but in 3D which means cubic grid represented by $Z^3|Z$, the set of integers. A grid point referred to as a point or voxel is an element of Z^3 and represented by a triplet of integer coordinates (x, y, z). An object $\mathcal{O}$ is a fuzzy subset $\{(p, \mu_{\mathcal{O}}(p)) | p \in Z^3\}$ of Z^3, where $\mu_{\mathcal{O}}: Z^3 \rightarrow [0, 1]$ is the membership function. The support O of an object $\mathcal{O}$ is $O = \{p | p \in Z^3 \text{and} \mu_{\mathcal{O}}(p) \neq 0\}$; $\bar{O} = Z^3 - O$ is the background. Let S denote a set of voxels. A path π in S from $p \in S$ to $q \in S$ is a sequence of voxels $\langle p = p_1, p_2, \ldots, p_l = q \rangle$ in S such that every two successive voxels are adjacent. A link is a path $\langle p, q \rangle$ of exactly two adjacent voxels. The length of a path $\pi = \langle p_1, p_2, \ldots, p_l \rangle$ in a fuzzy object $\mathcal{O}$, denoted as $\pi_{\mathcal{O}}(\pi)$.

$$\pi_{\mathcal{O}}(\pi) = \sum_{i=1}^{l-1} \frac{1}{2} (\mu_{\mathcal{O}}(p_i) + \mu_{\mathcal{O}}(p_{i+1}))||p_i - p_{i+1}||, \tag{12}$$

where $||p_i - p_{i+1}||$ is the Euclidean distance between p *and* q. The fuzzy distance (Saha et al., 2002; Saha and Wehrli, 2004) between two voxels $p, q \in Z^3$ in an object $\mathcal{O}$, denoted by $\omega_{\mathcal{O}}(p, q)$, is the length of one of the shortest paths from p to q: $\omega_{\mathcal{O}}(p, q) = min_{\pi \in P(p,q)} \Pi_{\mathcal{O}}(\pi)$, where $P(p, q)$ is the set of all paths from p to q. The fuzzy DT (FDT) of an object $\mathcal{O}$ is an image $(p, \Omega_{\mathcal{O}}(p))|p \in Z^3$, where $\Omega_{\mathcal{O}} : Z^3 \rightarrow \Re^+|\Re^+$ is the set of positive real numbers, including 0, that is the fuzzy distance from the background: $\Omega_{\mathcal{O}}(p) = \min_{q \in O} \omega_{\mathcal{O}}(p, q)$.

In case of segmentation of dendrite and dendritic spine, we can define the dendrite as fuzzy object $\mathcal{O}_{Dendrite}$ and spine as $\mathcal{O}_{Spine}$. The main challenges are

1. separating the fuzzy objects $\mathcal{O}_{Spine}$ from $\mathcal{O}_{Dendrite}$, which are fused at various unknown locations and scales, and
2. morphologically defining spine compartments in the segmented fuzzy objects $\mathcal{O}_{Spine}$.

The first challenge is solved using the multiscale opening (MSO) algorithm (Saha and Udupa, 2001) in two steps.

Step 1: Segmentation of the combined region $\mathcal{O}_{Spine} \cup \mathcal{O}_{Dendrite}$ from the background.
Step 2: Separation of $\mathcal{O}_{Spine}$ and $\mathcal{O}_{Dendrite}$.

The first step is trivially achieved using simple thresholding (Saha et al., 2016) and connectivity analysis (Zhiyon et al., 2012). Let $\mathcal{O}$ be the fuzzy segmentation of the combined region that is obtained in Step 1. In all subsequent analyses, the support O of $\mathcal{O}$ is considered as the "effective image space." Let I:O $\rightarrow$ $[I_{min}, I_{max}]$ be the image intensity function over O. In the second step, segmentation is modeled as the opening of two fuzzy objects mutually fused at different unknown locations and scales in the shared intensity space, I. Here, the main challenge is to determine the local size of the suitable morphological operator that can separate spines from the dendrite. The developed MSO algorithm combines FDT (Saha et al., 2010) and fuzzy connectivity (Udupa and Saha, 2003) to iteratively open the two objects in I.

