

一种单目标路径覆盖分区动态优化算法

蒋一波 王 伟 何成龙

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 目标检测是无线传感器网络的一个重要应用。在目标检测过程中,用户在获取目标丰富的图像信息的基础上,对目标的路径覆盖情况更为关注。针对单目标在整个运动路径中均能被 K 级覆盖且传感器分布密度达到最小这一问题,首先结合有向感知模型和单目标位置数学预测模型,给出了传感器在理论上的最小分布密度,然后设计了单目标路径覆盖分区动态优化算法。该算法依据传感器自身与目标的距离,将整个监控区域中的传感器分为外部传感器、中部传感器和内部传感器,对每类传感器分别实施不同的旋转决策。仿真结果表明,提出的算法与现有的算法相比,有效地降低了监控区域中传感器的分布成本。

关键词 无线传感器网络,路径覆盖,分布密度,旋转决策

中图分类号 TP393 **文献标识码** A

Sub-regional Dynamic Optimization Algorithm for Path Coverage of Single Target

JIANG Yi-bo WANG Wei HE Cheng-long

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Target detection is an important application of wireless sensor networks. In the process of target detection, users pay more attention to the path coverage of the target on the basis of obtaining rich image information of the target. Aiming at the problem that a single target can be covered by the K-level and the sensor distribution density is minimized in the whole motion path, firstly, the theoretical minimum distribution density of the sensor was proposed by combining the directed distribution model and the single-target position mathematical prediction model. Secondly, the sub-regional dynamic optimization algorithm for path coverage of single target was designed. According to the distance between the sensor and the target, the sensors in the entire monitoring area is divided into external sensors, middle sensors and internal sensors by the algorithm to implement different rotation decisions for each type of sensor. The results of simulation show that compared with the existing algorithms, the proposed algorithm can effectively reduce the distribution cost of sensors in the monitoring area.

Keywords Wireless sensor network, Path coverage, Distribution density, Rotation decision

1 引言

得益于行业生产线的升级和半导体技术的进步,传感器的生产成本迅速降低,传感器密集型应用得以快速成长,加上视频传感器本身具有信息采集、数据处理和无线通信等特性,使其适用性在不同的应用领域得到扩展^[1]。

无线传感器网络(Wireless Sensor Network, WSN)是一种在传感器技术、微系统技术和通信技术的基础上发展起来的新兴技术。覆盖问题与定位、调度以及安全问题构成了无线传感器网络的基本问题。在某些应用场景下,如热点区域检测当中,对目标的覆盖质量提出了更高的要求,不仅要求拍摄清晰的目标图像信息,还要求对目标实施多角度的监控,以便获取更加全面的目标信息。即区域中的每个点至少被 k 个视频传感器同时覆盖到^[2],这样的问题也被称为目标的 k 覆盖问题。在另一些应用场景下,如入侵式目标检测当中,除了 k 覆盖的需求之外,还需要对目标的移动路径进行覆盖,这

种覆盖问题通常被称作路径覆盖^[3]。路径覆盖的实现依赖于高密度部署的视频传感器,即使当下传感器成本大幅下降,但大规模部署实施后,材料的总成本依然会对项目的具体实施造成巨大压力。提出一种既能使网络覆盖目标路径效果最大化,又能最小化工作传感器的数量的算法很有必要。

本文提出的单目标路径覆盖算法先对移动目标建立一个合适的数学模型,然后通过该模型预测目标一个时间步长后所处的位置,最后对传感器实施合适的旋转策略,从而覆盖目标下一刻的位置。出于对节能方面的考虑,本文采取休眠机制^[4],节点周围没有事件发生,节点自动进入低能耗状态,从而节约能量。另外,传感器的分布数量减少,也在一定程度上解决了能量损耗的问题。

2 相关工作

在目标覆盖的研究领域之中, Ai 等^[5]率先提出 MCMS 问题,即用最少的节点覆盖最多的静止目标,并给出解决

MCMS问题的经典算法——DGA算法和CGA算法。随后又有人针对该问题进行了大量研究^[6-8]。但MCMS问题设定目标处于静止状态,不适用于移动目标覆盖而且对于目标覆盖要求只是一级覆盖,不能进一步捕捉目标信息。

Liu等^[9]采用K级覆盖的方法解决了捕捉目标信息不清晰的问题,并给出目标K级覆盖概率公式。结合K级覆盖的概念,Fusco等^[10-11]针对用最少的节点来K级覆盖最多的目标这一问题进行研究,给出了一种简单贪心算法,选取最少节点调节感知方向从而K级覆盖更多目标,后来又将障碍物、能量等参考因素加入实验环境控制当中,提升算法的通用性。

对于移动目标覆盖的研究,Wang等^[12]通过节点来跟踪移动目标,得出目标的运动轨迹。对于移动目标K级覆盖,陈琼等^[13-14]针对单目标,提出了DPGKCA算法,该算法通过前两个时刻目标的位置,来预测目标第三个时刻的位置。其针对多目标的运动又提出了MPKCDA算法,调节节点感知方向,提升多目标覆盖期望值,可以使多目标的覆盖可能性大大提高。

