An algorithm for finding candidate synaptic sites
van Pelt, J.; Carnell, A.; de Ridder, A.; Mansvelder, H.D.; van Ooyen, A.

published in
Frontiers in Computational Neuroscience
2010

DOI (link to publisher)
10.3389/fncom.2010.00148

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)
van Pelt, J., Carnell, A., de Ridder, A., Mansvelder, H. D., & van Ooyen, A. (2010). An algorithm for finding candidate synaptic sites. Frontiers in Computational Neuroscience, 4, 148. https://doi.org/10.3389/fncom.2010.00148

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 12. Jan. 2025
Neurons make synaptic connections at locations where axons and dendrites are sufficiently close in space. Typically the required proximity is based on the dimensions of dendritic spines and axonal boutons. Based on this principle one can search those locations in networks formed by reconstructed neurons or computer generated neurons. Candidate synapses are then located where axons and dendrites are within a given criterion distance from each other. Both experimentally reconstructed and model generated neurons are usually represented morphologically by piecewise-linear structures (line pieces or cylinders). Proximity tests are then performed on all pairs of line pieces from both axonal and dendritic branches. Applying just a test on the distance between line pieces may result in local clusters of synaptic sites when more than one pair of nearby line pieces from axonal and dendritic branches is sufficient close, and may introduce a dependency on the length scale of the individual line pieces. The present paper describes a new algorithm for defining locations of candidate synapses which is based on the crossing requirement of a line piece pair, while the length of the orthogonal distance between the line pieces is subjected to the distance criterion for testing 3D proximity.

**Keywords:** neuronal morphology, synaptic connectivity, line crossing, spatial proximity

**INTRODUCTION**

Activity dynamics underlying cognition depends crucially on the patterns and strengths of synaptic connections between neurons. During development neurons grow out by elongation and branching of dendritic and axonal arbors. Synaptic connections can form when axonal and dendritic branches of neurons come sufficiently close to each other (Peters, 1979). The geometry of neuronal arborizations is therefore an important determinant of synaptic connectivity. How neuronal morphology shapes neuronal network connectivity is, however, still poorly understood. New developments in experimental and computational approaches make it now possible to shed light on this question. With multipatch techniques experimentalists are now measuring connectivities between the patched neurons using electrophysiological and anatomical criteria (e.g., Feldmeyer et al., 1999, 2002, 2006; Le Bé et al., 2007; Frick et al., 2008). Computational approaches make it possible to assemble a number of neurons (reconstructed or algorithmically generated) in a 3D geometrical arrangement and to study the emerging synaptic connectivity by searching the sites where potential synapses may occur. For instance, neuroConstruct (Gleeson et al., 2007) is a modeling tool for creating neuronal networks in 3D space using imported neuronal morphologies. The simulator L-Neuron creates virtual neurons using an L-systems approach by iteratively sampling experimental distributions of neuronal shape parameters (Senft and Ascoli, 1999; Ascoli and Krichmar, 2000; Samsonovich and Ascoli, 2007). Similarly, the simulator tool NeuGen (Eberhard et al., 2006) generates neuronal networks in 3D with morphologically realistic neurons by sampling experimental distributions of morphological shape parameters. Recently, Luczak (2006) generates neuronal morphologies using a diffusion limited aggregation (DLA) approach. The simulator CX3D (Zubler and Douglas, 2009) aims at simulating cortical development in 3D space, including the morphology of single neurons. Cuntz et al. (2010) apply a minimal spanning tree principle in generating neuronal morphologies. Costa and Coelho (2005, 2008) generate 2D neuronal morphologies by statistically sampling a probabilistic model of neuronal geometry based on branch probabilities per branch level using a Monte Carlo approach, and form connections when neuronal trees overlap in 2D. The simulator NETMORPH (Koene et al., 2009) is based on biological growth principles of neurons by stochastically modeling the elongation and branching of growth cones in developmental time.

Spatial closeness between axonal and dendritic branches, typically determined by the dimensions of dendritic spines and axonal boutons, is considered as a prerequisite for synapse formation. Hellwig (2000) used reconstructed neurons for estimating connectivity between overlapping dendritic and axonal arborizations. To this end reconstructed neurons were transformed into voxel fields (with voxels of 1 μm side length) followed by counting the number of voxels containing both axonal and dendritic elements and obtaining a number of potential synaptic sites, thus adopting a distance criterion of 1 μm. Kalisman et al. (2003) used sets of 3D reconstructions of biocytin-stained cells to obtain axonal and dendritic statistical representations in order to calculate the probabilities of close appositions. Stepanyants et al. (2002) and Stepanyants and Chklovskii (2005) estimated the number of potential synaptic sites in terms of axonal and dendritic length densities per unit volume, typical dendritic spine
length and average angle between axonal and dendritic branches. Amirikian (2005) used the concept of synaptic clouds (density fields) as representation of the complex structure of axonal/dendritic arbors for studying connectivity between different cell classes.

With the recently developed simulator NETMORPH, assemblies of neurons can be generated with realistic 3D morphologies and realistic morphological variability in any arbitrary 3D arrangement (Koene et al., 2009). The emerging connectivity in these networks can be determined by searching the locations where axons and dendrites are sufficient close to each other for synapse formation, thus identifying the sites of candidate synapses. Both in experimentally reconstructed neurons and in algorithmically generated neurons (such as by NETMORPH) axons and dendrites are approximated by piecewise-linear structures, i.e., represented by chains of line pieces or cylinders. Testing the proximity of axonal and dendritic branches then implies the search for pairs of individual dendritic and axonal line pieces that are closer to each other than a given criterion distance. When an axonal and dendritic branch are sufficiently close in space, the application of just a distance criterion, however, may result in several pairs of nearby line pieces obeying the proximity criterion and thus may fall into a cluster of candidate synaptic sites (see Figure 1B). In addition, the number of sufficiently close line piece pairs appears to depend on the typical length scale of the line pieces in the dendritic and axonal representation. As the clustering and the length scale dependency were considered to be artifacts of this approach a method was needed to identify uniquely the location where an axonal and dendritic branch are closest in space. To this end an additional requirement was introduced by searching for that line piece pair from an axonal and dendritic chain of line pieces that cross in space and taking the shortest (orthogonal) distance for that pair as the shortest distance between the axonal and dendritic branch (see Figure 1C). This shortest distance is subsequently subjected to the distance criterion for candidate synapses. Note that in the case of cylinders the distance is taken between their axial center lines.

The characterization of network connectivity has also become an important area of research. Graph-theoretical studies of brain connectivity focus amongst others on the presence of hubs and small-world properties (e.g., Watts and Strogatz, 1998; Stepanyants et al., 2002; Hilgetag and Kaiser, 2004; Stepanyants and Chklovskii, 2005). Other studies emphasize the hierarchy in neural network connectivity (e.g., Kaiser et al., 2010) or different spatial scales (e.g., Passingham et al., 2002; Sporns et al., 2005; Stam and Reijneveld, 2007). An interesting aspect during development of network connectivity is the critical phenomenon of percolation when the largest cluster of connected neurons makes an abrupt transition in size toward a giant cluster as studied by Costa and Manoel (2002) and Costa and Coelho (2005, 2008), who also showed the dependence of this phenomenon on the morphology of the developing neurons. Many of these studies rely on methods for estimating synaptic connectivity between axonal and dendritic arborizations, underscoring the need for algorithms that estimate the connectivity as realistic as possible.

The paper describes an algorithm for testing the proximity between chains of line pieces on the basis of crossing properties of line piece pairs. The paper includes a Section “Materials and Methods” with the derivation of geometrical conditions for crossing line pieces using a classical geometry approach (applying a series of 3D transformations). In the Section “Results” the outcomes of the synapse detection model are shown without and with the crossing requirement. A study of the multiplicity of the connections (the number of synapses between two connected neurons) is included. The paper includes an “Rotations in 3D” of Appendix summarizing the 3D rotation matrices, “Relative Position of Point D of the Perpendicular From Point C to Opposite Side AB in a Triangle ABC” of Appendix with some basic triangular geometry, “Expressing the Crossing Conditions in Terms of the Original Coordinates” of Appendix formulating the crossing conditions in the Section “Crossing Line Pieces” in terms of the original coordinates, and “Summary of Calculations and Crossing Conditions” of Appendix summarizing the calculations and the crossing conditions. For reason of completeness, “Analytical Approach” of Appendix includes a different approach using analytical geometry and based on an algorithm taken from literature (Dan Sunday, http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm), which also includes the shortest distance between non-crossing line pieces.