S4.2. MSO algorithm

The idea of the MSO algorithm (Saha et al., 2016) is to gradually erode the assembly of two fused objects until they become mutually disconnected and create two separate objects. The first iteration starts with two sets of seed voxels, S_{Spine} and $S_{Dendrite}$, and a set of common separators, S_S. The initial FDT map $\Omega_{Spine,0}$ for the first object is computed from O, except that the voxels in $S_{Dendrite} \cup S_S$ are added to the background. The FDT map $\Omega_{Dendrite,0}$ for the other object is calculated in the same way. The sets S_{Spine}, $S_{Dendrite}$, and S_S are mutually exclusive. The coupling of two objects is described by the intensity overlap of dendritic segment and the spines, which are fused with each other at different unknown locations and scales. Let $\mu_{Dendrite}$ and μ_{Spine} denote the dendrite and spine membership functions, defined as the following:

$$\mu_{Dendrite}(p) = \begin{cases} 0 & \text{if} \quad I(p) < I_{Spine} \\ \frac{I(p) - I_{Spine}}{I_{Dendrite} - I_{Spine}} & \text{if} \quad I_{Spine} \leq I(p) < I_{Dendrite} \\ 1 & \text{if} \quad I(p) \geq I_{Dendrite} \end{cases},$$

$$\mu_{Spine}(p) = \begin{cases} 0 & \text{if} \quad I(p) < I_{min} \\ 1 & \text{if} \quad I_{min} \leq I(p) < I_{Spine} \\ \frac{I_{Dendrite} - I(p)}{I_{Dendrite} - I_{Spine}} & \text{if} \quad I_{Spine} \leq I(p) < I_{Dendrite} \\ 0 & \text{if} \quad I(p) \geq I_{Dendrite} \end{cases},$$

where $I : O \rightarrow [I_{min}, I_{max}]$ is the image intensity function over O. I_{Spine} and $I_{Dendrite}$ are the representative spine and dendrite intensities that define the transition between pure and shared intensity space. Let $P_{Dendrite} \subset O$ and $P_{Spine} \subset O$ be the set of voxels that belong to the pure intensity band for dendrite and spine, respectively. Thus, the set of voxels that fall within the shared intensity band is $O_{Shared} = O - P_{Dendrite} - P_{Spine}$. A fuzzy representation of the composite object is obtained by taking the fuzzy union of the two membership functions. The iterative approach of the MSO of two structures takes several iterations to grow the path continuity of an object, starting from its seed voxels (commonly added in large-scale regions) to a peripheral location with fine-scale details. After the iterative propagation of the MSO algorithm, the dendrite region is segmented as a single connected component. O_{Spine} represents one or more disjointed spine regions (R_i), such that $O_{Spine} = \bigcup_{i=1}^{K} R_i$, where K is the total number of disjointed spine segments in O_{Spine}, and each such segmented spine region R_i contains at least one spine seed $p \in S_{Spine}$.

S4.3. Morphological definitions for the spine regions

Once the spines are segmented from the dendrite, the challenge is to assess the morphological attributes accurately. In this study, several key morphological features of 3D dendritic spines are defined. Specifically, four key spine features that are related to the base and head of a spine are defined using standard notations of digital topology and geometry (Borgefors, 1986; Saha et al., 2015). The base of spine, for a given spine $R_i \cup O_{Spine}$, is defined as the set of points $B_i \cup O_{Dendrite}$ such that $\forall p \in B_i$, $\exists q \in R_i$ is adjacent to p. The CBP, for a given spine $R_i \cup O_{Spine}$, is the centroid of the base of the spine R_i (i.e., $CBP_i = \frac{1}{|B_i|} \sum \forall p \in B_i p$, where $| \cdot |$ is the cardinality of a set).