随着目标多重覆盖要求的提高,监控区域部署的传感器数量不断增加,但以上算法并未对该问题提出合适的解决方案。本文结合有向感知模型和单目标的运动特性,提出了传感器最少部署问题,并研究了该问题的分布式解决方案。然后依据单目标位置数学预测模型,提出了基于目标预测模型的单目标路径覆盖分区动态优化算法,即保证了覆盖目标路径的效率,又降低了监控区域传感器的数量。本文通过一系列的仿真实验,验证了该算法的有效性。

3 传感器最少部署问题

3.1 有向感知模型

有向传感器(为了叙述简便,以下传感器和节点均指有向传感器)本身具有视角区域以及转动等特性。相较于传统的全向传感器,其在解决移动目标的覆盖问题上能够得到更好的效果。传感器通过旋转捕获移动目标的运动方向,达到实时跟踪的目的。在实际运用场景中,传感器分布属随机部署,传感器的感知方向也带有随机性。

有向传感器在理论研究中的模型如图1所示, O 点代表节点随机部署的位置, T 表示目标当前所在的位置, R 是节点的感知半径,单位向量 $\vec{V}(t)$ 是节点感知区域的中轴线,代表节点的感知方向。 α 是节点感知范围的边界和感知方向 $\vec{V}(t)$ 的夹角,称为感知方向偏移角。节点转动角速度用 ω 表示,在一个时间步长 Δt 内,节点最多可以向逆时针或顺时针旋转 $\omega\Delta t$ 度。

判断目标能否被节点覆盖的方法为:若 $distance(O, T) \leq R$ 且节点与目标连线向量 \vec{OT} 和节点感知方向 $\vec{V}(t)$ 之间的夹角不超过 α ,则认为目标被节点覆盖到。

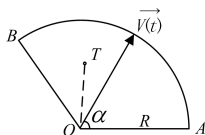


图1 有向感知模型

3.2 传感器最少部署问题

在研究问题之前,先做出如下假设:

(1)监控区域中的传感器均是同构的有向传感器,即具备同样的感知半径 R 、感知方向偏移角 α 和旋转的角速度 ω 。

(2)监控区域中的传感器均能获取自己的位置信息,且对感知方向可控。规定逆时针旋转方向为正方向。

(3)传感器通过红外线技术来确定目标的位置。

(4)在监控区域中的单目标保持速度大小不变的不定向运动,一个时间步长 Δt 内可看作方向不变的匀速直线运动,经过距离为 r 。

传感器最少部署问题是单目标维持速度大小不变,方向时刻发生变化的运动,监控区域保证目标任意时刻均能被K级覆盖,且传感器分布密度达到最小。

3.3 单目标位置数学预测模型

移动目标覆盖算法的重点在于找到一个合适的预测模型,用于准确预测移动目标下一时刻的位置。基于单目标运动特性,可以得出图2所示的目标预测模型——目标预测圆周。

该圆周的半径等于目标一个时间步长 Δt 所经过的距离,用 r 来表示。目标在第一时刻位于点 T ,下一个时刻肯定落在半径为 r 的圆周上。因此得出目标下一时刻位置的数学预测模型,以下研究都将围绕目标预测圆周进行。另外,节点的感知半径为 R ,目标预测圆周的半径为 r ,我们基于现实的情况认定 $R \gg r$ 。

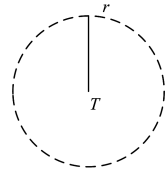


图2 目标预测圆周

3.4 理论最小分布密度

在保证目标K级覆盖的前提下,要使节点的分布密度达到最小,首先研究节点理论上的最小分布密度。我们先做出如下假设:监控区域中的节点不存在角速度 ω 的限制,节点可以随时转向任意方向。

节点分布密度与节点感知区域和目标预测圆周的交集情况有关。节点的感知区域覆盖目标预测圆周越多,也就是交集越大,节点覆盖目标下一时刻所在位置的可能性也就越大,从而减少监控区域中实施覆盖的节点数量。而其交集情况取决于节点与目标之间的距离 $distance(O, T)$ 。

如图3所示,目标位置在 T_2 时,目标预测圆周内切于节点感知区域直线边界,此时 $distance(O, T_2) = \frac{r}{\sin \alpha}$ 。目标在 T_1 时,目标预测圆周内切于节点弧边,此时 $distance(O, T_1) = R - r$ 。通过描述两个临界条件,当 $distance(O, T)$ 不处于 $[\frac{r}{\sin \alpha}, R - r]$ 时,节点感知区域不能完全覆盖目标预测圆周。

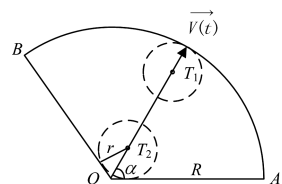


图3 目标预测圆周内切于节点感知区域

下面是 $distance(O, T)$ 与节点覆盖目标预测圆周的详细描述:

(1) $distance(O, T) > R + r$,节点感知区域不存在与目标

圆周交集的可能性。

(2) $R+r \geq \text{distance}(O, T) \geq R-r$, 节点感知区域与目标预测圆周存在交集,但不存在完全覆盖目标预测圆周的可能性,需要其他节点辅助覆盖目标下一时刻的位置。

(3) $R-r > \text{distance}(O, T) > \frac{r}{\sin \alpha}$, 节点通过旋转感知区域可以完全覆盖目标预测圆周,算法当中实施覆盖的主要是分布在这一距离的节点。

