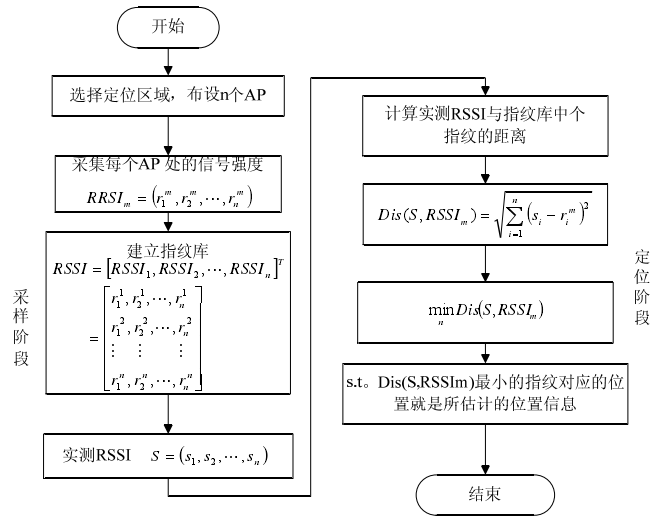
**基于 k-means聚类和 WKNNSS的位置指纹定位算法**

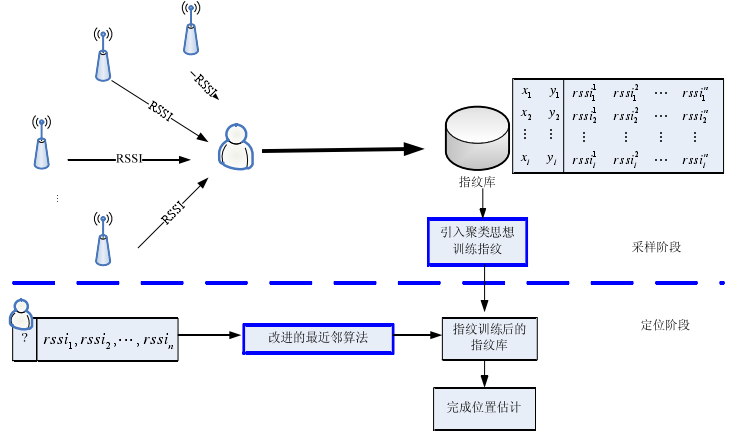
1. **引言**

位置指纹算法只需 RSSI 作为指纹库的指纹，无需其他特征参数，在定位阶段进行指纹匹配，便可以估计位置。虽然无需测距的定位算法相比于基于测距的定位算法，定位误差较大，但是位置指纹算法降低了对定位系统的硬件要求，采用 RSSI 作为指纹库的特征参数是低成本的并且实现简单。但是，采集 RSSI 建立指纹数据库的过程中不可避免地会引入测量误差，以至于影响指纹库的准确性，势必会导致后续的位置估算的结果。在位置指纹算法的位置估算阶段，由于最近邻算法实现简单的优点选择最近邻法作为估计算法，将实测指纹与指纹库中相似度最大的指纹作为定位结果，比较简单，但是由于参考位置的单一，也不可避免的影响待测目标的位置结果。因此，如何提高指纹数据库的准确性，如何减小最近邻算法对定位结果的影响，成为位置指纹算法研究中亟待解决的问题之一。



**图1.位置指纹算法的算法流程**

本文提出了一种适用于 Wi Fi 环境下的结合了 k-means 聚类和改进的 KNNSS 算法的位置指纹算法——KWKLF 算法。KWKLF 定位算法的基本思想是，在数据的离线采样阶段采集能够接收到的 RSSI 值，建立指纹库，利用经典的 k-means聚类方法对得到指纹进行聚类分析，找出有粗大偏差的 RSSI 信息，加以剔除，利用剩下的相对准确的 RSSI 组建新的指纹库，降低了指纹匹配过程的计算量；在实时定位阶段，采用实测指纹与指纹库匹配的思想，对最近邻算法进行改进，对未知位置进行估算。如图2 所示。



**图2.KWKLF算法的工作原理**

1. **基于 k-means 聚类和加权 KNNSS 的位置指纹算法**

基于 k-means 和 WKNNSS 的位置指纹算法（以下简称 KWKLF算法）是一种改进的位置指纹定位算法。位置指纹算法分为指纹训练阶段和位置估算阶段，为了提高位置指纹定位算法的定位精度，本文引入k-means 聚类算法对数据库进行训练，降低指纹库的搜索空间，并对现有的最近邻算法进行改进，以减小定位算法可能引起的定位误差。

