我们已经开发了一种称为全路径修剪（APP）的自动图形算法，用于跟踪神经元的3D结构。 为了避免神经元某些部分的潜在错误跟踪，APP首先通过追踪从种子位置到图像中每个可能的目标体元/像素位置的最佳测地线最短路径来产生初始过度重构。 由于初始重建包含所有可能的路径，因此可能包含冗余结构组件（SC），我们通过使用新的最大覆盖最小冗余（MCMR）子图算法，简化了整个重建，而不会通过修剪冗余结构元件而不损害其连接性。 我们表明MCMR具有线性计算复杂度并且将会收敛。 我们用模型生物（例如果蝇）的挑战性3D神经元图像数据集检查了我们的方法的性能。

The most successful strategy among these algorithms is to build-up the reconstruction by incrementally adding more and more such SCs into the morphological modeling of a neuron. Good examples include image voxel scooping (Rodriguez et al., 2009), ray shooting (Wearne et al., 2005) and template matching (Zhao et al., 2011). These bottom-up local searching methods are suitable for 3D images that have ideally continuous neurite tracts and good signal-to-noise ratio (SNR).

然而，通过各种显微镜方法（例如3D激光扫描显微镜）获得的神经元的3D形态结构的精确数字化在实践中仍然是非常有问题的。 当图像具有低SNR，和/或由于内在的突触神经突结构（例如突触boutons）或样品制备中的缺陷（图1f和g））引起的破碎和模糊的神经突片段时，这是特别困难的。 值得注意的是，这些数据集对于不同动物的神经系统是常见的。 例如，在果蝇（果蝇）（图1f），秀丽隐杆线虫和小鼠（图1g）的单一荧光图像中，穿刺的经常破裂的神经突可以普遍地看到。 上面讨论的局部搜索方法不能容易地处理这些困难的情况，因为很难跨越这些间隙（即，低信号区域）。

解决这一具有挑战性的情况的一个策略是结合全球和地方线索。全局先验信息，如神经突结构的起始和终点位置，将引导使用局部图像内容进行更精细的优化。可以使用新颖高效的三维可视化系统V3D（Peng et al。，2010b）轻松提供神经突结构的全球先验。图形增强变形模型（GD）算法（图形增强可变形模型）可用于自动跟踪从起始位置到每个终点的最优路径（Peng et al。，2010a）。然后，通过检测沿合并路径的分支点，可以自动组合整个重构（Peng等，2010b）。迄今为止，该方法已成功应用于最详细的小鼠脑图像图谱中重建神经元连接（Li et al。，2010）。不幸的是，当神经元的结构非常复杂或图像质量低时，输入这样简单的全局先导/指导信息可能仍然非常耗时（Zhang等，2008）。因此，非常需要一种完全自动的破碎结构和低SNR图像的方法。

我们以前建议基于这些终点的空间各向异性（正在审查的手稿）或自适应模板匹配（例如V3D系统中的AutoMarker功能），然后进行GD跟踪，自动检测神经元的提示/终点。 然而，当神经元的末端不形成锋利的尖端（例如用突触引发物终止的那些）时，尖端检测可能不准确（图1f）。 自适应模板匹配也可能错误检测不规则形状的终端。 因此，在这里，我们提出了一种迭代修剪神经元过度重建的新方法。 我们展现其效率和融合。 我们使用不同模型生物（例如果蝇）的挑战性3D神经元图像数据集来检查其性能。

神经元跟踪算法的目标是使用合理简洁的模型，尽可能地完成神经元形态的重建。准确地说，我们将神经元的重建定义为描述3D图像中该神经元的3D空间形态的拓扑结构组件的一组。结构组件（SC，图1）是一个松散定义的概念，可以意味着神经元重建中包含的神经元分支，个体重建节点，个体体素或其他子结构。当这个重建（例如图1d）覆盖神经元图像中的所有可见区域及其体素时，我们称之为重构完成。如果这种结构是完整的，我们称之为重建过度完成（即过度重构），但是其一些SC被其他SC所覆盖，因此似乎是多余的。如第1节所述，SC通常是球体，椭圆体或圆柱体。 SC被表示为重建节点，每个重建节点具有相应的形状描述符。在重建中，图形通常用于描述所有重建节点（以及SC）的拓扑关系。这样的图形可以用SWC格式编码（Cannon等人，1998）。通常，神经元重建被描述为具有根/种子重建节点，许多叶节点，分支节点和其他节点之间的树形图。我们在这里遵循同样的惯例。