(4) $\text{distance}(O, T) < \frac{r}{\sin \alpha}$, 情况如同(2)。这里不再赘述。

4 种情况如图 4 所示。

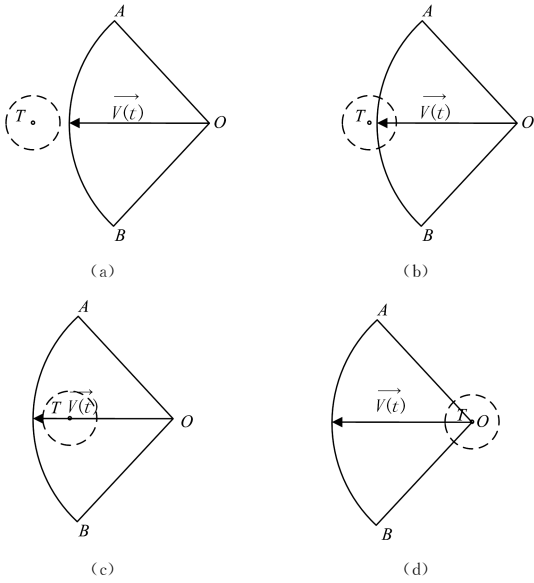


图 4 节点感知区域与目标预测圆周的交集情况

从图 4 可以看出,节点分布范围如(1)、(2)和(4)这 3 种情况,未能完全覆盖目标预测圆周,要覆盖目标下一时刻的位置,可能需要另外的节点辅助覆盖,因此会增加监控区域中的节点数量。

当节点分布如(3)情况,即 $R-r > \text{distance}(O, T) > \frac{r}{\sin \alpha}$ 的圆环,需要节点数量最少。如图 5 所示,灰色部分的区域 G 表示该范围(以下部分用 G 来表示目标这部分圆环区域),可以得出理论上节点最小分布密度,用 ρ 来表示理论上节点最小分布密度:

$$\rho = \frac{k}{\pi(R-r)^2} \quad (1)$$

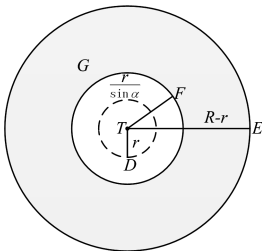


图 5 节点分布图

4 单目标路径覆盖分区动态优化算法

基于上节得出的 $\text{distance}(O, T)$ 与节点感知区域覆盖目标预测圆周情况的关系,根据 $\text{distance}(O, T)$ 把整个监控区域中的节点划分为外部、中部和内部 3 类。

3 类节点划分范围如下:

(1) $\text{distance}(O, T) > R-r$, 外部节点。

(2) $R-r \geq \text{distance}(O, T) \geq \frac{r}{\sin \alpha}$, 中部节点。

(3) $\text{distance}(O, T) < \frac{r}{\sin \alpha}$, 内部节点。

我们针对 3 类节点实施不同的旋转决策,开展各自的工作。外部节点的主要工作是调整感知方向,以便及时变成中部节点能够完全覆盖目标预测圆周。中部节点的主要工作是完全覆盖目标预测圆周,以便在不需要其他节点辅助的情况下覆盖目标下一时刻的位置。内部节点考虑到目标 G 区域中可能不足 K 个完全覆盖目标预测圆周的中部节点,需要内部节点启动旋转决策辅助覆盖。

在节点旋转过程中,目标与节点的连线向量 \vec{TO} 与节点的感知方向 $\vec{V}(t)$ 的夹角就是 $\angle LOX$, 我们称之为方向角,用 μ 来表示,如图 6 所示。

$$\mu = \arccos \left(\frac{\vec{V}(t) \cdot \vec{TO}}{|\vec{V}(t)| \cdot |\vec{TO}|} \right) \quad (2)$$

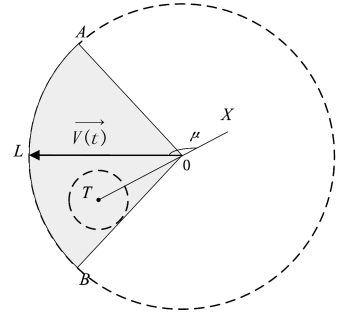


图 6 节点刚刚覆盖预测圆周的情况

在算法实施过程中节点会存在角速度的限制,节点可能在一个时间步长内不能旋转到完全覆盖目标预测圆周的方向。为了便于判断节点是否旋转到完全覆盖目标预测圆周的方向,我们用方向角来划定节点完全覆盖目标预测圆周的夹角范围。在 3 种节点当中,能够完全覆盖目标预测圆周的只有中部节点,分布范围就是目标 G 区域,为此,我们针对中部节点,提出临界角的概念。

在中部节点的 360° 旋转过程中,在保证节点完全覆盖目标预测圆周的前提下,我们称最小的方向角 μ 为该节点的临界角度,用 β 来表示。当节点的 $\mu > \beta$ 时,节点完全覆盖目标预测圆周。图 6 中的 μ 为该节点的临界角度。因此,可以得出以下公式:

$$\beta = 180^\circ + \arcsin \left(\frac{r}{\text{distance}(O, T)} \right) - \alpha \quad (3)$$

可以看出, β 与 $\text{distance}(O, T)$ 成负相关。 $\text{distance}(O, T)$ 越大, β 就越小,节点通过旋转完全覆盖目标预测圆周的概率越大。