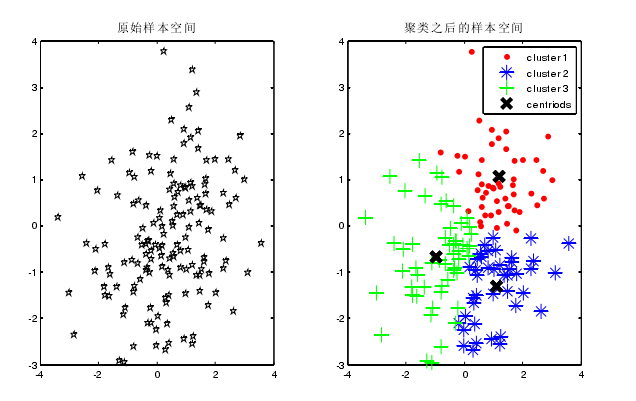
**2.1 k-means聚类**

k-means 聚类也可称为 k-均值聚类。k-means 聚类方法是一种使用广泛、适用于多种

数据类型的聚类算法，算法简单，实现快速。k-means 聚类是典型的基于距离的聚类算法，

以距离作为相似性的度量，其算法思想是根据现有的样本之间的相似度划分为 k 个子类，

是相似度较大的样品聚集在一起，相似度比较小的样品彼此远离。实验表明，如图3 所示，k-means 聚类可以高效分类，使得整个指纹库中划分为不同的小类，减少位置指纹搜索空间。



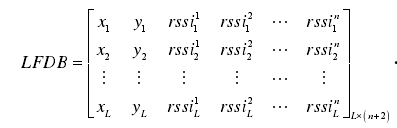
**图3.实验样本空间原始数据聚类前后的对比图**

**2.2基于k-means的位置指纹库**

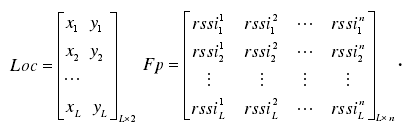
位置指纹算法的第一阶段是建立位置指纹库。在 Wi Fi 环境下，假设某个特定的室内

区域中，可以检测到n个 Wi Fi 信号强度即 RSSI，在这个定位区域内选定 L 个采样点，采

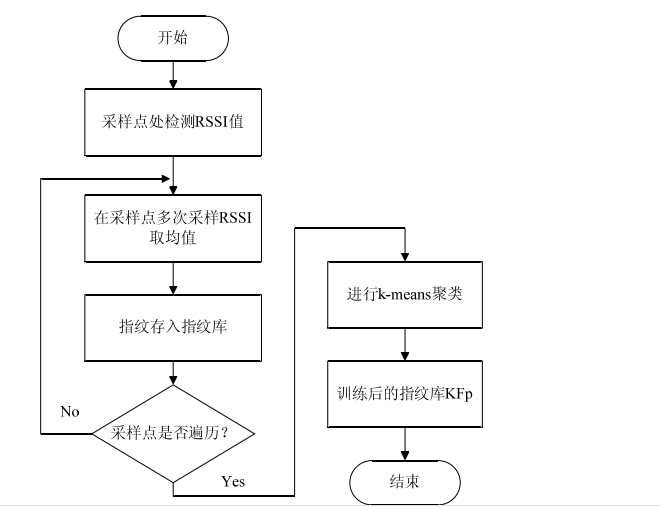
样点的位置信息已知，采用二维空间坐标（x,y）表示。在每个采样点可以观测到n 个 RSSI值，（rssi1 rssi2 ... rssin)作为这个采样点的指纹，每一个指纹与其采样点的位置一一对应，成一一映射的关系。那么这个指纹库可以表示为公式（1）所示。

**....................(1)**

其中，LFDB 包含 RSSI 序列和位置信息。那么指纹与位置信息就可以分开表示为位置集合 Loc和指纹集合 Fp（Fingerprint）：

**.................(2)**

训练指纹库的指纹时需要对指纹数据进行预处理，然后采用聚类算法进行分类处理，则训练之后的指纹库为 KFp 。训练过程如图4 所示。



**图4.指纹库训练流程**

图 4描述了指纹库的训练流程，指纹的训练是基于指纹库的建立的前提下进行的。

指纹库从建立到训练宜分为以下四个步骤：

Step1.选定室内环境作为定位区域，在这个定位区域中，选取L个采样点，且记录

每个采样点的位置坐标。

Step2. 在每个采样点，利用具有WiFi信号检测功能的移动终端进行检测信号强度，

可以检测到n个Wi Fi信号强度，多次测量得到求均值，得到这个采样点的指纹，；遍历 L 个采样点，得到 L 个指纹，存入指纹库。

Step3.将指纹库进行 k-means 聚类，以欧氏距离作为相似度的评价准则，距相近的指

纹聚集在一个子类，距离较大的指纹彼此远离。

Step4.多次执行 step3，直到聚类结束，指纹库变成具有 k 个子类的指纹样本空间。

其中在 Step3 中所提到的 k-means 聚类方法，其执行过程分为以下五步：

1．输入L 个指纹和聚类个数k(k≤L)；从L 个指纹中任意选择k 个指纹作为初始的聚类中心。

2. 对于剩下的(L-k)个指纹,计算每个指纹到每个类中心的距离，其中dij 表示第 i 个指纹到第 j 个聚类中心的距离，找到min(Distance)，将第i 个指纹分类到第 j 个聚类中，得到新的聚类结果。