MATERIALS AND METHODS
PROXIMITY-BASED SEARCH OF CANDIDATE SYNAPtic LOCATIONS
The search in a 3D network of neurons with complex axonal and dendritic arborizations for locations where axonal and dendritic branches are sufficiently close can proceed by searching the shortest distance between any two branches from the axonal and dendritic arborizations. The shortest distance between two straight infinite lines is at the site where they are crossing and determined by the length of the orthogonal connection line (Figure 1A). Axonal and dendritic branches in neuronal reconstructions, however, are not infinite straight lines, but composed of many line pieces in sequence with varying orientations and lengths. Then the distance between any line piece pair from the axonal and dendritic branch must be determined and tested against the criterion value. As illustrated in Figure 1B several pairs of nearby line pieces may meet the distance criterion resulting in a cluster of potential synaptic sites. In addition the number of “nearby” line pieces is dependent on the length scale of the line pieces which may influence the outcome for
Given are two line pieces \( \text{crossing} \) algorithm for determining whether two line pieces are crossing. If they do, the algorithm provides the length of the orthogonal connection and the coordinates where this connection line intersects the two line pieces. Applying this algorithm to all pairs of line pieces of an axonal and dendritic branch will test the existence of a crossing pair, and whether the shortest distance in this pair meets the distance criterion.

Algorithm for determining whether two line pieces are crossing

Given are two line pieces \( PQ \) and \( RS \) and the coordinates of the points \( P(x, y, z), Q(x, y, z), R(x, y, z) \) and \( S(x, y, z) \). A “classical” geometry approach is used in which a series of 3D transformations is applied on the two line pieces in order to bring them in such a geometric arrangement that the condition for crossing can easily be applied. These transformations include one translation and three rotations around the three axes of the orthogonal coordinate system.

**Figure 2**

**Figure 2** (A) Example of two line pieces \( A \) and \( D \) that do not cross. Although the projection \( d_{A} \) of \( D \) onto the plane \( W \) intersects with \( A \), the projection \( d_{A} \) of \( D \) onto the plane \( W \) through \( A \) and parallel to \( D \) does not. (B) Two line pieces \( A \) and \( D \) that do cross, because the projection \( d_{A} \) of line piece \( D \) onto the plane \( W \) through \( A \) and parallel to \( D \) intersects line piece \( A \). The distance between \( A \) and \( D \) is given by the length of the orthogonal connection line \( C \).

**Figure 3**

Translation of the line pieces \( PQ \) and \( RS \) toward line pieces \( pq \) and \( rs \), respectively such that point \( p \) coincides with the origin \( O \).

**Figure 4**

Rotation of line pieces \( pq \) and \( rs \) around \( Z \)-axis into \( p,q \) and \( r,s \), respectively, and such that \( q \) rotates into \( ZX \)-plane.
Here, \( l \) denotes the length of the projection of \( Oq \) onto the \( XY \)-plane with
\[
l = \sqrt{q_x^2 + q_y^2} = \sqrt{PQ^2 - q_z^2}
\] (4)
with \( PQ = \sqrt{q_x^2 + q_y^2 + q_z^2} \) denoting the length of line piece \( PQ \).

For the rotation equations we obtain
\[
\begin{align*}
1q_x &= q_x \cdot \cos \gamma - q_y \cdot \sin \gamma = \frac{q_x^2 + q_y^2}{l} = l \\
1q_y &= q_x \cdot \sin \gamma + q_y \cdot \cos \gamma = -q_y q_z + q_z q_x = 0 \\
1q_z &= q_z
\end{align*}
\] (5)

By this rotation, point \( q \) is rotated into the \( ZX \)-plane making its \( y \)-coordinate equal to zero, and its \( x \)-coordinate equal to the length of the projection of \( Oq \) onto the \( XY \)-plane. Of course, the lengths of line pieces \( Oq \) and \( Oq \) are equal to the length of the original line piece \( PQ \) (\( Oq = Oq = PQ \)). The rotations of \( r \) and \( s \) toward \( r \) and \( s \) proceed via
\[
\begin{align*}
\vec{r} &= \text{Rot}_x(\gamma) \cdot r \\
\vec{s} &= \text{Rot}_x(\gamma) \cdot s
\end{align*}
\] (6)
respectively, and we obtain
\[
\begin{align*}
1r_x &= r_x \cdot \cos \gamma - r_y \cdot \sin \gamma = \frac{1r_x + 1r_y}{l} \\
1r_y &= r_x \cdot \sin \gamma + r_y \cdot \cos \gamma = 0 \\
1r_z &= r_z
\end{align*}
\] (7)
and
\[
\begin{align*}
1s_x &= s_x \cdot \cos \gamma - s_y \cdot \sin \gamma = \frac{1s_x + 1s_y}{l} \\
1s_y &= s_x \cdot \sin \gamma + s_y \cdot \cos \gamma = 0 \\
1s_z &= s_z
\end{align*}
\] (8)

**Transformation III – Second rotation of \( q, r, \) and \( s \) around \( Y \)-axis into \( \vec{q}, \vec{r}, \) and \( \vec{s} \), respectively, such that \( \vec{q} \) is on the \( X \)-axis, thus making the line piece \( Oq \) to coincide with the \( X \)-axis**

The rotation around the \( Y \)-axis, see Figure 5, proceeds via
\[
\begin{align*}
\vec{q} &= \text{Rot}_y(\beta) \cdot \vec{q} \\
\begin{bmatrix}
1q_x \\
1q_y \\
1q_z
\end{bmatrix} &= \begin{bmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta
\end{bmatrix} \begin{bmatrix}
1q_x \\
1q_y \\
1q_z
\end{bmatrix}
\end{align*}
\] (9)

If \( \varepsilon \) equals the angle of \( Oq \) with the \( x \)-axis then the rotation angle \( \beta \) is equal to \( -\varepsilon \), thus
\[
\begin{align*}
\cos \beta &= \cos \varepsilon = \cos \varepsilon = \frac{1q_x}{Oq} = \frac{l}{PQ} \\
\sin \beta &= \sin \varepsilon = -\sin \varepsilon = \frac{1q_y}{Oq} = -\frac{q_y}{PQ}
\end{align*}
\] (10)

Thus:
\[
\begin{align*}
2q_x &= 1q_x \cdot \cos \beta - 1q_y \cdot \sin \beta = \frac{1q_x^2 + 1q_y^2}{PQ} = PQ \\
2q_y &= 1q_y = 0 \\
2q_z &= 1q_y \cdot \sin \beta + 1q_y \cdot \cos \beta = 0
\end{align*}
\] (11)

By this rotation, point \( Q \) is rotated onto the \( X \)-axis, thus giving point \( \vec{q} \) zero \( y \)- and \( z \)-coordinates and making the \( x \)-coordinate equal to the length of line piece \( PQ \).

The rotations of \( \vec{r} \) and \( \vec{s} \) toward \( \vec{r} \) and \( \vec{s} \) proceed similarly via
\[
\begin{align*}
2r_x &= \text{Rot}_y(\beta) \cdot r \\
2s_x &= \text{Rot}_y(\beta) \cdot s
\end{align*}
\] (12)
respectively, and we obtain
\[
\begin{align*}
2r_x &= r_x \cdot \cos \beta - r_y \cdot \sin \beta = r_x \cdot \frac{l + r_x \cdot q_x}{PQ} = \\
2s_x &= r_x \cdot \sin \beta + r_y \cdot \cos \beta
\end{align*}
\] (13)

using the expressions for \( \vec{r} \) and \( \vec{s} \). Because the expression for \( \vec{s} \) will occur frequently in following expressions we will assign it a special symbol \( C_{s} = s \).

\[
\begin{align*}
2r_x &= \frac{-r_x \cdot q_y + r_y \cdot q_x}{l} \\
2s_x &= \frac{r_x \cdot q_x + r_y \cdot q_y}{l}
\end{align*}
\] (14)

\[
\begin{align*}
2r_x &= \frac{-r_x \cdot q_y + r_y \cdot q_x}{l} + \frac{r_x \cdot l}{PQ} = \\
2s_x &= \frac{r_x \cdot q_x + r_y \cdot q_y}{l} + \frac{r_y \cdot l}{PQ}
\end{align*}
\] (15)
using the expressions for \( \vec{r} \) and \( \vec{s} \). In a similar way we obtain for \( \vec{s} \)
2 \mathbf{s}_x = \frac{s_x \cdot q_x + s_y \cdot q_y + s_z \cdot q_z}{PQ} = C_s \\
2 \mathbf{s}_y = \frac{-s_x \cdot q_x + s_z \cdot q_z}{l} = s_y \\
2 \mathbf{s}_z = \frac{-C_s \cdot q_x + PQ \cdot s_x}{l}

Because also the expression for \( s_z \) will occur frequently in following expressions we have assigned it the special symbol \( C_s = s_z \).

Transformation IV – third rotation of \( q, r, \) and \( s \) around X-axis into \( q, \tilde{r}, \) and \( \tilde{s} \) such that the rotated line piece \( \tilde{r}, \tilde{s} \) runs parallel to the ZX-plane, thus with \( \tilde{r} \) and \( \tilde{s} \) having equal y-coordinates.

With the original line piece \( PQ \) transformed to coincide with the X-axis, the final rotation will be around the X-axis in order to position the original line piece \( RS \) parallel to the ZX-plane (Figure 6).