接下来定义两个角度以便判断节点旋转方向。称节点右边感知边界向量 \vec{OA} 与 \vec{TO} 的夹角 $\angle AOX$ 为右角,用 $right$ 表示。以同样方式定义节点左边感知边界 \vec{OB} 与 \vec{TO} 的夹角,即 $\angle BOX$, 称之为左角,用 $left$ 表示。

3 种节点的旋转决策都与 $\text{distance}(O, T)$ 有关,同时方向角和 $\text{distance}(O, T)$ 是节点旋转决策的重要依据。

4.1 外部节点旋转决策

假定外部节点在目标运动过程中均保持睡眠状态,可能会

遇到图 7 所示的情况。当外部节点首次进入 G 区域,也就是位于图 7 中的 O 时,感知方向为 $\vec{V}(t)_1$ 。此时,外部节点变成中部节点。但由于存在角速度的限制,节点经过一个时间步长的旋转,方向角 μ 仍然小于临界角度 β ,也就是说如图 7 所示,感知方向 $\vec{V}(t)_1$ 在一个时间步长内无法回旋到 $\vec{V}(t)_2$,可以用公式表示为 $\beta - \alpha > \omega \Delta t$,这种情况导致刚刚从外部转变为中部的节点无法完全覆盖目标预测圆周。

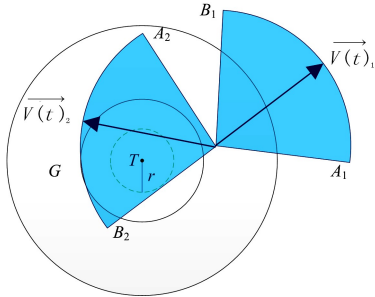


图 7 节点与目标示意图

为了避免出现上述情况,外部节点必须提前旋转。判断依据从 3 个角度出发:1)划定外部节点首次进入目标 G 区域的位置范围,以便确定变成中部节点时的临界角度 β ;2)判断目标是否有靠近外部节点的趋势;3)分析在目标靠近过程中外部节点是否有足够的时间通过旋转可以使方向角大于转变后中部节点的临界角度。

首先,确定外部节点变成中部节点的位置范围。目标在一个时间步长内经过的距离为 r ,在目标 G 区域中可以划定外部节点变成中部节点的区域范围。如图 8 所示,外部节点首次进入目标的 G 区域,肯定会落在 $R - r \geq \text{distance}(O, T) \geq R - 2r$ 圆环上。

由式(3)可以得出,临界角度 β 与 $\text{distance}(O, T)$ 成负相关, $\text{distance}(O, T)$ 越小, β 越大,中部节点的方向角 μ 回旋到临界角度 β 的可能性越小。我们以最差的情况作为考量,认为外部节点首次进入目标 G 区域均是分布在了 $\text{distance}(O, T) = R - 2r$ 的圆周上,也就是图 8 中 P 所指的圆周上。此时,临界角度是最大的,所以旋转到临界角度的可能性是最小的。可以得出该圆周上中部节点的临界角度,即当 $\text{distance}(O, T) = R - 2r$ 时的临界角度,用 γ 表示:

$$\gamma = 180 + \arcsin\left(\frac{r}{R-2r}\right) - \alpha \quad (4)$$

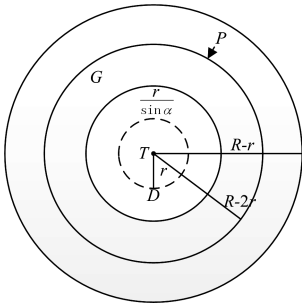


图 8 节点首次进入目标 G 区域的示意图

在确定了外部节点变成中部节点位置范围之后,再分析目标在移动过程是否有靠近节点的趋势。

外部节点与运动目标之间的关系如图 9 所示。目标每经过一个时间步长, $\text{distance}(O, T)$ 可能增加,也可能减少,这取决于目标下一时刻的位置。而目标下一时刻的位置在目标预测

圆周上呈一定概率分布,如图 9 所示,用 θ 表示目标预测圆周任意一点构成的半径向量 \vec{TC} 与 \vec{TO} 之间的夹角。用 $f(\theta)$ 表示目标下一时刻在目标预测圆周上的概率密度分布函数。用 D 表示目标经过一个时间步长 $\text{distance}(O, T)$ 的变化值, $D > 0$ 表示目标有靠近节点的趋势, $D < 0$ 表示目标有远离节点的趋势。

$$D = \int_0^{2\pi} r f(\theta) \cos(\theta) d\theta \quad (5)$$

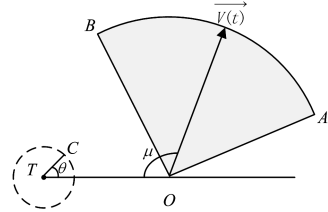


图 9 节点与目标预测圆周转动示意图

通过以上分析,可以确定外部节点首次进入目标 G 区域的分布范围,得出该范围的临界角度 γ 并且确定目标是否有靠近节点的趋势。假定目标有靠近外部节点的趋势 $D > 0$ 且外部节点此刻的方向角 $\mu < \gamma$,外部节点需要考虑下一个问题——在目标靠近过程当中,方向角 μ 是否可以及时回旋到 γ 。