3.重复第 2 个步骤，将剩下的的指纹分配完成，形成 k 个聚类G1,G2...Gj...Gk，每个类Gj 都包含其聚类中心，和属于该类的指纹成员及其个数nj。

4.根据公式计算新的聚类中心，其中rssii表示Gj 类中的第 i 个RSSI 值。计算每个类的类中心，得到新的聚类中心。

5.若C\*= C，即相邻两次的聚类中心相同，即分类趋于稳定，聚类结束，当前的G1,G2...Gj...Gk代表了最终形成的聚类。否则令C= C\* ，即更新类中心，返回第 2 步骤继续执行聚类过程。 k-means 聚类算法的流程如图 5 所示。

**2.3 改进的加权KNNSS定位算法**

k-近邻算法（KNNSS）是最近邻算法（NNSS）的改进算法。由于最近邻算法易于实现，算法简单，在位置指纹算法的位置估计阶段选择最近邻算法作为位置估计算法。在实测阶段测得的实测指纹与指纹库匹配，寻的最相似的指纹，其对应的采样点的位置作为估计位置。最近邻算法选择的参考指纹较为单一，定位结果不稳定，易产生较大误差。

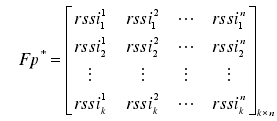
针对参考指纹单一的问题，本文提出了一种加权的 k 近邻算法（Weighted K-Nearest

Neighbor in Signal Space，WKNNSS）。在 WKNNSS 算法中，本文引入了权重系数 ，用以表示每个参考指纹的贡献程度。经过训练后的指纹库 KFp 由多个指纹聚集的小类组成。位置估算阶段，待测目标实际测量到的实测指纹为If=(rssi1,rssi2, ... ,rssin)，与 KFp 匹配；此时的指纹库表示为KFp=[G1;G2;...;Gk]，其中每个类中聚类中心分别为C\*=[FP\*1;FP\*2;...;FP\*k]。在KNNSS算法中，通常采用的加权算法是直接对位置信息进行加权即

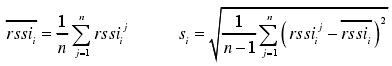
**...................................(3)**

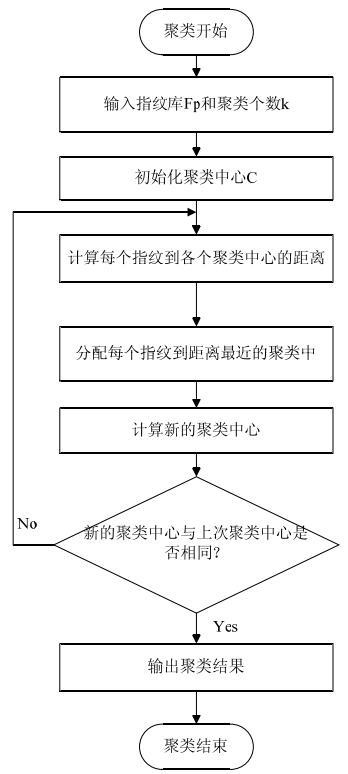
其中，权重系数为ωi=1/k，每个参考位置的权重是相同的。而本文提出的WKNNSS算法中，权重系数的选择是与指纹库的指纹相关的，是通过对相似度最大的k 个指纹进行相关的运算，计算每个指纹的贡献程度，并把这个贡献程度映射到对应的位置信息，进行位置的估算。

假设与实测指纹相似度最大的指纹空间Fp\*有k 个指纹：

**...................................(4)**

计算Fp\*中的每个指纹的权重系数ωi，将Fp\*进行转置Tfp =(Fp\*)T；分别计算每个参考点指纹的均值和标准差。

**..........................(5)**



**图5.k-means聚类流程**

其中，i=1,2, ... ,k。令，则权重系数为

ωi =vi/**............................(6)**

那么待测目标的位置如公式（7）所示。

**............................(7)**

WKNNSS 算法的执行过程如图6 所示。可以分为以下六步：

1.将实测指纹If=(rssi1,rssi2, ... ,rssin)与训练之后的指纹库 KFp 进行匹配，计算lf 与每个类中心的距离记为DIS=[d1,d2, ... ,dk]

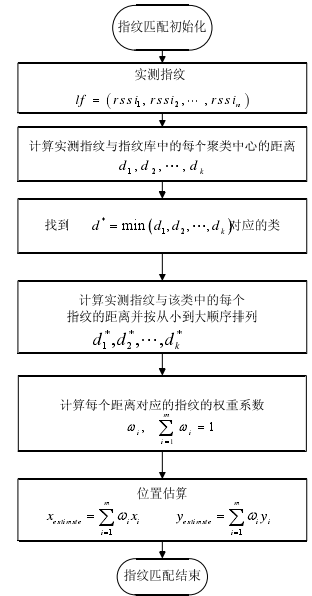
2.寻找min(DIS)对应的类，记为GSPECIAL。

3.计算实测指纹lf与GSPECIAL中的每个指纹的距离，记为DIS=[d1\*,d2\*, ... ,dng\*],其中ng表示GSPECIAL 中指纹的个数。