The head and tip of a spine are defined using the FDT (Borgefors, 1986; Saha et al., 2015) Ω_i of R_i. A locally deepest point in a spine R_i is a point $p \in R_i$ such that $\forall q \in N_l(p)\Omega_i(q) \leq \Omega_i(p)$, where $N_l(p)$ is the $(2l+1)^3$ neighborhood of p. Here, $l=2$ is used to avoid noisy local maxima. The center of the head CH_i of a spine R_i is the locally deepest point in the spine. If multiple locally deepest points criterion, their centroid is used. The tip of a spine T_i of a spine R_i is a point $T_i \in R_i$ that is farthest from its CBP_i. In a situation wherein multiple points of R_i satisfy the farthest distance criterion, their centroid is used. CBP_i, CH_i, and T_i play key roles in estimating spine attributes, such as length of the spine, neck length, neck width, and head width, for each individual spine R_i. The geodesic path from base to head BH_i of the spine R_i is estimated by joining the two central points CBP_i and CH_i such that $\sum_{\forall p \in BH_i} \Omega_i(p)$ is minimized. Likewise, the central path from head to spine tip HT_i of the spine R_i is computed by joining CH_i and T_i, such that $\sum_{\forall p \in HT_i} \Omega_i(p)$ is minimized. The neck length NL_i of R_i is $NL_i = BH_i - \Omega_i(CH_i)$. Minimum neck width MNW_i of R_i, $MNW_i = min_{\forall p \in BH_i}(\Omega_i(p))$. Average head width AHW_i of the spine R_i, $AHW_i = avg \forall p \in HP_i(\Omega_i(p))$ such that HP_i is the set of all locally deepest points in R_i. Finally, the length of the spine Li is estimated as $L_i = |BH_i| + |HT_i|$.

For more general information about 3D spine analysis (3dSpAn) descriptor, see Basu et al. (2018).

S5. VITERBI ALGORITHM DESCRIPTION

The Viterbi algorithm was introduced by Viterbi (1967) as a decoding algorithm for convolution codes over noisy digital communication links. It is a dynamic programming algorithm to find the most likely sequence of hidden states. In hidden Markov model (HMM), the observation is independent of all the other observations. In this situation, Viterbi algorithm only uses the information of the previous state of a hidden state and does not consider the dependency among observations. Of course, some authors introduce approaches in which the dependency among observations are considered. For this purpose, the extension of HMM called autoregressive HMM (ARHMM) is considered (Rezaei Tabar et al., 2018).

SUPPLEMENTARY DATA S2

In this supplementary data, we describe in detail how to prepare data from experiment to use in the ARHMM algorithm. Our data should be saved as matrix, each column representing one timestamp, each row representing a single dendritic spine. Each dendritic spine numerically encodes the spine classes (e.g., stubby class ‘‘s’’ was encoded as 1, mushroom class ‘‘m’’ as 2, etc.). After this we could finally proceed to run the descriptor using the following parameters:

data, the matrix with the data;
level, how many classes are used (in our study we use 4 and 5);
transition probability matrix, a square matrix $n \times n$, where n is the number of timestamps. Here, since we had three timestamps, the matrix was of shape 3×3. Since each row represents transition probability from a given state, the sum of the values in a row must be equal to 1;

initial probability vector, vector with probabilities, of size n, representing the initial probability of each state (sum must be equal to 1);

number of iterations, total number of iterations;

dependence matrix, this matrix is defined as $P(O_t|S_t, O_{t-1}=i)$. In other words, it represents the probability of getting particular observation O_t given the previous observation O_{t-1} and current hidden state S_t. Again sum of each row must be equal to 1. The size of the matrix is $k \times n$, where k is the number of spine classes and n is the number of columns.

Hereunder we show how the matrices and vectors look like and what values were used to plot curves on Figures 2–4.

Transition probability matrix, $\begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ & & \end{bmatrix}$;

initial probability vector, [0.3;0.4;0.3];

dependence matrix,

for 4 classes, $\begin{bmatrix} 0.3 & 0.2 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.4 & 0.1 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ & & & \end{bmatrix}$;

for 5 classes, $\begin{bmatrix} 0.3 & 0.2 & 0.2 & 0.2 & 0.1 \\ 0.4 & 0.1 & 0.2 & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.1 & 0.2 & 0.2 \\ & & & & \end{bmatrix}$.