我们可以用 Num 估计当前节点的方向角 μ 需要经过多少个时间步长才能回旋到 γ 。用 Mum 估计需要经过多少个时间步长外部节点进入目标 G 区域。当 $Num > Mum$ 时,在目标靠近过程中,外部节点方向角 μ 不能及时旋转到 γ ,所以外部节点必须提前旋转。

$$Num = \left\lceil \frac{\mu - \gamma}{\omega} \right\rceil + 1 \quad (6)$$

$$Mum = \frac{\text{distance}(O, T) - R + r}{D} \quad (7)$$

综合以上的分析,可以用 Num 和 Mum 判断外部节点是否需要提前旋转。若 $Num > Mum$ 表示方向角 μ 未能旋转到 γ ,外部节点必须提前旋转;否则,节点保持睡眠状态,节约能量。

4.2 中部节点旋转决策

中部节点的分布范围就是目标的 G 区域。算法当中的中部节点主要负责完全覆盖目标预测圆周,以此降低传感器部署数量,节约能量消耗。

中部节点的旋转决策首先将中部节点划分为 A, B, C 类。 A 类节点的覆盖结果决定 B 类节点是否实施覆盖。同时, B 类节点的覆盖结果也决定了 C 类节点是否有实施覆盖的必要。

3 类中部节点定义如下:

- (1) 方向角 μ 大于或等于临界角度 β 的中部节点为 A 类节点。
- (2) 方向角 μ 小于临界角度 β ,但是在一个时间步长内可以旋转到临界角度 β 的中部节点为 B 类节点。
- (3) 方向角 μ 小于临界角度 β ,并且在一个时间步长内不能旋转到临界角度 β 的中部节点为 C 类节点。

中部节点旋转决策中应优先选择 A 类节点实施覆盖,其无须旋转,可以降低传感器能量消耗。如果 A 类节点的数量小于 K ,不能满足目标的 K 级覆盖要求,则需要启动 B 类节点实施覆盖,此时应优先选择 $\text{distance}(O, T)$ 较大的 B 类节点, $\text{distance}(O, T)$ 越大,临界角度 β 越小,旋转角度越小,传感器能量消耗越少。 C 类节点启动条件是 A 类和 B 类节点

总数依旧小于 K , 当然 C 类节点本身不可能在一个时间步长内回旋到临界角度 β , 但是为了提升覆盖的质量, 算法要求 C 类节点尽可能回旋使方向角 μ 接近临界角度 β , 包含更多的目标预测圆周的圆弧, 以提高目标下一时刻 K 级覆盖的可能性。

4.3 内部节点旋转决策

当 C 类的中部节点需要启动参与覆盖, 说明了目标 G 区域中不足 K 个完全覆盖目标预测圆周的中部节点。为了提升算法覆盖的质量, 需要内部节点参与覆盖目标预测圆周。

内部节点的分布范围是 $\frac{r}{\sin\alpha} > \text{distance}(O, T)$ 的区域, 该区域分布的节点不能完全覆盖目标预测圆周, 只与圆周有所交集, 根据 $\text{distance}(O, T)$ 不同, 交集情况可分为两种。

我们先做如下定义: 内部节点感应区域包含最多目标预测圆周的圆弧时的方向角 μ 称为最大角 δ :

落在 $\frac{r}{\sin\alpha} > \text{distance}(O, T) > r$ 区域中的节点与目标预测

圆周交集的情况如图 10 所示。为了提高目标下一时刻 K 级覆盖的可能性, 节点要尽可能包含更多目标预测圆周的圆弧。通过严格的数学逻辑推导可以得出, 分布在该范围的内部节点的最大角 δ 。

$$\delta = 180^\circ - \arcsin\left(\frac{r}{\text{distance}(O, T)}\right) + \alpha \quad (8)$$

落在 $\text{distance}(O, T) < r$ 区域中的内部节点与目标预测圆周交集情况如图 11 所示, 最大角 $\delta = 180^\circ$ 。

当目标的 G 区域中 C 类中部节点参与覆盖时, 则需要启动内部节点旋转决策。所有内部节点均要旋转, 使其方向角 μ 尽可能接近最大角 δ , 以便覆盖更多的目标预测圆周的圆弧。

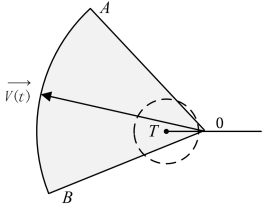


图 10 内部节点与目标预测圆周交集情况示意图(一)

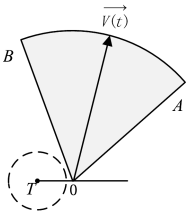


图 11 内部节点与目标预测圆周交集情况示意图(二)

4.4 算法描述

单目标在监控区域内做速度大小不变且方向时刻发生变化的运动, 力求目标在整个运动路径中的任一时刻均能被 K 级覆盖到, 且节点分布密度达到最小。结合单目标运动特性, 提出了单目标路径覆盖分区动态优化算法(KLDOA), 该算法论证了要覆盖运动中的单目标很大程度上取决于目标跟节点之间的距离, 因此根据节点与目标之间的距离 $\text{distance}(O, T)$, 可以将整个监控区域中的节点划分成外部节点、中部节点和内部节点。每种节点实施旋转决策都以 $\text{distance}(O, T)$ 和方向角 μ 为重要依据。这 3 类节点在目标移动过程当中相互转化, 相互协助, 共同实现 K 级覆盖移动目标的路径。