4.将Dis按照从小到大的顺序排列，取最小的 m 个距离，并将这 m 个距离对应的指纹选定作为参考指纹，其对应的位置坐标作为参考坐标。

5.按照公式（5-6）计算每个指纹的权重系数ωi。

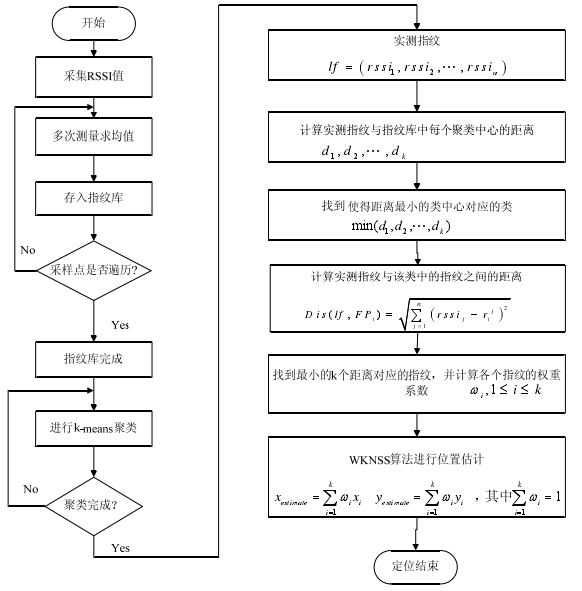
6.按照公式（7）计算实测指纹的位置坐标(xestimate, yestimate)。



**图6.WKNNSS算法位置估计流程**

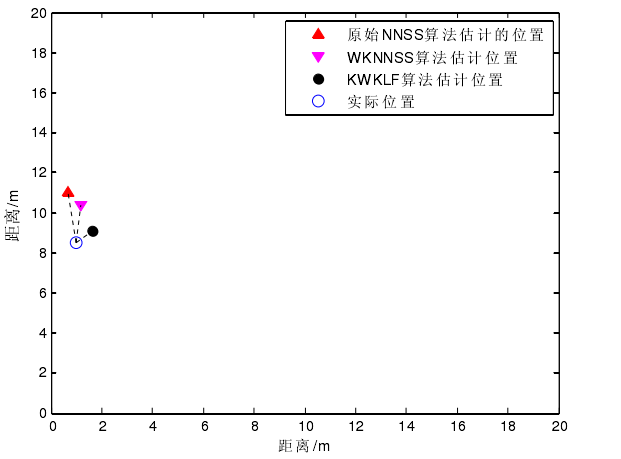
**2.4算法实现**

本文在对位置指纹算法的研究和深入分析的基础上，提出了一种基于k-means和加权KNNSS算法的位置指纹算法即KWKLF算法，在指纹库的训练阶段首次引入了k-means聚类，利用聚类的思想进行指纹库的搜索空间的划分；在位置估算阶段，提出了一种新的权重指数的计算法，使得改进的加权k近邻算法估计的位置信息更为精准。此外，在实测指纹与k-means聚类训练后的指纹库进行匹配，找到最近的类，其他的类指纹就可以去掉不做考虑。KWKLF算法的执行过程如图7所示。



**图7.KWKLF算法流程**

在选定的室内区域中，选择多个采样点，在每个采样点多次测量RSSI值求均值，得到该采样点的指纹，遍历所有采样点，建立位置指纹数据库，且每个指纹对应唯一一份采样点的位置。在定位区域任意选取一个位置进行采样，得到一个实测指纹且位置未知。将未知位置信息的指纹与指纹库匹配，分别采用最初的k近邻算法（即权重系数为均值）和改进的加权KNNSS算法进行位置的估算。两种算法得到的位置信息，显示在节点分布图上，结果如图8所示。图8清晰明了的显示改进算法的定位结果要比原始指纹算法的估计更为精准。



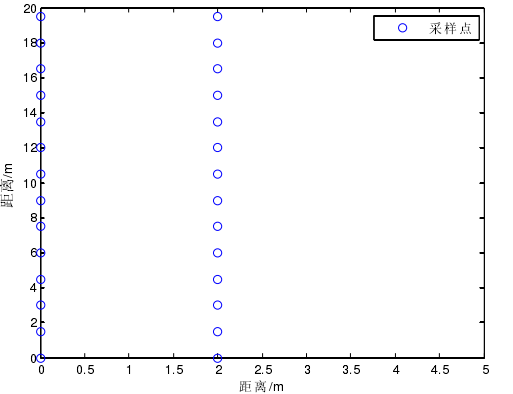
**图8.KWKLF与KNNSS定位结果显示**

**3.算法仿真与分析**

为了验证本文所提出的改进算法——KWKLF算法的性能，本文对其进行了仿真实验，采用matlab仿真软件对KWKLF 算法进行仿真，并与原有的位置指纹算法进行对比，验证了改进算法的性能。