**我们的新的神经元跟踪方法被称为全路径修剪（APP）。 它包括两个主要步骤，即（i）产生神经元的初始过完整重构（ICR），和（ii）修剪其冗余SC。 我们认为以神经元的每个可能体素位置为中心的球形SC。 我们设计一种高效可靠的方法，以便根据这些SC（2.2节）产生完整的重建。 然后，我们设计一个最大最小冗余（MCMR）子图搜索算法，自动确定冗余SCs并将其删除，直到没有人能被进一步删除（第2.3节）。 以这种方式，我们有效地简化了使用最少数量的SC的完整重建，同时保持了整个重构神经元结构的最佳覆盖**。

我们的方法将三维神经元图像I和位于神经元区域内的种子位置Ps =（xs，ys，zs）作为输入。种子通常是神经元的神经元或另一个大的亮点（在没有成像的情况下），并且可以在许多情况下自动检测（这超出了本文的讨论范围;因此，这里我们假定它的可用性）。让我们假设明亮但不是黑暗的图像体素代表神经元信号。为了产生神经元的初始完全重建（ICR），我们使用整个图像的平均强度值ta作为全局阈值来定义图像的“前景”，即任何体素值比假设ta是神经元的一部分，否则它属于图像背景。值得注意的是，在荧光图像中，亮体素的量通常很小;因此，ta通常非常低。因此，使用ta作为全局阈值非常保守，从而确保所有可能的信号被捕获。在全局阈值处理之后，我们还可以选择运行带有3×3×3窗口的中值滤波器来消除噪声。

接下来，我们在图像前景上创建无向图G =（V，E），其中图形顶点V的集合代表图像体素，并且图形边缘集合E编码下面定义的测地公制度量函数。

由于我们的测地度量函数仅输出正值，所以我们使用Dijkstra算法（Dijkstra，1959）找到从种子位置Ps到G中每个其他顶点的最短路径。由于G仅包含相邻体素顶点的边缘， 它是非常稀疏的。 因此，求解几何最短路径的时间复杂度为O（NlogN）（N为G中的顶点数）。 实际上，这个步骤通常只需要几秒钟就可以在当前的笔记本电脑上运行512×512×128的图像堆栈。

在最终的最短路径图中，我们搜索没有子节点的顶点，并调用这样的集合作为叶集。 显然，我们可以回溯每个叶顶点到种子位置的路径，也称为根顶点。 所有这些路径共享许多共同点。 我们将整个解决方案整理为树形图。 由于此步骤检测到从根到每个图像前景体素的所有有序路径，因此不包括假阴性，我们将该树形图称为全路径重建，其是ICR。 这种自动ICR方法类似于我们早期的GD（Peng et al。，2010a）和V3D-Neuron 1.0（Peng et al。，2010b）中的图形步骤。 不同的是，在GD中，我们有一个关于神经突的开始和结束位置的先前信息，而在这里我们使用所有可能的前景体素作为神经元的结束位置。 由于底层图G保持不变，与V3D-Neuron 1.0相比，初始重建具有相同的计算复杂度。

我们将ICR中的每个顶点/节点视为SC。 然后，我们在接下来的三个子部分中采取三个步骤来修剪冗余的SC，同时保持所有SC的整体覆盖。 整个策略被称为最大覆盖最小冗余（MCMR）子图搜索。

2.3.1 Dark-leaf pruning (DLP)

我们使用了一个非常保守的全局阈值ta，它是整个图像的平均强度，用于定义图像前景。 原因是为了捕获神经元中的所有可能的路径，我们必须最大化任何可能是连接到种子位置的神经元区域的明亮图像区域的潜在连接。 这个阈值通常很低，以致于人眼不可见许多“前景”体素。 在ICR中，这些不可见区域对应于许多冗余分支。 因此，我们使用另一个阈值tv，其定义了最低的“可见”体素强度。 我们选择tv = 30（假设我们有一个8位图像，最大强度为255）。 然后，由于重建是一个树形图，我们迭代地去除其各个体素强度低于电视的所有叶节点; 直到不再检测到叶节点。 以这种方式，我们可以最大限度地提高不同地区的连通性，同时降低重建的结构复杂性。