下面给出算法详细描述。

算法 1 单目标路径覆盖分区动态优化

输入: 节点的感知方向 $\vec{V}(t)$ 和节点的位置

输出: 节点的感知方向 $\vec{V}(t+1)$

1. $t \leftarrow 0$ // 初始化时间计数步长
2. 求一个时间步长 Δt 内目标移动的距离 r 和节点最大旋转角度 $\omega \Delta t$
3. while(true)
 - 3.1 $t \leftarrow t + 1, P = \text{FALSE}$ // P 指是否启动内部节点旋转决策
 - 3.2 计算节点的方向角 μ 和 $\text{distance}(O, T)$, 根据 $\text{distance}(O, T)$ 和目标一个时间步长移动距离 r , 判断节点属于哪一种类型。
 - 3.3 如果是外部节点
 - 3.3.1 计算节点的方向角 $\mu, \gamma, D, \text{Num}$ 和 Mum , 实施外部节点旋转决策
 - 3.3.2 continue
 - 3.4 如果是中部节点
 - 3.4.1 确定自己在中部节点中属于哪一类节点(A, B, C)并广播, 得出目标 G 区域中的 $L(A), L(B)$ 和 $L(C)$ 列表。
 - 3.4.2 实施中部旋转决策
 - 3.4.3 如果 $\text{Len}(L(A)) + \text{Len}(L(B)) < K$, 令 $P = \text{TRUE}$, 并广播 // 说明目标的 G 区域中 C 类节点参与覆盖, 所以启动内部节点旋转决策
 - 3.4.4 continue
 - 3.5 如果是内部节点
 - 3.5.1 等待一段时间能否接收到目标 G 区域中的节点发出有关 P 的信息
 - 3.5.2 如果 P 为 TRUE , 实施内部节点决策
 - 3.5.3 否则, $\text{sleep}(\Delta t)$
 - 3.5.4 continue
4. 结束

5 实验与仿真

5.1 仿真环境与实验实例

本文基于 Python3. 6. 1 的 Matplotlib 模块进行仿真实验, 仿真环境是在 300×300 的监控区域内随机部署 60 个视频传感器节点用来跟踪覆盖目标。节点的感知半径 $R = 50$, 感知方向偏移角 $\alpha = 30^\circ$, 节点转动角速度 $\omega = 30^\circ/s$ 。目标以 10m/s 做匀速不定向运动, 单位时间为 1s 。

我们通过一个实例说明提出的算法对单目标在监控区域中移动的路径覆盖过程。按照上述环境配置参数, 获取不同时间段网络的覆盖情况, 实验结果如图 12—图 15 所示。

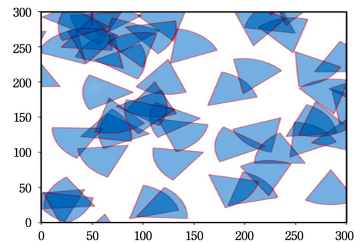


图 12 网络初始覆盖

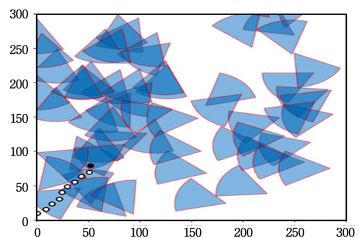


图 13 10 个时间步长后目标覆盖情况

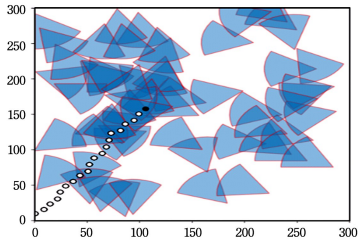


图 14 20 个时间步长后目标覆盖情况

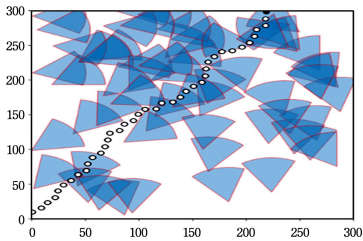


图 15 目标离开监控区域时的覆盖情况

5.2 对比实验与评价指标

本文提出的实验评价指标,考虑到了监控区域的边界可能没有充足的节点,所以认定若在监控区域运动中的目标至少有 90% 的时间步长能够被 K 级覆盖,则每个时刻都达到 K 级覆盖。在满足覆盖要求的前提下,节点分布的最小密度就是实验评价指标。

本实验参照的算法是 CONTINUE 算法(节点持续以角速度保持旋转)和陈琼等^[10]提出的 DPGKCA 算法,将 KLDOA 算法与两种算法比较,从节点的半径、节点感知方向偏移角、角速度和 K 级覆盖要求,这 4 个方面进行对照实验。

评价指标用 η 来表示, N 表示监控区域中节点的数目。 S 表示监控区域的面积。

$$\eta = \frac{N}{S}$$

本文实验采用 4 条目标运动轨迹,每一条运动轨迹实验 50 次,逐渐增加节点的数目,直到 4 条轨迹 K 级覆盖的时间步长占总的时间步长的百分比都超过 90%,则认定此时的节点分布密度为最小分布密度。