**3.1仿真环境设置**

借助于现有的WiFi基础设施，定位区域选择了吉林大学南湖校区通信工程学院第一教学楼的3楼的走廊，在走廊可以检测到分布在未知地点的WiFi信号。由于客观条件的限制，选择了2×20m的矩形区域，把313室门口作为起点，每隔1.5m作为一个采样点。WiFi信号检测器采用智能手机APP，智能手机型号MI2SC，处理器四核1.7GHz，系统MIUI-JL50.0，APP是WiFi检视仪。定位区域分布如图9所示。在此定位区域，利用智能手机App观测RSSI的值，经过多次测量求均值，最终得到的位置指纹库如表1 所示。



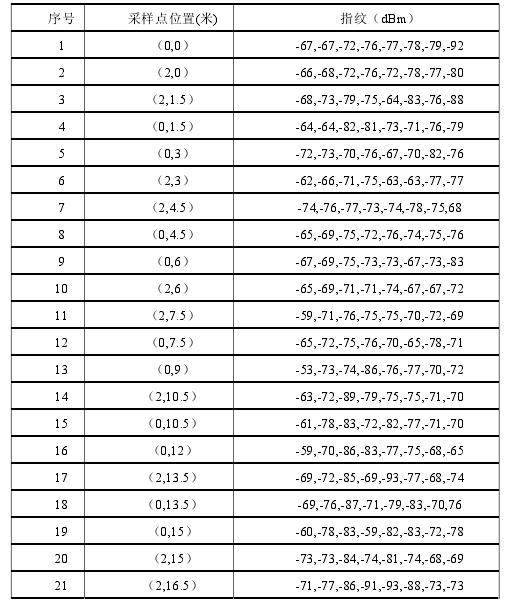
**图9.定位区域的采样点分布图**

**3.2仿真结果分析**

为了验证本文所提算法——KWKLF算法的性能，本节中选择多个实测指纹验证算法的定位性能。算法的定位性能通过两个指标来描述，一个是定位误差累积概率，一个是均方根误差（root-mean-square error，RMSE）。假设待待测目标的实际位置是,(xactual, yactual) ，通过定位算法估计的位置,(xestimate, yeatimate)，那么 RMSE 的计算公式如公式（8）所示。

**.............................(8)**

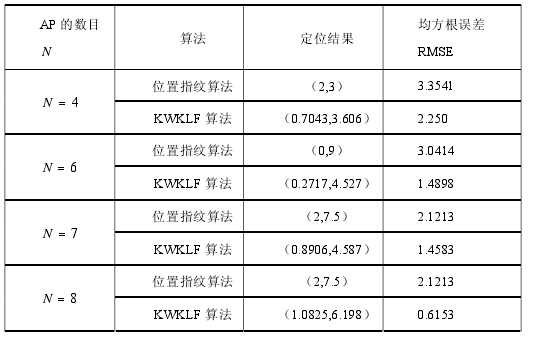
**表1.位置指纹库**

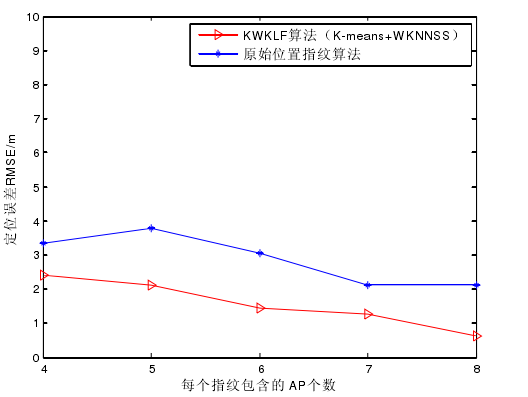


1.同一实测指纹验证AP数目对仿真结果的影响

指纹库的准确性直接影响定位结果的准确性。指纹库的每个指纹都包含多个RSSI值，每个RSSI值来源于不同的AP，那么AP的数目对定位结果也有影响。本仿真实验在分别对AP数目为4，6，7，8 四种场景进行仿真实验，并对比了原始位置指纹算法与KWKLF算法的性能，仿真结果如表2所示。根据表1的仿真结果对算法的性能进行对比，对比结果如图10所示。图中显示，随着AP数目的增加，两种算法的RMSE值都是减小的，即定位结果越来越准确。

**表2.仿真结果**





**图10.可测AP数目对定位结果的影响**

从表1可以看出，不论是AP的数目是4个还是6个甚至更多，改进的位置指纹算法较原有的位置指纹算法具有更小的定位误差。由于KWKLF引入了聚类的思想和改进的WKNSS算法，提高了定位结果的准确性。当AP数目N=4 时，KWKLF算法的均方根误差降低了27.98%；