The rotation around the X-axis proceeds via \( \mathbf{r} = \text{Rot}_x(\alpha) \cdot \mathbf{r} \) and \( \mathbf{s} = \text{Rot}_x(\alpha) \cdot \mathbf{s} \) or

\[
\begin{pmatrix}
2 \mathbf{r}_x \\
2 \mathbf{r}_y \\
2 \mathbf{r}_z
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{pmatrix}
\begin{pmatrix}
2 \mathbf{r}_x \\
2 \mathbf{r}_y \\
2 \mathbf{r}_z
\end{pmatrix}
\quad \text{and}
\begin{pmatrix}
2 \mathbf{s}_x \\
2 \mathbf{s}_y \\
2 \mathbf{s}_z
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{pmatrix}
\begin{pmatrix}
2 \mathbf{s}_x \\
2 \mathbf{s}_y \\
2 \mathbf{s}_z
\end{pmatrix}
\tag{17}
\]

\[
\begin{align*}
\mathbf{f}_v &= f(V | r', s') = \frac{l_{\mathbf{r}', \mathbf{s}'}}{l_{V, \mathbf{r}', \mathbf{s}'}} = \frac{l_{\mathbf{r}', \mathbf{s}'} - l_{\mathbf{r}, \mathbf{s}} + l_{\mathbf{r}, \mathbf{s}'} - l_{\mathbf{r}', \mathbf{s}'}}{2l_{\mathbf{r}', \mathbf{s}'} + l_{\mathbf{r}', \mathbf{s}'}} = \frac{1}{2} \frac{l_{\mathbf{r}', \mathbf{s}'}}{l_{\mathbf{r}', \mathbf{s}'}} + \frac{1}{2}
\end{align*}
\]

with

\[
\begin{align*}
l_{\mathbf{r}', \mathbf{s}'} &= l_{\mathbf{r}', \mathbf{s}'}^2 + l_{\mathbf{r}'}^2 \\
l_{\mathbf{r}, \mathbf{s}} &= l_{\mathbf{r}, \mathbf{s}}^2 + l_{\mathbf{s}}^2 \\
l_{\mathbf{r}', \mathbf{s}'} &= (z_{\mathbf{r}'} - z_{\mathbf{s}})^2 + (z_{\mathbf{r}'} - z_{\mathbf{s}})^2
\end{align*}
\tag{18}
\]

the squared length of the projections of the respective line pieces in the YZ-plane. With the fraction \( f_v \) the \( y \)- and \( z \)-coordinates of point \( V \) can be calculated as

\[
\begin{align*}
V_x &= r_x' + f_v(z_{\mathbf{s}'} - z_{\mathbf{r}'}) = r_x + f_v(z_{\mathbf{s}} - z_{\mathbf{r}}) \\
V_z &= r_z' + f_v(z_{\mathbf{s}'} - z_{\mathbf{r}'}) = r_z + f_v(z_{\mathbf{s}} - z_{\mathbf{s}})
\end{align*}
\tag{19}
\]

When both line pieces \( PQ \) and \( RS \) are in one plane and are running parallel, then line piece \( r, s \) runs parallel to the X-axis, the projection points \( \tilde{r}, \tilde{s} \) and \( r, s \) coincide and the length \( l_{\tilde{r}, \tilde{s}} \) is zero. Then the coordinates \( \tilde{V}_x \) and \( \tilde{V}_z \) coincide with those of \( \tilde{r}' \) and \( \tilde{s}' \)

\[
\begin{align*}
\tilde{V}_x &= r_x' = s_x' \\
\tilde{V}_z &= r_z' = s_z'
\end{align*}
\tag{20}
\]

With the length of the line piece \( OV(l_{O,V}) \) the orientation angle \( \delta \) of line piece \( OV \) is obtained as

\[
\sin \delta = \frac{V_z}{\sqrt{V_y^2 + V_z^2}} = \frac{V_z}{l_{O,V}}, \quad \text{and \ cos} \delta = \frac{V_y}{\sqrt{V_x^2 + V_z^2}} = \frac{V_y}{l_{O,V}}
\tag{21}
\]

Concerning the rotation angle \( \alpha \), vector \( OV \) needs to be rotated toward the positive \( Y \)-axis implicating a rotation over \( -\delta \), thus

\[
\begin{align*}
\sin \alpha &= \sin \delta = -\sin \delta = -\frac{V_z}{l_{O,V}} \\
\cos \alpha &= \cos \delta = \frac{V_y}{l_{O,V}}
\end{align*}
\tag{22}
\]

The rotations in Eq. 17, \( \mathbf{r} = \text{Rot}_x(\alpha) \cdot \mathbf{r} \) and \( \mathbf{s} = \text{Rot}_x(\alpha) \cdot \mathbf{s} \) can now be made explicit as
\[ r_s = r_r = C_r, \]
\[ s_{\mathbf{Z}} = s_{\mathbf{Y}} = \cos \alpha - s_{\mathbf{X}} \cdot \sin \alpha = \frac{1}{l_{\mathbf{O}_{\mathbf{V}}}} (s_{\mathbf{z}} \cdot V_r + s_{\mathbf{z}} \cdot V_s) = V_r \left[ -s_{\mathbf{x}} \cdot q_s + s_{\mathbf{y}} \cdot q_s - s_{\mathbf{z}} \cdot q_s - \mathbf{C}_r \cdot q_s + \mathbf{PQ} \cdot r_s \right] \]

and similarly
\[ s_{\mathbf{x}} = s_{\mathbf{Y}} = \sin \alpha + s_{\mathbf{Z}} \cdot \cos \alpha = \frac{1}{l_{\mathbf{O}_{\mathbf{V}}}} (s_{\mathbf{y}} \cdot -V_r + s_{\mathbf{y}} \cdot V_s) = -V_s \left[ -s_{\mathbf{z}} \cdot q_s + s_{\mathbf{x}} \cdot q_s + s_{\mathbf{z}} \cdot q_s + \mathbf{C}_r \cdot q_s + \mathbf{PQ} \cdot s_s \right] \]

By the rotation the points \( r \) and \( s \) have obtained equal \( y \)-coordinates.

**Line pieces \( PQ \) and \( RS \) in one plane.** When both line pieces \( PQ \) and \( RS \) are in one plane but not parallel they (or their elongations) have an intersection point, the line piece \( r', s' \) has an orientation through the origin (i.e., the line piece \( r', s' \) or its elongation intersects the \( X \)-axis), and the length of line piece \( O\mathbf{V} \) is equal to zero. Then the projections of both points have equal ratios for their \( Z \)- and \( Y \)-coordinates

\[ \frac{s_{\mathbf{Z}'}}{s_{\mathbf{Y}'}} = \frac{s_{\mathbf{Z}'}}{s_{\mathbf{Y}'}} = \frac{s_{\mathbf{A}_1}}{s_{\mathbf{A}_2}} \]

The orientation of the perpendicular to line piece \( r', s' \) is then given by

\[ \sin(90 + \delta) = \cos \delta = \frac{r_{\mathbf{Y}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

\[ \cos(90 + \delta) = -\sin \delta = \frac{s_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

such that

\[ \sin \alpha = \sin -\delta = \sin \delta = \frac{r_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

\[ \cos \alpha = \cos -\delta = \cos \delta = \frac{r_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

Thus for non-parallel line pieces in one plane, the rotations in Eq. 17 \( r = \text{Rot}_x (\alpha) \) and \( s = \text{Rot}_s (\alpha) \) become

\[ r_{\mathbf{X}'} = r_{\mathbf{X}} = C_r \]

\[ s_{\mathbf{Y}'} = s_{\mathbf{Y}} = \cos \alpha - s_{\mathbf{Z}} \cdot \sin \alpha = \frac{r_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = 0 \]

\[ r_{\mathbf{Z}'} = r_{\mathbf{Z}} \cdot \sin \alpha + s_{\mathbf{Z}} \cdot \cos \alpha = \frac{r_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

and similarly

\[ s_{\mathbf{X}'} = s_{\mathbf{X}} = C_s \]

\[ s_{\mathbf{Y}'} = s_{\mathbf{Y}} = \cos \alpha - s_{\mathbf{Z}} \cdot \sin \alpha = \frac{s_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = 0 \]

\[ s_{\mathbf{Z}'} = s_{\mathbf{Z}} \cdot \sin \alpha + s_{\mathbf{Z}} \cdot \cos \alpha = \frac{s_{\mathbf{X}'}}{l_{\mathbf{O}_{\mathbf{V}}}} = \frac{-s_{\mathbf{Z}'}}{l_{\mathbf{O}_{\mathbf{V}}}} \]

**Applying the conditions for crossing line pieces**

After the series of transformations the line pieces \( O,q \) and \( r,s \) are in the geometrical positions of \( O,q \) coinciding with the \( X \)-axis, and \( r,s \) running parallel to the \( ZX \)-plane (Figure 7).

Now it can easily be determined whether they are crossing. With \( r,s \) running parallel to the \( ZX \)-plane, its projection \( r', s' \) onto the \( XY \)-plane runs parallel to the \( X \)-axis. If (A) \( r', s \) intersects the \( XY \)-plane, and (B) if this intersection point \( u \) of \( r,s \) with the \( XY \)-plane has an \( x \)-coordinate within the length of \( p,q \) \((p_1 \leq u \leq q_1)\) then \( O,q \) and \( r,s \) are crossing line pieces and the perpendicular line of \( u \) onto \( X \)-axis \( u \) is the orthogonal shortest distance between the line pieces.