5.3 对比实验的结果

对照 K 级覆盖要求的实验,首先设定节点 $R=50$, $\alpha=30$, $\omega=30^\circ/s$,目标一个时间步长移动的距离 $r=10$,实验结果如图 16 所示。随着 K 级覆盖要求的提升,3 种算法的最小分布密度逐渐上升,整体趋势呈线性上升。而无论什么时刻,KLDOA 算法的最小分布密度远小于其他两种算法。CONTINUE 算法和 DPGKCA 算法,整体上比较接近,但 DPGKCA 算法略优于 CONTINUE 算法。

通过实验分析节点半径对于节点分布密度的影响,结果如图 17 所示。设定节点的 $\alpha=30$, $\omega=30^\circ/s$,目标一个时间步长移动的距离 $r=10$,要求目标是 4 级覆盖。半径从 30 逐渐增加到 75,节点分布密度逐渐降低。可以从图 17 中看出,前半段随着半径增加,分布密度下降较快,后半段下降较慢。这主要因为随着节点感知半径越来越大,覆盖的范围越来越大导致节点感知区域出现大量重叠现象,所以分布密度下降变化不会过大。但整体上,KLDOA 算法远好于其他两种算法。

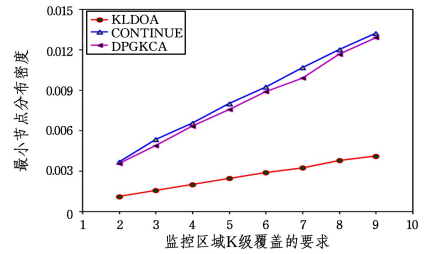


图 16 不同 K 级覆盖的实验结果

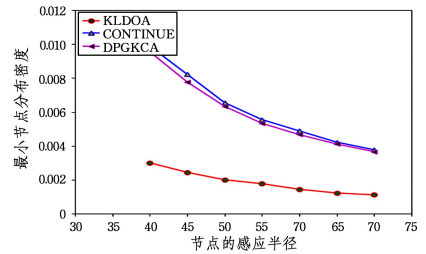


图 17 不同感知半径的实验结果

节点感知方向的偏移角($R=50$, $\omega=30^\circ/s$, $K=4$, $r=10$)和角速度($R=50$, $K=4$, $r=10$)对于节点分布密度有所影响,实验结果如图 18 和图 19 所示。在 KLDOA 算法中由于外部节点提前旋转,使得节点的感知方向偏移角对结果的影响大大的降低。对于 CONTINUE 算法,感知方向偏移角在一定范围内影响了节点旋转过程中覆盖目标的可能性,DPGKCA 算法中感知方向偏移角大小影响其旋转的范围,进而改变了覆盖目标的概率。如图 18 所示,相比于 CONTINUE 和 DPGKCA 算法,KLDOA 算法中感知方向偏移角对结果的影响最小。

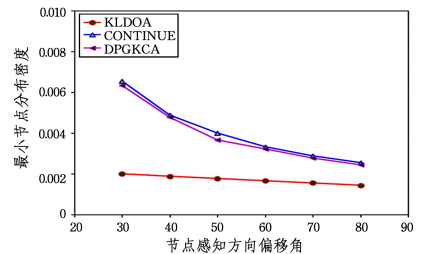


图 18 不同感知方向偏移角的实验结果

CONTINUE 算法覆盖目标完全是一个概率问题,所以角速度大小不会影响实验的结果。DPGKCA 算法覆盖目标效果取决于目标下一时刻的预测准确性,而其预测模型没有参考角速度,因此节点分布密度不会受到角速度大小的影响。在 KLDOA 算法中,外部节点旋转决策去除了角速度大小对于实验结果的波动。如图 19 所示,角速度对节点最小分布密度没有太大的影响。这个实验结果可以验证 KLDOA 算法远好于其他两种算法。

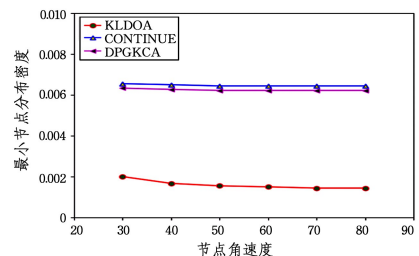


图 19 不同角速度的实验结果

结束语 单目标在监控区域中做匀速不定向运动,保证目标在整个运动路径中的任一时刻均能被 K 级覆盖到,且要求节点分布密度尽可能最小。本文提出的 KLDOA 算法依托于节点本身的特性,根据节点与目标之间的距离来实施节点旋转决策,从而使节点分布密度尽可能达到最小。通过仿真实验,可以验证该算法的有效性。在后期的研究工作中,我们会扩展目标个数,增加障碍物等更加现实的条件,以提升算法的通用性。