SUPPLEMENTARY DATA S3

In this supplementary data, we show results with comments about shape analysis based on parameters that were given from 3dSpAn descriptor after segmentation and classification by this descriptor. Principal component analysis method was used (separately) on all three timestamps for parameters (volume, length, head width, and neck length) calculated by 3dSpAn descriptor. For the first two features (components) in timestamps t_0 and t_{10}, the reduced representation we cover $\sim$72.7% of the variance. For timestamp t_{40} for first two features (components), we cover 71.8% of the variance. These results are very similar/close to results that we have in our previous study (Urban et al., 2019). This can be explained by number of parameters that describe dendritic spines (in 3D we have 4 parameters and in two-dimension [2D] we have 11 parameters).

In fuzzy partition coefficient (fpc), we could observe that the highest value was for only one center—in contrast to many centers from 2D as in Urban et al. (2019). Also, here we do not have the highest value (1) but only $\sim$0.9. The difference can be that 3D classification does not take into account subpopulations of the dendritic spines (e.g., longer and shorter dendritic spines for filipodia class). Because of this result, it can be hard (or even useless) to use hierarchical clustering where one of the parameters that we must give is number of centers. Also, the curve for fpc is logarithmic going down (more centers mean less fpc value). One of the explanations of this situation is too small data set (300 dendritic spines), second is that four parameters describing each dendritic spine can also be too small [in Urban et al. (2019), data set was for 2000 dendritic spines and 11 parameters but for only 1 timestamp t_0].

SUPPLEMENTARY DATA S4

In this supplementary data, we present the estimated transition probability matrix for version 5 (square dark gray line from Fig. 2B):

$$\begin{bmatrix} 0.0000 & 0.4570 & 0.5429 \\ 0.9966 & 0.0033 & 0.0000 \\ 0.0000 & 0.0549 & 0.9451 \\ & & \end{bmatrix}$$

and the estimated dependence probability matrix for version 4 (light gray vline from Fig. 3A):

$$\begin{bmatrix} 0.3457 & 0.1345 & 0.4150 & 0.1049 \\ 0.2385 & 0.1382 & 0.4382 & 0.1851 \\ 0.4719 & 0.2654 & 0.1353 & 0.1274 \\ & & & \end{bmatrix}.$$

The matrix for transition can be explained that first row (and first column) is t_0, second row (and column) is t_{10}, and third row (and column) is t_{40}. The transition is between states that are in our study timestamps. The transition probability is the conditional probability, it means that we are now at time t given the previous state was at time $t-1$, the zero probability (first row, first column) means that the transition probability from being at state t_0 at time t given being at state t_0 at time $t-1$ is 0.

The dependence matrix is constructed this way: first row is t_0, second row is t_{10}, and third row is t_{40}. Column is dendritic spine class: first column is stubby, second column is mushroom, third column is filipodia, and last column is not existing. Probability that the dendritic spine from time point t_0 with class stubby will still be in class stubby in time point t_{10} with probability 0.2385. This information can be used for example in building networks.

For 3dSpAn for four classes (gray square line from Fig. 4B), the best estimated transition and dependence probability matrices are as follows, respectively:

$$\begin{bmatrix} 0.0000 & 0.9725 & 0.0275 \\ 0.6754 & 0.3120 & 0.0126 \\ 0.0000 & 0.6098 & 0.3902 \\ & & \end{bmatrix}$$

$$\begin{bmatrix} 0.8108 & 0.0415 & 0.0079 & 0.1398 \\ 0.4824 & 0.1807 & 0.1010 & 0.2359 \\ 0.4884 & 0.0536 & 0.4580 & 0.0000 \\ & & & \end{bmatrix}.$$

SUPPLEMENTARY REFERENCES

Adams, R., and Bischof, L. 1994. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 641–647.