**Condition A:** Line piece \( r,s \) crosses the \( XY \)-plane if the product of the \( z \)-coordinates of \( r \) and \( s \) is negative or if one or both of them equals zero, thus if

\[ r_{\mathbf{Z}} \cdot s_{\mathbf{Z}} \leq 0 \]
If both coordinates are zero, line piece $r_s$ runs parallel to $O_xq$ and is in the XY-plane. If one of the coordinates is zero, line piece $r_s$ has one point in the XY-plane.

**Condition B** requires first (B1) an overlap of x-coordinates of $O_xq$ and $r_s$, thus when

\[ s_x > 0 \text{ and } s_x < q_x, \text{ when } t_x > s_x \]

or

\[ s_x < 0 \text{ and } t_x < q_x, \text{ when } s_x > t_x \]  \hspace{1cm} (30)

As $O_xq$ originates from the transformed line piece PQ its length equals that of PQ, thus $q_x = PQ$. Using the symbols $C_{xq}$ and $C_{x}$, Eq. 30 can also be phrased as

\[ C_{x} > 0 \text{ and } C_{x} < PQ, \text{ when } C_{x} > C_{x} \]

or

\[ C_{x} > 0 \text{ and } C_{x} < PQ, \text{ when } C_{x} > C_{x} \]  \hspace{1cm} (30a)

If both conditions (A) and (B1) are fulfilled the perpendicular distance from intersection point $u$ of $r_s$ with the XY-plane to the point $t$ on the line piece $O_xq$ can be determined as

\[ u_s - s_s = \frac{s_s}{s_s} \]

with

\[ u_s = \frac{y_s - s_s - C_{x} - s_s}{s_s - s_s} \text{ or } s_s = 0, \text{ a choice is made for the position of the orthogonal connection line with} \]

\[ u_s = \frac{\min(s_y, q_y) - \max(0, s_y)}{2} = \frac{\min(C_{x}, PQ) - \max(0, C_{x})}{2} \text{ or } s_s = 0, \text{ a choice is made for the position of the orthogonal connection line with} \]

\[ u_s = \frac{\min(s_y, q_y) - \max(0, s_y)}{2} = \frac{\min(C_{x}, PQ) - \max(0, C_{x})}{2} \text{ when } s_s > r_s (C_{x} > C_{x}). \]

The final condition (B2) requires that the intersection point is within the length of $O_xq$, thus

\[ s_p < u_s < q_x, \text{ or } 0 < u_s < PQ \]

The fractions of the line pieces where the orthogonal connection intersects the line pieces is independent of the transformations applied and therefore we can derive the crossing point $T$ on PQ to be at a fraction

\[ f(T \mid PQ) = \frac{t_s}{q_s} = \frac{t_s}{PQ} \]  \hspace{1cm} (35)

on the line piece $PQ$ and the crossing point $U$ on $RS$ to be at a fraction of

\[ f(U \mid RS) = \frac{u_s - r_s}{r_s - s_s} = \frac{u_s - C_{x}}{C_{x} - C_{x}} \]  \hspace{1cm} (36)

of the line piece $RS$. The orthogonal distance between two crossing line pieces thus is between the point $T(x, y, z)$ on $PQ$

\[ T_x = P_x + f(T \mid PQ) \cdot (Q_x - P_x) \]

\[ T_y = P_y + f(T \mid PQ) \cdot (Q_y - P_y) \]

\[ T_z = P_z + f(T \mid PQ) \cdot (Q_z - P_z) \]  \hspace{1cm} (37)

and the point $U(x, y, z)$ on $RS$

\[ U_x = R_x + f(U \mid RS) \cdot (S_x - R_x) \]

\[ U_y = R_y + f(U \mid RS) \cdot (S_y - R_y) \]

\[ U_z = R_z + f(U \mid RS) \cdot (S_z - R_z) \]  \hspace{1cm} (38)

and is equal to

\[ TU = \sqrt{(T_x - U_x)^2 + (T_y - U_y)^2 + (T_z - U_z)^2} \]  \hspace{1cm} (39)

In the final step the orthogonal distance $TU$ is tested against a given proximity criterion value.

**ALTERNATIVE APPROACH FOR TESTING CROSSING OF LINE PIECES**

The approach followed in this paper was first to test the crossing property of line pieces, followed by the calculation of the orthogonal distance. An alternative approach was introduced by Dan Sunday in his algorithm on the softsurfer.com website (http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm). This algorithm first determines the crossing point of the infinite lines through the line pieces, followed by a test of whether or not the orthogonal connection line intersects the line pieces themselves and if so, followed by the distance calculation. The procedure of Sunday is described in “Analytical Approach” of Appendix and includes an algorithm for calculating the shortest distance between two non-crossing line pieces, i.e., when the line piece extensions appear to cross outside the range of the line pieces.

**RESULTS**

The “old“ synapse formation model based on distance only and the “new“ one based on crossing and distance have been applied to a network of 25 Layer 2/3 pyramidal neurons, with their cell bodies generated within a sphere of radius 43 μm with minimum neuron separation of 20 μm. The neurons were generated using the NETMORPH simulator (Koene et al., 2009) based on growth parameter values optimized on a data set of reconstructed rat cerebral cortex L2/3 pyramidal neurons made available by Svoboda through the www.neuromorpho.org website (Ascoli et al., 2007). Examples are illustrated in Figure 8.

Outcomes of the “old“ and “new“ synapse formation model are illustrated in Figures 9A–D. The figures show the axons (green), dendrites (red) and the synapse locations (small blue spheres). Figures 9A,B show the locations of candidate synapses when the crossing condition was not applied and only the shortest distance
van Pelt et al. Algorithm for proximity-based connectivity

between line pieces was tested, using the algorithm based on Sunday’s approach (Shortest Distance Between Two Non-Crossing Line Pieces). Figures 9C, D show the locations of candidate synapses when the crossing condition was applied and the orthogonal connection was subjected to the distance criterion. In all four panels the distance criterion was 4 μm. The results in Figures 9A, C were obtained with a time step parameter of 20000 s, and those in Figures 9B, D with a time step parameter of 20 s. The number of synapses formed is 32799 in Figure 9A, 111104 in Figure 9B, 1188 in Figure 9C, and 1555 in Figure 9D, respectively. Comparing these numbers as well as the pictures in the top and bottom row makes clear how the distance-based approach (top row) results in a strong clustering of synapses and how this is prevented by applying the crossing condition (bottom row). Comparing the synapse numbers as well as the pictures for “long” (left column) and “short” line pieces (right column) illustrates that the synapse formation model based on distance only is highly
van Pelt et al. Algorithm for proximity-based connectivity

(i.e., synapses were formed when an axon and a dendrite came within 2 μm of each other, a distance criterion also used by Stepanyants and Chklovskii, 2005). The new synapse formation model has been applied with distance criteria of 2, 4, and 6 μm, respectively. The results of these simulations are shown in Figure 10, summarizing the distributions of the number of synaptic contacts per connection.

The left column of Figure 10 shows the distribution of the number of contacts/connection versus frequency for a synapse formation model based only on distance (see Shortest Distance Between Two Non-Crossing Line Pieces). The distance-based synapse formation model (first column) makes the number of contacts highly sensitive to the length scale. The new synapse formation model (columns 2–4) strongly reduces this dependency.

With the new synapse formation model, a first series of simulations have been performed on the number of synaptic contacts per connection between neuron pairs. To this end networks were generated with the NETMORPH simulator of 250 layer 2/3 pyramidal neurons with their cell bodies arranged in a sphere with realistic density of 75,000 mm⁻³ (Hellwig, 2000). The radius of the sphere is 93 μm. The minimum separation between somata is 20 μm. Both the old synapse formation model based on distance only, as well as the new one based on crossing and distance, were used. For the old model a distance criterion of 2 μm was used (i.e., synapses were formed when an axon and a dendrite came within 2 μm of each other, a distance criterion also used by Stepanyants and Chklovskii, 2005). The new synapse formation model has been applied with distance criteria of 2, 4, and 6 μm, respectively. The results of these simulations are shown in Figure 10, summarizing the distributions of the number of synaptic contacts per connection.

The distance-based synapse formation model (first column) makes the number of contacts highly sensitive to this length scale. The new synapse formation model (columns 2–4) strongly reduces this dependency.

FIGURE 10 | Distributions of the number of contacts per connection. Synaptic connectivity in a simulated NETMORPH network of 250 layer 2/3 pyramidal neurons with their cell bodies arranged in a sphere with realistic density of 75,000 mm⁻³ (Hellwig, 2000). The radius of the sphere is 93 μm. The distance-based synapse formation model (first column) makes the number of contacts highly sensitive to this length scale. The new synapse formation model (columns 2–4) strongly reduces this dependency.
formation model based on distance and crossing. The 2nd column was obtained with the same distance criterion of 2 μm as the 1st column and comparing these columns shows how the additional crossing criterion reduces the number of contacts per connections to realistic values by eliminating the clustering effect. The third and fourth column are obtained for larger values of the distance criterion. As expected the number of contacts per connection increases with this criterion as the likelihood for sufficient proximity also increases, with mn(sd) values of 2.0(1.8), 3.4(3.9), and 5.1(6.5) contacts per connection for distance criteria of 2, 4, and 6 μm, respectively, and for using a time step of df = 20 s (top row).