N=6 时，KWKLF算法的RMSE降低了58.09%；N=8 时，KWKLF算法的RSME降低了高达70.99%

。由此可见，室内环境中，利用Wi Fi信号进行定位，改进的位置指纹算法相比于原有的位置指纹算法性能有了显著的提升。从图10和表1可以看出，不管是原有算法还是改进算法，都是在N=8 时，定位效果较好，因此，接下来的仿真实验都是在N=8 的条件下进行的。

2．选择不同的实测指纹验证改进算法的性能

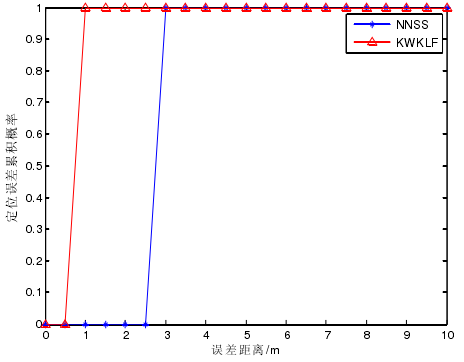
为了验证本文所提算法的性能，本文针对不同的实测指纹进行仿真实验。在此过程中，选择定位误差累积概率和RMSE作为评价指标。定位误差的累积概率曲线的收敛速度在一定程度上反映了定位算法的性能。在这个仿真实验中，本文选择2个实测指纹对误差累积概率进行仿真。本文分别对实际位置为（1, 8.5），（0.5, 6）所测的指纹与指纹库匹配，并通过定位误差的累积概率来验证算法的性能。仿真参数如表3所示。该仿真实验结果都是进行100

次迭代之后得出的结果。

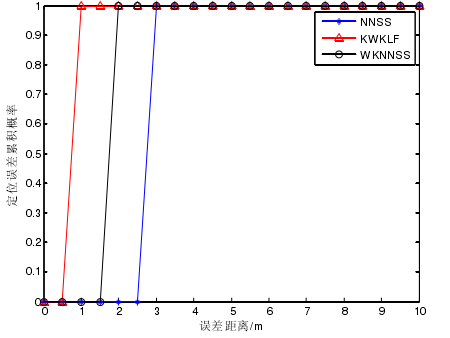
**表3.仿真参数**



图11、图12分别给出了在实际位置为（1, 8.5）、（0.5, 6）的实测指纹通过原始位置指纹算法与改进算法得到的定位误差的累积概率的变化情况。在图中NNSS代表的是原有的位置指纹算法，没有对指纹库进行训练，指纹匹配采用最近邻算法；WKNNSS代表只是对最近邻算法NNSS进行改进，并没有引入 k-means聚类来训练指纹库；KWKLF算法是本文提出的改进算法，采用k-means聚类对指纹库训练，提高指纹库的准确性，并对最近邻算法进行改进，提出了WKNNSS算法，因此基于k-means和WKNNSS的位置指纹算法简称为KWKLF算法。



1. KWKLF 算法与原始位置指纹算法的性能比较



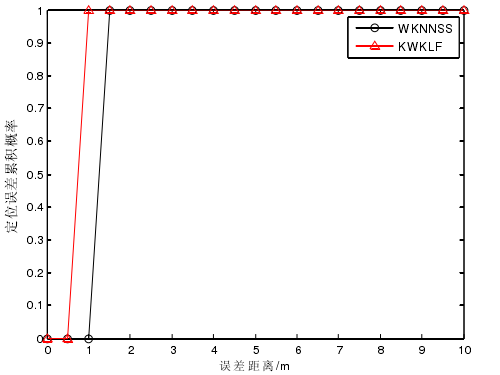
(b)KWKLF、WKNNSS、NNSS 算法的性能比较

**图11.实际位置（1, 8.5）时KWKLF与原始算法的性能比较**

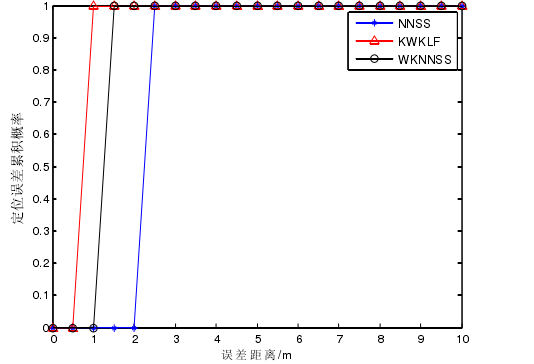
比较图11（a）、（b）可以看出：（1）在实际位置为（1, 8.5）的情况下，本文提出的改进算法相比于原有算法，定位误差的累积概率分布曲线的收敛速度有显著的提高。这说明，对原有算法进行改进之后，定位误差减小，定位结果更加准确。（2）图 11（a）对比了原有算法与改进算法的性能，由图中所示，原始算法的定位误差累积概率分布曲线在误差为 2.5m左右开始收敛，在定位误差为2.6m处，收敛完成，说明原始算法的定位误差最大可达