Basu, S., Chaudhuri, C., Kundu, M., et al. 2007. Text line extraction from multi-skewed handwritten documents. *J. Pattern Recogn.* 40, 1825–1839.

Basu, S., Plewczynski, D., and Saha, S. 2016. 2dSpAn: Semiautomated 2-D segmentation, classification and analysis of hippocampal dendritic spine plasticity. *Bioinformatics* 32, 2490–2498.

Basu, S., Saha, P.K., Roszkowska, M., et al. 2018. Quantitative 3-D morphometric analysis of individual dendritic spines. *Sci. Rep.* 8, 1–13.

Borgefors, G. 1986. Distance transformations in digital images. *Comput. Vis. Graphics Image Process.* 34, 344–371.

Dickstein, D.L., Dickstein, D.R., Janssen, W.G.M., et al. 2016. Automatic dendritic spine quantification from confocal data with Neurolucida 360. *Curr. Protoc. Neurosci.* 10, 1–27.

Jasińska, M., Miłek, J., Cymerman, I.A., et al. 2016. miR-132 regulates dendritic spine structure by direct targeting of matrix metalloproteinase 9 mRNA. *Mol. Neurobiol.* 53, 4701–4712.

Otsu, N. 1975. A threshold selection method from gray-level histograms. *Automatica* 11, 23–27.

Rezaei Tabar, V., Plewczynski, D., and Fathipour, H. 2018. Generalized Baum-Welch and Viterbi Algorithms based on the direct dependency among observations. *J. Iran. Stat. Soc.* 17, 205–225.

Rodriguez, A., Ehlenberger, D., Kelliher, K., et al. 2003. Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images. *Methods* 1, 94–105.

Rodriguez, A., Ehlenberger, D.B., Dickstein, D.L., et al. 2008. Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PLoS One* 3, 1–12.

Rodriguez, A., Ehlenberger, D.B., Hof, P.R., et al. 2006. Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. *Nat. Protoc.* 4, 2152–2161.

Saha, P.K., Basu, S., and Hoffman, E.A. 2016. Multiscale opening of conjoined fuzzy objects: Theory and applications. *IEEE Trans. Fuzzy Syst.* 24, 1121–1133.

Saha, P.K., Gao, Z., Alford, S.K., et al. 2010. Topomorphologic separation of fused isointensity objects via multiscale opening: Separating arteries and veins in 3-D pulmonary CT. *IEEE Trans. Med. Imaging* 29, 840–851.

Saha, P.K., Strand, R., and Borgefors, G. 2015. Digital Topology and Geometry in Medical Imaging: A Survey. *IEEE Trans. Med. Imaging* 34, 1940–1964.

Saha, P.K., and Udupa, J.K. 2001. Optimum image thresholding via class uncertainty and region homogeneity. *Trans. Pattern Anal. Mach. Intell.* 23, 689–706.

Saha, P.K., and Wehrli, F.W. 2004. Measurement of trabecular bone thickness in the limited resolution regime of in vivo MRI by fuzzy distance transform. *IEEE Trans. Med. Imaging* 23, 53–62.

Saha, P.K., Wehrli, F.W., and Gomberg, B.R. 2002. Fuzzy distance transform: Theory, algorithms, and applications. *Comput. Vis. Image Und.* 86, 171–190.

Udupa, J.K., and Saha, P.K. 2003. Fuzzy connectedness and image segmentation. *Proc. IEEE* 91, 1649–1669.

Urban, P., Rezaei Tabar, V., Bokota, G., et al. 2019. Dendritic spines taxonomy: The functional and structural classification • Time-dependent probabilistic model of neuronal activation. *J. Comput. Biol.* 26, 1–14.

Viterbi, A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* 13, 260–269.

Wearne, S.L., Rodriguez, A., Ehlenberger, D.B., et al. 2005. New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience* 3, 661–680.

Zhiyon, G., Grout, R.W., Holtze, C., et al. 2012. A new paradigm of interactive artery/vein separation in noncontrast pulmonary CT imaging using multiscale topomorphologic opening. *IEEE Trans. Biomed. Eng.* 59, 3016–3027.