Experimental data for the number of synaptic contacts per connection show an average value of mn(sd) 2.8(0.7) (n = 8 pairs) between L2/3 pyramidal neurons in the barrel cortex of juvenile rats (Feldmeyer et al., 2006), an average value of 4.5(0.5) (n = 13 pairs) between juvenile rat barrel cortex L4 spiny neurons and L2/3 pyramidal cells (Feldmeyer et al., 2002), an average of 3.4(1.0) (n = 11 pairs) between spiny rat somatosensory cortex L4 neurons (Feldmeyer et al., 1999), and an average number of 3.5 ± 1.8 putative contacts per connection between juvenile rat somatosensory cortex L5A pyramidal cells (Frick et al., 2008). Hellwig (2000) estimated the number of synapses per connection from the overlap of voxelized images for nearby neurons to be about 2.5 for L2–L2 and in computer generated morphologies from neural simulations where they come sufficiently close to each other, typically within a distance determined by the length of dendritic spines and axonal boutons. The present outcomes overlap so nicely the range of experimental data this prediction is still waiting validation. The present outcomes are also in good agreement with these estimated numbers.

An additional outcome of our present simulations is that the frequency distributions are monotone decreasing with the number of contacts per connection, a property that is not yet experimentally observed due to the low numbers in the published distributions. The monotone decreasing distribution may not be surprising as the probability of having two contacts per connection is expected to be the square of the probability of having one contact per neuron pair.

**DISCUSSION**

Axons and dendrites can make synaptic connections at those locations where they come sufficiently close to each other, typically within a distance determined by the length of dendritic spines and axonal boutons. The present study aimed at defining a procedure for finding such locations in networks of neurons whose 3D geometry is represented by piecewise-linear structures (line pieces or cylinders) as is the case in experimental reconstructions and in computer generated morphologies from neural simulators. Proximity of axonal and dendritic branches is then tested by evaluating proximity of all pairs of line pieces that constitute both neuronal structures. Note that proximity is here taken between the axial center lines and that a criterion of proximity should consider both the branch diameters and the distance between the branch exteriors (such as spine and bouton lengths).

The disadvantage of a straightforward distance criterion is that it may be met by several combinations of nearby line pieces, resulting in clusters of candidate synaptic sites. Additionally, this number appeared to be sensitive to the length scale of the individual line pieces. The procedure introduced in this study aimed at representing the places where axonal and dendritic branches are sufficiently close by a single candidate synaptic site. To this end the line piece crossing criterion was introduced in addition to the distance criterion. This was based on the notion that the shortest distance between two infinite lines in 3D is determined by the length of the orthogonal connection line at the unique site of crossing of the infinite lines. Also two chains of line pieces will generally contain only one unique pair of mutual line pieces that fulfills the geometrical condition of crossing and identifies the unique location of shortest distance between the two chains of line pieces. For this reason, the crossing condition was introduced in addition to the distance criterion to eliminate the clustering artifact and to reduce the length scale dependency.

The paper describes the derivation of the crossing conditions by applying a series of 3D transformations to the line piece pair to bring them in such a 3D position that crossing can easily be determined. The conditions for crossing are finally summarized in a series of tests to be applied to all dendritic and axonal line piece pairs of two neuronal representations. The performance of the “new” synapse detection method has been illustrated visually in tests of the “old” and the “new” method, and quantitatively in terms of distributions of the number of contacts between connected neuron pairs. The obtained mean number of contacts per connection appeared to be in excellent agreement with the available experimental data. The frequency distributions for the number of contacts per connection had a monotone decreasing shape. Because of the small number of observations in experimental data this prediction is still waiting validation. The present outcomes clearly demonstrated that the new synapse detection model based on crossing and distance eliminated the clustering effect of the old model based on distance only, and reduced significantly the dependence on the length scale of the line pieces.

Although two chains of line pieces will generally contain only one unique pair of crossing line pieces, more than one crossing pair may occur under certain geometrical conditions. For instance, when a certain line piece A1 of chain A is orthogonal and intersects the plane formed by two subsequent line pieces B1 and B2 of another chain B, then both B1 and B2 may cross the A1 line piece. If so, and if the orthogonal connection in both pairs is meeting the distance criterion, then they result in two nearby candidate sites. Another case is when two line piece chains are running more or less in parallel. Then, crossing line piece pairs may occur at varying distances along both chains and, when they meet the distance criterion, result in as many candidate synaptic sites.

Axonal and dendritic branches may follow straight courses but may also meander strongly. Deviations from a straight line influence the way axons and dendrites are filling space. The probability of sites of near proximity is therefore expected to depend on the tortuosity of axonal and dendritic branches. It is an interesting question...
whether in real nervous tissue strongly meandering axons and dendrites make relative more synapses in comparison with straight ones. In our computer generated networks this is certainly expected when the meandering of the branches is approximated by line pieces with varying orientations. In a given area in space one may expect more nearby line piece pairs to cross and meeting the distance criterion at small length scales than at large length scales of the line pieces. It is thus not surprising that a remaining length scale dependency was also found with the new synapse formation model.

The algorithm for testing crossing line pieces derived in this paper differs from the one of Dan Sunday (Analytical Approach) by first testing whether or not two line pieces are crossing, and if so, followed by the distance calculation. The algorithm of Sunday, in contrast, first calculates the site where the infinite lines through the line pieces cross, followed by a test of whether or not this site of crossing coincides with the line pieces and if so, also followed by the distance calculation. This difference makes the present algorithm about 20% faster than Sunday’s algorithm, when implemented in FORTRAN code on a Linux system.

The procedure described in this paper identifies the location where an axonal and a dendritic branch are sufficiently close by a single candidate synaptic site, and is purely based on geometrical considerations. Whether or not in the biological tissue at these sites synapses will actually form and in which number is a question not addressed in this paper. Dendrites may be covered by spines in high density suggesting that a nearby axon may form more than one synapse close in space. Such a clustering of synapses between a single axon and dendrite then has a biological origin and is thus principally different from the artifactual clustering caused by the search procedure that gave rise to this study. Nearby spines may of course also be contacted by axonal branches from different neurons. When they all meet the crossing and distance criteria, a clustering of candidate synaptic locations between a single dendritic branch and these different axons will then still occur.

ACKNOWLEDGMENTS

This work was supported by EU BIO-ICT Project SECO (216593), and by NWO-CLS Project NETFORM (635.100.107).
APPENDIX

ROTATIONS IN 3D

Figure A1.1 illustrates the used symbols and the clockwise directions of the rotations around the axes. For the derivation of the rotation matrix with, for instance, rotation over angle alpha around the X-axis (see Figure A1.2) we obtain:

\[ A_y' = A_y \]  
\[ A_x' = A_x \cos(\alpha + \delta) = AO \cdot \left[ \cos(\alpha \cos \delta - \sin \alpha \sin \delta) \right] 
\]
\[ = A_y \cos \alpha - A_z \sin \alpha \]
\[ A_z' = A_z \cos(\alpha + \delta) = AO \cdot \left[ \sin(\alpha \cos \delta + \sin \delta \cos \alpha) \right] 
\]
\[ = A_y \sin \alpha + A_z \cos \alpha \]

In matrix notation

\[
\begin{pmatrix}
A_x' \\
A_y' \\
A_z'
\end{pmatrix} =
\begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{pmatrix}
\begin{pmatrix}
A_x \\
A_y \\
A_z
\end{pmatrix}
\]

Thus the rotation matrix for a rotation around the X-axis becomes

\[
\text{Rot}_x(\alpha) =
\begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{pmatrix}
\]

Figure A1.2 | Rotation of line piece AO around X-axis into A'O.

Similarly for rotations around the Y-axis and Z-axis we obtain

\[
\text{Rot}_y(y) =
\begin{pmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta
\end{pmatrix}
\]

\[
\text{Rot}_z(\gamma) =
\begin{pmatrix}
\cos \gamma & -\sin \gamma & 0 \\
\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

The 3D coordinates of point D can now directly be obtained from the coordinates of A and B as

\[
D_x = A_x + f(D \perp AB) \cdot (B_x - A_x) \\
D_y = A_y + f(D \perp AB) \cdot (B_y - A_y) \\
D_z = A_z + f(D \perp AB) \cdot (B_z - A_z)
\]

or

\[
D_x = B_x + f(D \perp BA) \cdot (A_x - B_x) \\
D_y = B_y + f(D \perp BA) \cdot (A_y - B_y) \\
D_z = B_z + f(D \perp BA) \cdot (A_z - B_z)
\]

If points A and B coincide then point D adopts their coordinates and thus

\[
D_x = A_x = B_x, \quad D_y = A_y = B_y, \quad D_z = A_z = B_z
\]

In the case of skewed triangles (lowest two triangles in Figure A2), the orthogonal lines CD on AB do not end any more on the line piece AB itself and the relative positions \(f(D \perp AB)\) and \(f(D \perp BA)\) can get values outside the range of [0, 1]. The relative position of the perpendicular ending in a triangle was used in Figure 6B to obtain the position of point V (see also Eq. 18). In a skewed triangle this point V would not be within the line piece \(\mathcal{r}_5\) in Figure 6B, implicating that after rotation line piece \(\mathcal{r}_5\) does not intersect the XY-plane. Thus a skewed triangle already indicates that the line pieces are not crossing.