2.6m。KWKLF算法的定位误差累积概率分布曲线在不到1m处就开始收敛，在1m左右收敛完成，说明KWKLF算法的定位误差最大不到1m。相比于原始算法，KWKLF算法的定位误差降低了65.12%左右。（3）图11（b）显示了对原有算法的不同改进方法的性能比较。比较

WKNNSS与NNSS算法的曲线，可以看出，对最近邻算法改进，引入权重系数之后，定位结果的误差有了显著的降低。与原有算法相比，加权的k近邻算法WKNNSS的定位误差降低了25.88%左右。KWKLF算法与WKNNSS算法相比，增加了k-means聚类对指纹库的训练过程，定位误差降低了近52%左右。



1. KWKLF、WKNNSS算法的性能比较

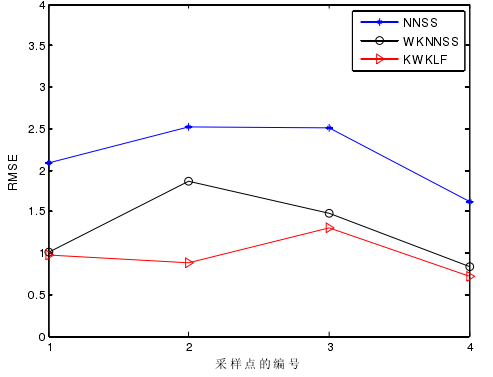


(b) KWKLF、WKNNSS、NNSS 算法的性能比较

**图12 实际位置(0.5, 6)时KWKLF算法与原始算法的性能比较**

比较图12（a）、（b）可以看出：（1）在实际位置为（0.5,6）的情况下，相比于原有算法，改进算法KWKLF算法的定位误差累积概率分布曲线的收敛速度更快。可见，改进算法的定位误差减小，定位结果更加准确。（2）图12（a）对比了WKNNSS与KWKLF算法的性能，由图中所示，WKNNSS 算法的定位误差累积概率分布曲线在误差为1m左右开始收敛，在定位误差为1.1m处，收敛完成说明WKNSS算法的定位误差在1m左右。KWKLF 算法的定位误差累积概率分布曲线在0.6m处就开始收敛，在不到1m处收敛完成，说明KWKLF 算法的定位误差在0.5~1m之间。从定位性能上看，KWKLF算法相比于WKNNSS 算法，定位误差最大降低了近40%。（3）图12（b）显示了对原有算法的不同改进方法的性能比较。比较WKNNSS与NNSS 算法的曲线，可以看出，对最近邻算法改进，引入权重系数之后，定位结果的误差有了显著的降低。与原有算法相比，加权的k 近邻算法WKNNSS的定位误差相比于NNSS算法，降低了50%左右。KWKLF算法与NNSS算法相比，增加了k-means聚类对指纹库的训练过程，并在NNSS算法的基础上，提出了加权的k紧邻算法WKNNSS，定位误差降低了近52%。

纵观图11和图12，不管是法改进的WKNNSS算法还是既有指纹库训练和WKNNSS的KWKLF算法在定位性能上都有显著的提高，定位结果的均值误差如图13所示。图13的仿真实验选择了4个不同的位置测量的指纹进行验证改进算法的有效性。原始位置指纹算法的RMSE在2-3m之间，WKNNSS算法的定期误差在1-2m之内波动；KWKLF算法的定位误差在1m之内。相比于原有算法。改进算法KWKLF的定位误差最大降低了65.12%；引入聚类思想后，同采用WKNNSS算法进行位置估计，KWKLF算法相比于WKNNSS算法，定位误差最大降低了52%。这说明了，KWKLF算法中，引入的k-means聚类有效减少了指纹库的相对而言“不准确”的指纹，又对最近邻算法进行改进提出的WKNNSS算法，利用与实测指纹最相似的K个指纹对应位置估计实测指纹的位置。这样做既提高了指纹库的准确性，又避免了未知位置信息受限于最近的一个指纹所对应的位置所包含的误差，引入的权重系数还均衡了多个指纹对应的位置信息对位置的估计产生的影响。更重要的是，本文的实验环境是吉林大学南湖校区的第一教学楼的三楼的走廊，测量分布在不同的地点的AP信号，那么本文提出的KWKLF算法比原有算法不仅具备更小的定位误差，性能稳定，还更适应于实际环境。



**图13 不同的实测指纹利用不同的定位算法位置估计的定位误差**

1. **小结**

本文首先研究了如何对指纹库进行训练以减小指纹库的不准确对定位结果的重要意义，并分析了NNSS算法对位置估计的局限性。在对现有的解决方法进行详细的分析和研究之后提出了本文的KWKLF算法，然后详细的阐述了KWKLF算法的基本思想和实现步骤。为了验证本文所提算法的性能，本文选定了实际的环境，实测了指纹库，而不是利用仿真软件的函数产生的数据。本文分析了AP数目对KWKLF算法的定位结果的影响，以及选定不同的实测指纹对KWLF算法进行验证。