参考文献

- [1] LIU X. A Survey on Wireless Camera Sensor Networks[C]// International symposium on IT in medicine and education. Xining: Lecture Notes in Electrical Engineering, 2014: 1085-1094.
 - [2] 费娟,刘桂英,刘瑶. k 重覆盖设置算法的百分比覆盖研究[J]. 传感技术学报, 2018, 31(12): 1925-1930.
 - [3] 夏扬波,杨文忠,张振宇,等. 一种移动无线传感器网络的节点位置预测方法[J]. 计算机科学, 2018, 45(8): 113-118.
 - [4] CASTANO F, ROSSI A, SEVAUX M, et al. An Exact Approach to Extend Network Lifetime in a General Class of Wireless Sensor Networks[J]. Information Science, 2018; 433(4): 274-291.
 - [5] AI J, ABOUZEID A A. Coverage by directional sensors in randomly deployed wireless sensor network [J]. Journal of Combinatorial Optimization, 2006, 11(1): 21-41.
 - [6] CHEN U R, CHIOU B S, CHEN J M, et al. An Adjustable Target Coverage Method in Directional Sensor Networks [C] // IEEE Asia-Pacific Services Computing Conference. Taiwan: IEEE Xplore, 2008: 174-180.
 - [7] HSU Y C, CHEN Y T, LIANG C K. Distributed Coverage-Enhancing Algorithms in Directional Sensor Networks with Rotatable Sensors [C] // International Conference on Distributed Computing and Networking. Hong Kong: Springer, 2012: 201-213.
 - [8] WU M C, LU W F. On target coverage problem of angle rotatable directional sensor networks [C] // Seventh International Conference on Innovative Mobile & Internet Services in Ubiquitous Computing. Taiwan: IEEE, 2013: 605-610.
 - [9] LIU L, MA H, ZHANG X. On Directional K-Coverage Analysis of Randomly Deployed Camera Sensor Networks [C] // IEEE International Conference on Communications. Beijing: IEEE, 2008: 2707-2711.
 - [10] FUSCO G, HIMANSHU G. Selection and orientation of directional sensors for coverage maximization [C] // IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks. Rome: IEEE, 2009: 1-9.
 - [11] FUSCO G, GUPTA H. Placement and Orientation of Rotating Directional Sensors [C] // IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks. Boston: IEEE, 2010: 1-9.
 - [12] WANG Z, BULUT E, SZYMANSKI B K. Distributed Target Tracking with Directional Binary Sensor Networks [C] // IEEE Global Telecommunications Conference. Honolulu: IEEE, 2009: 1-6.
 - [13] 蒋一波,陈琼,王万良,等. 视频传感器网络中基于移动目标轨迹预测 K 级覆盖增强算法[J]. 传感技术学报, 2014(7): 956-963.
 - [14] 蒋一波,陈琼,王万良,等. 视频传感器网络中多路径 K 级覆盖动态优化算法[J]. 仪器仪表学报, 2015(4): 830-840.
-
- (上接第 347 页)
- [40] WILLIAMS P, SION R, CARBUNAR B. Building castles out of mud: Practical access pattern privacy and correctness on untrusted storage [C] // Proc. of the 15th ACM Conf. on Computer and Communications Security. ACM Press, 2008: 139-148.
 - [41] WILLIAMS P, SION R. Access privacy and correctness on untrusted storage [J]. ACM Trans. on Information and System Security, 2013, 16(3): 12.
 - [42] 孙晓妮. 二叉树结构的多用户茫然 RAM 方案 [D]. 济南: 山东大学, 2016.
 - [43] BOYLE E, CHUNG K M, PASS R. Oblivious parallel RAM and applications [C] // Proc. of the 13th Theory of Cryptography Conference. Springer-Verlag, 2016: 175-204.
 - [44] GOODRICH M T, MITZENMACHER M, OHRIMENKO O, et al. Privacy-Preserving group data access via stateless oblivious RAM simulation [C] // Proc. of the 23rd Annual ACM-SIAM Symp. on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2012, 13(S1): 157-167.
 - [45] 孙晓妮,蒋瀚,徐秋亮. 基于二叉树存储的多用户 ORAM 方案 [J]. 软件学报, 2016, 27(6): 1475-1486.
 - [46] BINDSCHAEDLER V, NAVEED M, PAN X, et al. Practicing oblivious access on cloud storage: The gap, the fallacy, and the new way forward [C] // Proc. of the 22nd ACM Conference on Computer and Communications Security. ACM Press, 2015: 837-849.
 - [47] SAHIN C, ZAKHARY V, ABBADI E, et al. Taostore: Overcoming asynchronicity in oblivious data storage [C] // Proc. of the 37th IEEE Symp. on Security and Privacy. IEEE, 2016: 198-217.
 - [48] 李红卫,上官经纶,古春生. 基于 ORAM 存储外包安全访问的研究 [J]. 微电子学与计算机, 2015, 32(5): 6-10, 15.
 - [49] BOGDANOV D, LAUR S, WILLEMSON J. Sharemind: A framework for fast privacy-preserving computations [C] // Proc. of the 13th European Symp. on Research in Computer Security. Springer-Verlag, 2008: 192-206.
 - [50] BEN-DAVID A, NISAN N, PINKAS B. FairplayMP: A system for secure multi-party computation [C] // Proc. of the 15th ACM Conf. on Computer and Communications Security. ACM Press, 2008: 257-266.
 - [51] 李红卫,叶飞跃,陈丹. 一种基于 ORAM 的数据可恢复性证明与访问模式的隐藏 [J]. 电信科学, 2013, 29(12): 101-106.
 - [52] WANG X S, NAYAK K, LIU C, et al. Oblivious data structures [C] // Proc. of the 21st ACM Conf. on Computer and Communications Security. ACM Press, 2014: 215-226.