RELATIVE POSITION OF POINT D OF THE PERPENDICULAR FROM POINT C TO OPPOSITE SIDE AB IN A TRIANGLE ABC

We have

\[
c_i = b \cdot \cos \alpha, \quad c_i = a \cdot \cos \beta
\]

and

\[
\cos \alpha = b^2 + c^2 - a^2 / 2bc, \quad \cos \beta = a^2 + c^2 - b^2 / 2ac
\]

thus

\[
c_1 = b^2 + c^2 - a^2 / 2c, \quad c_2 = a^2 + c^2 - b^2 / 2c
\]

For the relative position of D on AB we obtain

\[
f(D \perp AB) \equiv c \equiv \frac{b^2 - a^2 + c^2}{2c^2}
\]

\[
f(D \perp BA) = \frac{c_2}{c} = \frac{a^2 - b^2 + c^2}{2c^2}
\]

EXPRESSING THE CROSSING CONDITIONS IN TERMS OF THE ORIGINAL COORDINATES

Some crossing conditions as described in the previous section still need to be expressed in terms of the coordinates of the original points P, Q, R, and S, or their relative coordinates, \(p, q, r, s\)
after their translations. In particular the quantities \(f_v\), \(V_y\), and \(V_z\) need further elaboration.

Equations 18 and 19 give us the expressions for \(f_v\), \(V_y\), and \(V_z\), respectively

\[
f_v = \frac{1}{2} \frac{F_{\text{tr}}^t}{F_{\text{tr}}^*} - \frac{1}{2} \frac{F_{\text{tr}}^*}{F_{\text{tr}}^t}, \quad V_y = \frac{1}{2} \left( y_s - r_y \right), \quad V_z = \frac{1}{2} \left( z_s - r_z \right)
\]

Let us define

\[
c_{y} = -r_y \cdot q_y + r_y \cdot s_y, \quad c_{x} = -C_{k} \cdot q_x + P_{Q} \cdot r_x,
\]

\[
c_{y} = -s_y \cdot q_y + s_y \cdot q_z, \quad c_{x} = -C_{k} \cdot q_x + P_{Q} \cdot s_z,
\]

\[
c_{x} = -C_{k} \cdot q_x + P_{Q} \cdot s_z
\]

then we obtain

\[
\begin{align*}
\frac{\dot{r}_y}{l} &= \frac{c_x}{l}, \quad \frac{\dot{r}_z}{l} = \frac{c_z}{l}, \quad \frac{\dot{s}_y}{l} = \frac{c_s}{l}, \quad \frac{\dot{s}_z}{l} = \frac{c_{s_x}}{l}
\end{align*}
\]

and

\[
\frac{F_{\text{tr}}^t}{F_{\text{tr}}^*} = r_y + \frac{c_x}{l} \quad \frac{F_{\text{tr}}^*}{F_{\text{tr}}^t} = r_z + \frac{c_x}{l}
\]

\[
\frac{F_{\text{tr}}^t}{F_{\text{tr}}^*} = s_y + \frac{c_s}{l} \quad \frac{F_{\text{tr}}^*}{F_{\text{tr}}^t} = s_z + \frac{c_{s_x}}{l}
\]

\[
\frac{F_{\text{tr}}^t}{F_{\text{tr}}^*} = (s_y - s_z)^2 + (s_z - r_y)^2 = \left( c_x - c_{s_x} \right)^2 + \left( c_s - c_{s_x} \right)^2
\]

such that we obtain for \(f_v\), \(V_y\), and \(V_z\)

\[
\begin{align*}
f_v &= \frac{1}{2} \frac{F_{\text{tr}}^t}{F_{\text{tr}}^*} - \frac{1}{2} \frac{F_{\text{tr}}^*}{F_{\text{tr}}^t} + \frac{1}{2} \left( c_x + c_{s_x} - c_x - c_{s_x} \right) + \frac{1}{2} \\
V_y &= \frac{1}{2} \left( r_y + f_v \cdot (s_y - r_y) \right) = \frac{1}{l} \left( c_y + f_v \cdot (c_y - c_{y}) \right) + \frac{1}{l} \left( c_y + f_v \cdot (c_y - c_{y}) \right) + \frac{1}{l} \\
V_z &= \frac{1}{2} \left( r_z + f_v \cdot (s_z - r_z) \right) = \frac{1}{l} \left( c_z + f_v \cdot (c_z - c_{z}) \right) + \frac{1}{l} \left( c_z + f_v \cdot (c_z - c_{z}) \right) + \frac{1}{l}
\end{align*}
\]

and for the ratio \(V/V_z\)

\[
\frac{V_y}{V_z} = \frac{c_y + f_v \cdot (c_y - c_{y})}{c_z + f_v \cdot (c_z - c_{z})}
\]

\[
\frac{V_z}{V_y} = \frac{c_z + f_v \cdot (c_z - c_{z})}{c_y + f_v \cdot (c_y - c_{y})}
\]

\[\text{(A3-5)}\]

Next, the quantities \(r_y, s_y, r_z, s_z\) can be expressed in terms of the coordinates of the original points \(P, Q, R,\) and \(S\), with the subsequent calculation of \(u_x\). Equations 23 and 24 give us the expressions for these quantities as

\[
\begin{align*}
s_y &= C_{s_y}, \quad s_z = C_{s_z}, \quad s_
\end{align*}
\]

and for the denominator in the expression for \(u_x\) (Eq. 31) we obtain

\[
\begin{align*}
s_y &= \frac{-V_y \left[ -r_y \cdot q_y + r_y \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_y + C_{k} \cdot P_{Q} \cdot r_y \right]}{1 \cdot l^{l_{QW}}} \\
s_z &= \frac{-V_y \left[ -r_z \cdot q_z + r_z \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_z + C_{k} \cdot P_{Q} \cdot s_z \right]}{1 \cdot l^{l_{QW}}}
\end{align*}
\]

For the numerator in the expression for \(u_x\) (Eq. 31) we obtain

\[
\begin{align*}
s_y &= \frac{-V_y \left[ -r_y \cdot q_y + r_y \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_y + C_{k} \cdot P_{Q} \cdot r_y \right]}{1 \cdot l^{l_{QW}}} \\
s_z &= \frac{-V_y \left[ -r_z \cdot q_z + r_z \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_z + C_{k} \cdot P_{Q} \cdot s_z \right]}{1 \cdot l^{l_{QW}}}
\end{align*}
\]

and for the denominator in the expression for \(u_x\) (Eq. 31)

\[
\begin{align*}
s_y &= \frac{-V_y \left[ -r_y \cdot q_y + r_y \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_y + C_{k} \cdot P_{Q} \cdot r_y \right]}{1 \cdot l^{l_{QW}}} \\
s_z &= \frac{-V_y \left[ -r_z \cdot q_z + r_z \cdot q_s \right] + V_y \left[ -C_{k} \cdot q_z + C_{k} \cdot P_{Q} \cdot s_z \right]}{1 \cdot l^{l_{QW}}}
\end{align*}
\]

Now we obtain for the expression of \(u_x\) (Eq. 31)
\[
u_s = \frac{C_k \cdot r_s - C_r \cdot s_s}{s r_s - s s_s} = \frac{A}{B}
\]

with
\[
A = q_s \cdot (-C_s \cdot r_s + C_r \cdot s_s) + q_r \cdot (C_s \cdot r_r - C_r \cdot s_r) +
\]
\[
-\frac{V_{v_s}}{V_z} \cdot P_C \cdot (C_s \cdot r_s - C_r \cdot s_s)
\]
\[
B = q_s \cdot (r_s - s_s) - q_r \cdot (r_r - s_r) +
\]
\[
-\frac{V_{v_s}}{V_z} \cdot \left[ q_s \cdot (C_s - C_r) + P_Q \cdot (r_s - s_s) \right]
\]

(A3-6)

with (Eqs. 13 and 16)
\[
C_k = \frac{r_s \cdot q_s + r_r \cdot q_r + r_s \cdot q_r}{P_Q}, \quad C_s = \frac{s_s \cdot q_s + s_r \cdot q_r + s_s \cdot q_s}{P_Q},
\]

and via Eq. 4 \( P_Q = \sqrt{q_s^2 + q_r^2 + q_z^2} \)

**SUMMARY OF CALCULATIONS AND CROSSING CONDITIONS**

Given: the coordinates of the points \( P(x, y, z) \), \( Q(x, y, z) \), \( R(x, y, z) \) and \( S(x, y, z) \)

1. **Translation of \( P \) to origin:**
\[
q_s = Q_s - P_x, \quad q_r = Q_r - P_y, \quad q_s = Q_s - P_x
\]
\[
r_s = R_s - P_x, \quad r_r = R_r - P_y, \quad r_s = R_s - P_x
\]
\[
s_s = S_s - P_x, \quad s_r = S_r - P_y, \quad s_s = S_s - P_x
\]

(A4-1)

2. **Constants:**
\[
q \cdot q = q_s \cdot q_s + q_r \cdot q_r + q_z \cdot q_z
\]
\[
r \cdot r = r_s \cdot r_s + r_r \cdot r_r + r_z \cdot r_z
\]
\[
s \cdot s = s_s \cdot s_s + s_r \cdot s_r + s_z \cdot s_z
\]

(A4-2)

3. **Test on crossing condition BI that after the rotations the line pieces should have at least overlapping \( x \)-coordinates:**
\[
P_Q = \sqrt{q \cdot q}, \quad C_k = \frac{r \cdot q}{P_Q}, \quad C_s = \frac{s \cdot q}{P_Q}
\]

If \( C_k > C_s \) then \( (C_k > 0 \text{ and } C_s < P_Q) \) (A4-3)

or

If \( C_k > C_s \) then \( (C_k > 0 \text{ and } C_s < P_Q) \)

If condition BI is not fulfilled then no crossing and stop.