仿真实验结果表明，相比于原有的位置指纹算法，改进算法——KWKLF算法的定位

结果更为准确，误差最大降低了65.12%；此外，改进算法具有更好的鲁棒性，定位误差

波动较小，定位性能在1m之内，定位性能较为稳定。

本文从位置指纹算法的指纹库与定位算法两个方面进行改进，研究的内容和方法有

一定的局限性，算法的适用场景也具有一定的局限性。随着Wi Fi定位技术的广泛发展，

那么对定位技术的要求越来越严格。总体来讲，今后的工作主要集中在以下几个方面：

1.本文研究的是Wi Fi环境下的位置指纹算法，选择的定位区域有限，指纹库的空间

也比较小。如何增加指纹样本和扩大定位区域，将会是今后的研究问题之一。

2.本文研究的位置指纹算法选择最近邻算法作为位置估计算法，对算法进行改进提

出加权的k近邻算法，即使减小了定位误差，但是在WKNNSS中的k值的选择是人工设

定的。如何自适应的设定k值，也会成为以后工作的研究点。

无线网络信号强弱的影响因素如下  
一、频段干扰  
理论上讲同一个频段内无线网络过多会严重影响信号的强弱，也就是说如果一用户采用的无线信号频段与其它用户的无线信号发射频段一样的话，那么在一定程度上两用户的无线网络都会受到影响  
若使用的是路由器默认配置时的信道，这样就很容易发生信道的干扰建议换信道  
  
二、射频干扰   
如果网络带宽中噪音强度超过了-85dBm，那么射频干扰就存在着损害网络性能的可能。在这种情况下，用户的重试率将超过10%,这时用户会感到网络速度受到了影响。如，在无线用户位于一个与正在运转的微波炉同样的房间时，就会发生这种情况。  
若断定问题是由于射频干扰引起的，就得找到这种干扰来自哪里，并设法去掉这种干扰。如果仅仅在微波炉或无线电话运行时有这种症状，就得试着将AP接入点放到不同的信道。如果还是无法将射频干扰减少到一个可以令人接受的水平，就可以试着在受影响的区域增加射频信号的强度。例如，可以增加发射功率，就可以用更强的单元代替原有的天线，或者是将接入点靠得更接近一些。这此方法会增加信噪比，从而改善性能。  
  
三、路由器的摆放位置   
由于无线信号在穿越障碍物后，尤其是在穿越金属后，信号会大幅衰减。而在我们家庭的房子里，有很多钢筋混凝土墙，所以为了增强无线网络信号，我们在摆放无线路由器的时候，应该使信号尽量少穿越墙壁。一般来说，无线网卡的客户端都具备信号强度的检测能力，大家拿笔记本在房间各处查看信号强度，从而选择一个最佳的摆放点。   
  
四、天线的扩展  
由于天线增益的大小直接影响到信号的发射强度和接收能力，而市场上有些路由器的天线采用的是可拆卸设计，所以给无线路由器更换一个高增益的天线是增强信号最直接的方法。增益天线市场上有很多，价格也便宜。不过一点值得注意的是，在购买的时候应该询问清楚是否是全向天线，否则使用定向天线只能向一个方向传输无线信号。

无线传输路径分析是无线传输网络设计的重要步骤，通过对传输路径的分析便于网络设计者根据无线链路的裕量大小选择合适类型的天线（方向，极化，增益等指标），安装天线高度，选择合适的馈缆和长度等。下面将简单介绍一下无线传输路径分析中的自由空间损耗的计算，信号接收强度的计算，链路系统裕量的计算几个主要方面的内容。

1.自由空间损耗的计算

自由空间损耗是指电磁波在传输路径中的衰落,计算公式如下：

Lbf=32.5+20lgF+20lgD

Lbf=自由空间损耗(dB)

D=距离（km）

F=频率(MHz）

2400MHz：Lbf=100+20lgD

5800MHz：Lbf=108+20lgD

通过查找上表和通过公式计算我们可以得到从发射站到接收站电磁波传输的理论衰落.

2.信号接收强度的计算：

信号接收强度是指接收站设备接收到的无线信号的强度。

RSS=Pt+Gr+Gt-Lc-Lbf

RSS=接收信号强度

Pt=发射功率

Gr=接收天线增益

Gt=发射天线增益

Lc=电缆和缆头的衰耗

Lbf=自由空间损耗

举例说明，如果发射站与接收站两站点相距25Km，设备发射功率20dBm，发射天线增益为17dBi，接收天线增益为24dBi，电缆和缆头损耗3dBi。则接收信号强度    RSS=20+17+24-3-128=-70dB （需要已知无线发射点的位置，在实际应用中不太现实）

<http://bbs.c114.net/thread-749287-1-1.html>

http://wenku.baidu.com/link?url=AVvlTM5LBmgYMRrOLlztlSRZQu8vfWBH0zAjz\_mi8lI8GbvvxoufdgXa9p5j-ObKdomtsucGF\_WBDM1ztDke0RpkI45AKiQXVqp2li1V7jK