2.3.2 Covered-leaf pruning (CLP)

为了有效地修剪叶节点，我们估计每个剩余重建节点的覆盖半径。 可以使用图像距离变换来估计这样的半径。 然而，3D距离变换对噪声和不规则前景边界敏感。 因此，我们开发了一种更强大的方法。 我们定义以重建节点为中心的半径可调整球体，然后逐渐放大半径，直到该球体内的图像体素的0.1％比全局阈值ta（即整个图像的平均体素强度）更暗。 在大多数情况下，0.1％的选择使得神经突的估计边界与图像背景有明显的对比。 实际上，0.1％的阈值与V3D-Neuron 1.0系统中的缺省值（Peng et al。，2010b）相同，已被神经元重建社区越来越多地使用。 我们将每个重建节点及其估计的半径，即覆盖范围，作为SC进行处理。

在ICR中，有很多冗余重建节点彼此高度重叠。显然，被覆盖的重建节点可能被删除，而不影响重建的完整性。然而，哪些重建节点应该先删除？一般来说，这是一个高度组合的子集选择问题，这是很难的。然而，生产ICR的一个显着优点是，由于我们已经有所有重建节点的有序连接，我们可以设计以下线性时间方法来去除这些冗余节点。我们注意到（i）当没有节点覆盖叶节点时，应始终保留该叶节点（图2a）和（ii）如果叶节点被另一个节点（图2b）或其他几个节点共同覆盖图2c），可以安全地修剪该叶节点。因此，我们检查重建中的所有叶节点，并删除其他剩余节点明显覆盖的叶节点。我们重复这个修剪过程，直到不再修剪叶节点。

为了使此过程尽可能高效，我们首先扫描整个重建，其中所有暗叶节点已被删除。我们创建一个查找表来记录重建节点的所有3D空间位置。然后，对于每个节点a，我们创建一个空的覆盖列表Ca，它将存储覆盖节点a的其他节点的标识。接下来，我们再次扫描整个重构，对于每个节点b，我们计算b的覆盖范围（由其半径确定）是否包括任何其他节点（例如，节点a）空间位置。如果是，那么我们把b放在一个覆盖列表中，这个列表是按半径排序的（从大到小）。排序是自动完成的，因为我们正在扩展每个节点的覆盖列表。可以看出，由于每个节点仅由本地邻居覆盖，所以确定覆盖列表的过程只需要线性时间。前段讨论的实际修剪只需要扫描重建当前剩余叶节点的几次;因此，该算法的整体时间复杂度是线性时间。还要注意，这种修剪算法保证收敛，因为在重建中存在有限数量的节点。当我们修剪越来越多的叶子节点时，当没有叶被其他重建节点覆盖时，这个过程必须停止。

2.3.3 Inter-node pruning (INP) and refinement

在覆盖叶节点修剪之后，重建已经是简洁的，在所有神经元区域已经被最少数量的叶节点到达的意义上。 但是，通过删除将叶节点连接到分支节点或根节点的冗余节点，可以进一步降低重构的复杂性，而不影响其连接性和完整性。

当满足等式（5）的条件时，则该节点间被修剪，并且a的父节点被更新为b的原始父节点。 如果没有修剪b，那么我们检查一下b的非分支节点的父节点，如c所示，将根据c和b的覆盖关系进行修剪。 对于每个叶节点，我们迭代此过程，直到达到分支点或根。 对于每个分支点，我们做同样的事情，直到它的父分支点或根到达。 这样我们可以修剪所有冗余节点。 值得注意的是，在等式（5）中，我们使用了与等式（3）不同的阈值。 这是因为整体尽可能少的重建节点应该用于维持整个神经元区域的覆盖。 因此，应该修剪最少数量的叶节点，并且应该修剪节点间数量最多的节点。

所有重建节点，除了根，在简化重构中都有整数空间坐标，这对应于我们最初使用的图像体素顶点（第2.2节）。 因此，重建的各个部分可能看起来并不完全平滑，这对人类的观察更直观。 因此，我们可以使用基于梯度下降的可变形曲线优化（Peng et al。，2008，2010a）来改进所有节点间的位置。 这实际上等同于对于段中的每个节点之间运行平均偏移算法，直到收敛，相对于路径最小弯曲的正则化约束。 节点间细化步骤具有与处理所有叶节点作为神经元末端然后运行我们以前的V3D-神经元跟踪方法类似的效果（Peng等人，2010b）。