Note: because the length \( P_Q \) is a positive number one may also compare the quantities

\[
C_k' = C_k = \frac{r \cdot q}{P_Q} \quad q \cdot q \quad C_s' = C_s = \frac{s \cdot q}{P_Q} \quad q \cdot q
\]

making condition BI equivalent to:

If \( C_k' > C_s' \) then \( (C_k' > 0 \text{ and } C_s' < 1) \)

(A4-4)

or

If \( C_k' > C_s' \) then \( (C_k' > 0 \text{ and } C_s' < 1) \)

As this BI test does not require a square root floating point operation (for \( P_Q \)) it generates faster computer code, which is relevant if most comparisons in actual circumstances do not meet condition BI. If condition BI is not fulfilled then no crossing and stop, else calculate \( P_Q \). If condition BI is fulfilled then proceed.

4. **More constants:**
\[
l = \sqrt{P_Q^2 - q_s^2}
\]
\[
c_{\eta} = -r_s \cdot q_r + r_r \cdot q_s
\]
\[
c_{\eta} = -C_k \cdot q_r + P_Q \cdot r_s
\]
\[
c_{\eta} = -C_k \cdot q_s + P_Q \cdot s_s
\]
\[
(A4-5)
\]

5. **Calculate coordinates \( r_s \) and \( s_s \)**

First test if \( P_Q \) and \( RS \) in one plane, i.e., when
\[
\frac{r_s'}{r_s'} = \frac{s_s'}{s_s'}, \quad \text{or} \quad \frac{r_s}{r_s} = \frac{s_s}{s_s}, \quad \text{thus if} \quad c_{\eta} = c_{\eta} \quad \text{or} \quad c_{\eta} = c_{\eta}
\]
\[
c_{\eta} \cdot c_{\eta} = c_{\eta} \cdot c_{\eta} \quad \text{(see Expressing the Crossing Conditions in Terms of the Original Coordinates)}
\]

but not running parallel, then coordinates \( r_s \) and \( s_s \) are given by
\[
r_s = -l_{O_z}, \quad s_s = -l_{O_z}
\]

with (see Expressing the Crossing Conditions in Terms of the Original Coordinates)
\[
l_{O_z} = \sqrt{c_{\eta}^2 + c_{\eta}^2}, \quad l_{O_z} = \sqrt{c_{\eta}^2 + c_{\eta}^2}
\]

(A4-8)

5b. If \( P_Q \) and \( RS \) in one plane, i.e., when
\[
\frac{r_s'}{r_s'} = \frac{r_s'}{s_s'}, \quad \text{or} \quad \frac{r_s}{r_s} = \frac{s_s}{s_s}, \quad \text{thus if} \quad c_{\eta} = c_{\eta} \quad \text{or} \quad c_{\eta} = c_{\eta}
\]
\[
c_{\eta} \cdot c_{\eta} = c_{\eta} \cdot c_{\eta} \quad \text{(see Expressing the Crossing Conditions in Terms of the Original Coordinates)}
\]

5c. If \( P_Q \) and \( RS \) not in one plane then coordinates \( r_s \) and \( s_s \) are obtained as follows

5c1. Ratio \( f_r \)
\[
f_r = \frac{1}{2} \left( \frac{c_{\eta}^2 + c_{\eta}^2 - c_{\eta}^2 - c_{\eta}^2}{c_{\eta} - c_{\eta}} \right)^2 + \frac{1}{2}
\]

(A4-9)

5c2. Coordinates and length of \( V \)
\[
V_y = \frac{1}{2} \left( c_{\eta} \cdot f_r \cdot (c_{\eta} - c_{\eta}) \right)
\]
\[
V_s = \frac{1}{2} \left( c_{\eta} \cdot f_r \cdot (c_{\eta} - c_{\eta}) \right)
\]
\[
l_{O_y} = \sqrt{V_y^2 + V_s^2}
\]

(A4-10)

5c3. Coordinates \( r_s \) and \( s_s \):
\[
r_s = \frac{V_y \cdot c_{\eta} + V_s \cdot c_{\eta}}{l_{O_y}}, \quad s_s = \frac{-V_y \cdot c_{\eta} + V_s \cdot c_{\eta}}{l_{O_y}}
\]

(A4-11)

6. **Test on crossing condition A** that after the rotations line piece \( s_r \cdot s_s \) intersects the \( XY \)-plane:
\[
(s_r') \cdot (s_s') \leq 0
\]

(A4-12)
If condition A is fulfilled (and condition B1 was already fulfilled – see step 3) then proceed with calculation of x-coordinate of the intersection point.

7. **Coordinate $u_x$**:
   
   7a. If both line pieces are running parallel, thus both $r_x = 0$ and $s_x = 0$, a choice is made for the position $u_x$ as
   
   $$
   u_x = \frac{(\min(C_y, PQ) - \max(0, C_y))}{2}, \text{ when } C_r > C_y \text{ or } C_r > C_y,
   $$
   
   or
   
   $$
   u_x = \frac{\left(\min(C_y, PQ) - \max(0, C_y)\right)}{2}, \text{ when } C_y > C_r
   $$
   
   (A4-13)
   
   7b. If both line pieces are not running parallel then
   
   $$
   u_x = \frac{C_x \cdot r_x - C_y \cdot s_x}{s_x - r_x}
   $$
   
   (A4-14)
   
8. Test on condition B2 that the intersection point is within the length of PQ
   
   $$
   0 \leq u_x \leq PQ
   $$
   
   (A4-15)
   
   If finally condition B2 is also fulfilled, then the line pieces are crossing and therefore proceed

9. **Ratios $f_x$ and $f_y$**:
   
   $$
   f_x = \frac{u_x}{PQ}, \text{ and } f_y = \frac{u_x - C_y}{C_y - C_y}
   $$
   
   (A4-16)

10. **Coordinates of the endpoints T and U of the orthogonal distance $TU$** between crossing line pieces PQ and RS, respectively:
   
   $$
   \begin{align*}
   T_x &= P_x + f_x \cdot (Q_x - P_x) \\
   T_y &= P_y + f_y \cdot (Q_y - P_y) \\
   T_z &= P_z + f_z \cdot (Q_z - P_z) \\
   U_x &= R_x + f_x \cdot (S_x - R_x) \\
   U_y &= R_y + f_y \cdot (S_y - R_y) \\
   U_z &= R_z + f_z \cdot (S_z - R_z)
   \end{align*}
   $$
   
   (A4-17)
   
   $$
   TU = \sqrt{(T_x - U_x)^2 + (T_y - U_y)^2 + (T_z - U_z)^2}
   $$

**ANALYTICAL APPROACH**

An alternative approach uses an algorithm taken from the site http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm by Dan Sunday.

**Shortest distance between two infinite lines**

In this analytical geometry approach the shortest distance between the two infinite lines through the line pieces PQ and RS is first been determined, followed by the test whether the orthogonal connection line intersects the line pieces PQ and RS, respectively.

