子空间追踪算法和OMP以及BP算法相比，它兼具同OMP一样的低复杂度和BP一样的信号重建质量。

SP与CoSaMP几乎是一致的，他们的主要区别在于“Ineach iteration, in the SP algorithm, only K new candidates are added, while theCoSAMP algorithm adds 2K vectors.”，即SP每次选择K个原子，而CoSaMP则选择2K个原子；这样带来的好处是“This makes the SP algorithm computationally more efficient,”，因此可以说由于SP算法每次选择的是K个原子，因此造成了SP算法相对于MP算法而言具有更优的计算效率。

“贪婪类算法虽然复杂度低运行速度快，但其重构精度却不如BP类算法，为了寻求复杂度和精度更好地折中，SP算法应运而生”，“SP算法与CoSaMP算法一样其基本思想也是借用回溯的思想，在每步迭代过程中重新估计所有候选者的可信赖性”，“SP算法与CoSaMP算法有着类似的性质与优缺点”。

子空间追踪算法的提出相较于CoSaMP略晚，但他们的算法原理几乎是完全一样的。SP与CoSaMP几乎是一致的，他们的主要区别在于SP算法每次迭代选择K个原子，而CoSaMP选择的则是2K个原子，这样做的好处是使得SP具有相对于CoSaMP而言更优的计算效率。SP与CoSaMP的另外一个区别是SP在迭代过程中增加了一个停止迭代的条件：当残差经过迭代后却变大了的时候就停止迭代。由上节可知，SP算法的输入输出及详细步骤如下：

输入：

维的传感矩阵

维的观测向量

信号的稀疏度

输出：

信号稀疏表示系数估计值

维残差

算法流程为：

1. 初始化残差值，，，；
2. 计算（即计算），选择中个最大值，将这些值对应的列序号构成集合（列序号集合）；
3. 令，；
4. 求的最小二乘解： ；
5. 从中选出绝对值最大的项记为，对应的中的列记为，对应的的列序号记为
6. 更新残差：；
7. 如果，停止迭代直接进入第9）步；
8. ，如果则返回第2)步继续迭代，如果或残差则停止迭代并进入第8）步；
9. 重构所得的稀疏矩阵在处有非零项，其值分别为最后一次迭代所得。获得后，就可以获得重构后的信号： 。

上述流程中，表示残差值，表示迭代次数，表示空集，表示每次迭代找到的索引（序列号），。表示第次迭代的索引（列序号）集合， 表示矩阵的第列，表示按索引选出的矩阵的列集合，为的列向量，表示迭代次数。符号表示集合并运算，表示求向量的內积，表示求模值（绝对值）。