The infinite line $A$ through the points $P$ and $Q$ can be described as

$$
A(s) = P + s \cdot \bar{u} \text{ with vector } \bar{u} = (Q_x - P_x) \quad (A5-1)
$$

and the infinite line $B$ through the points $R$ and $S$ as

$$
B(t) = R + t \cdot \bar{v} \text{ with vector } \bar{v} = (S_x - R_x)
$$

(A5-2)

Note that for $s \in [0, 1]$, the point $A(s)$ is positioned on the line piece PQ, and that for $t \in [0, 1]$, the point $B(t)$ is positioned on the line piece RS. Let vector $\bar{w}(s,t)$ connects a point $A(s)$ on line $A$ with a point $B(t)$ on line $B$ as

$$
\bar{w}(s,t) = A(s) - B(t) = P - R + s \bar{u} + t \bar{v}
$$

(A5-3)

with

$$
\bar{w}_0 = P - R
$$

(A5-4)

Lines $A$ and $B$ cross each other where the distance between points on both lines is the shortest, say at $A(s_c)$ and $B(t_c)$. The vector $\bar{w}$, between these points then is orthogonal to line $A$ and line $B$, and we obtain

$$
\bar{u} \cdot \bar{w}_0 = 0, \text{ and } \bar{v} \cdot \bar{w}_0 = 0
$$

(A5-5)

such that

$$
\bar{u} \cdot \bar{w}_0 + s_c (\bar{u} \cdot \bar{u}) - t_c (\bar{u} \cdot \bar{v}) = 0
$$

(A5-6)

and

$$
\bar{v} \cdot \bar{w}_0 + s_c (\bar{v} \cdot \bar{u}) - t_c (\bar{v} \cdot \bar{v}) = 0
$$

(A5-7)

Let us define:

$$
\begin{align*}
   a &= \bar{u} \cdot \bar{u} = (Q_x - P_x)^2 + (Q_y - P_y)^2 + (Q_z - P_z)^2 \\
   b &= \bar{u} \cdot \bar{v} = (Q_x - P_x) \cdot (S_x - R_x) \\
   &+ (Q_y - P_y) \cdot (S_y - R_y) + (Q_z - P_z) \cdot (S_z - R_z) \\
   c &= \bar{v} \cdot \bar{v} = (S_x - R_x)^2 + (S_y - R_y)^2 + (S_z - R_z)^2 \\
   d &= \bar{u} \cdot \bar{w}_0 = (Q_x - P_x) \cdot (P_x - R_x) \\
   &+ (Q_y - P_y) \cdot (P_y - R_y) + (Q_z - P_z) \cdot (P_z - R_z) \\
   e &= \bar{v} \cdot \bar{w}_0 = (S_x - R_x) \cdot (P_x - R_x) \\
   &+ (S_y - R_y) \cdot (P_y - R_y) + (S_z - R_z) \cdot (P_z - R_z)
   \end{align*}
$$

(A5-8)

then we obtain

$$
da + a_s \cdot b_t = 0 \text{ and } e + b_s \cdot c_t = 0
$$

(A5-9)

and their solutions as

$$
s_c = \frac{be - cd}{ac - b^2} \text{ and } t_c = \frac{ae - bd}{ac - b^2}
$$

(A5-10)

With the solutions for $s_c$ and $t_c$, the points $A(s_c)$ and $B(t_c)$ are defined and their distance can be calculated as
**Distance between parallel line segments**

\[
\text{dist} = |A(s) - B(t)| = |\overrightarrow{w_0} + s \cdot \overrightarrow{u} - t \cdot \overrightarrow{v}|
\]

\[
= \left( w_{0x} + s \cdot u_x - t \cdot v_x \right)^2
+ \left( w_{0y} + s \cdot u_y - t \cdot v_y \right)^2
+ \left( w_{0z} + s \cdot u_z - t \cdot v_z \right)^2
\]

\[
= \left( P_x - R_x + s \cdot (Q_x - P_x) - t \cdot (S_x - R_x) \right)^2
+ \left( P_y - R_y + s \cdot (Q_y - P_y) - t \cdot (S_y - R_y) \right)^2
+ \left( P_z - R_z + s \cdot (Q_z - P_z) - t \cdot (S_z - R_z) \right)^2
\]

\[
(A5-11)
\]

The requirement that the lines pieces PQ and RS themselves need to cross (Figure A5.1B) rather than their extensions (Figure A5.1A) can be tested by the values of the parameters \(s\) and \(t\). When they are both in the domain \([0, 1]\), then when

\[
s \in [0, 1] \quad \text{and} \quad t \in [0, 1]
\]

then the points \(A(s)\) and \(B(t)\) are on the line pieces PQ and RS, respectively.

Two particular cases need further attention.

**Line pieces are running parallel.** When the two line pieces (and their infinite extensions) are running parallel the two equations in (Eq. A5-9) are dependent with the denominator (Eq. A5-10) being zero. The distance between the parallel lines can be determined by taking for instance an arbitrary value for \(s\), say \(s = 0\), and then solving (Eq. A5-9) resulting in \(t = -d/b = e/c\), followed by the calculation of the distance using (Eq. A5-11). By running parallel, an infinite number of orthogonal connections can be drawn between both lines. Additional tests are now needed to determine whether both line pieces share a range where they intersect both these orthogonal connection lines. If not, then both line pieces are not crossing. If yes, then a choice need to be made for a particular orthogonal connection line and the intersection points at both line pieces (in order to locate a synaptic connection); see the choice made in Eqs. 32 and 33.

**Line pieces are not parallel but in one plane.** When the two line pieces are not parallel but still in Plane then they intersect and a solution exists for the equality \(A(s) = B(t)\). Then, using Eqs. A5-1 and A5-2 we obtain

\[
P + s \cdot \overrightarrow{u} = R + t \cdot \overrightarrow{v} \text{ or } P - R + s \cdot (Q - P) - t \cdot (S - R) = 0
\]

This set of equations (for \(x, y, \text{and } z\)-coordinates) has a solution for \(s\) and \(t\) when

\[
\begin{align*}
P_x - R_x & \quad Q_x - P_x \quad S_x - R_x \\
P_y - R_y & \quad Q_y - P_y \quad S_y - R_y \\
P_z - R_z & \quad Q_z - P_z \quad S_z - R_z
\end{align*}
\]

resulting in

\[
s = \frac{b_1 c_y - b_y c_1}{a_1 b_y - a_y b_1} \quad \text{and} \quad t = \frac{a_1 c_y - a_y c_1}{a_1 b_y - a_y b_1}
\]

For the corresponding points \(A(s) = P + s \cdot \overrightarrow{u}\) and \(B(t) = R + t \cdot \overrightarrow{v}\), it has subsequently to be determined if they are located on the line segments PQ and RS, respectively.

**Shortest distance between two non-crossing line pieces**

The procedure for calculating the shortest distance between two line pieces PQ and RS is based on an algorithm of Dan Sunday (http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm) and is taken as the shortest distance between any two points on these line pieces, respectively.

A line piece PQ is given by the points on the line A (Figure A5.2):

\[
A(s) = P + s(Q - P) = P + s \overrightarrow{u} \text{ with } 0 \leq s \leq 1
\]

A line piece RS is given by the points on the line B (Figure A5.2):

\[
B(t) = R + t(S - R) = R + t \overrightarrow{v} \text{ with } 0 \leq t \leq 1
\]

If the points \(A(s)\) and \(B(t)\) at the crossing of infinite lines \(A\) and \(B\) are located on the line segments PQ and RS, respectively, then they define the orthogonal connection and the line pieces PQ and RS themselves do cross. If \(A(s)\) and \(B(t)\) are located outside of the ranges of either one or both of the two line pieces, then we...
need to find the values of \( s \) and \( t \) which define the shortest connection \( \tilde{w}(s,t) = A(s) - B(t) = \tilde{w}_0 + \tilde{u} - \tilde{v} \) between the line pieces. Minimizing the length of \( \tilde{w}(s,t) \) is equivalent to minimizing \( |\tilde{w}(s,t)|^2 \), thus minimizing \( (\tilde{w}_0 + \tilde{u} - \tilde{v}) \cdot (\tilde{w}_0 + \tilde{u} - \tilde{v}) \). For the infinite lines this function has its true, global minimum at their crossing site \((s_c, t_c)\). With the restrictions of \( 0 \leq s \leq 1 \) and \( 0 \leq t \leq 1 \), different configurations need to be distinguished. When the points \( A(s_c) \) and \( B(t_c) \) are outside both line pieces, the shortest distance is between the points of both line pieces closest to \( s_c \) and \( t_c \), i.e., \( SQ \) in Figure A5.2A, and \( SP \) in Figure A5.2C. If only one of the points \( A(s_c) \) and \( B(t_c) \) is on its respective line piece the shortest distance is between some point on this line piece and the closest point on the other line piece. Suppose that \( A(s_c) \) is on line piece \( PQ \) and the nearest point on line piece \( RS \) is \( S \) (i.e., \( t = 1 \)) (see Figure A5.2B) then we need to minimize

\[
|\tilde{w}^2(1,t)| = (\tilde{w}_0 + \tilde{u} - \tilde{v}) \cdot (\tilde{w}_0 + \tilde{u} - \tilde{v}).
\]

Taking the derivative with respect to \( s \) we obtain

\[
0 = \frac{d|\tilde{w}^2|}{ds} = -2\tilde{u} \cdot (\tilde{w}_0 + \tilde{u} - \tilde{v}) \text{ such that}
\]

\[
s_c = \frac{\tilde{u} \cdot \tilde{v} - \tilde{u} \cdot \tilde{w}_0}{\tilde{u} \cdot \tilde{u}}
\]

Therefore the shortest distance occurs between the points \( S \) and \( A(s_c) \). A similar procedure applies for other configurations, with for instance \( B(t_c) \) on line piece \( RS \) and a search for the closest point on \( PQ \).