

二次修正误差的 DV-Hop 改进算法

马肖旭, 刘文菊, 王 赧

(天津工业大学 计算机科学与软件学院, 天津 300387)

摘 要: 针对 DV-Hop 算法中估算节点间距离和确定未知节点坐标时存在较大误差的问题, 提出了一种基于二次修正误差的 DV-Hop 改进算法(Secondary Correction Error DV-Hop, SCEDV-Hop). SCEDV-Hop 算法首先采用多通信半径进行通信, 细化节点间跳数值, 同时采用正交多项式拟合方法, 修正未知节点和信标节点间的估算距离. 最后将未知节点与信标节点间的估计距离和距离误差共同作为加权因子构造加权矩阵, 采取先相减然后再平方的策略求解定位方程组得到未知节点坐标, 并且利用解得冗余信息再一次修正未知节点的位置坐标. 通过仿真实验验证, SCEDV-Hop 算法对未知节点进行定位时可以有效降低定位误差, 提高定位精确度.

关键词: DV-Hop 定位算法; 正交多项式拟合; 距离误差; 加权矩阵; 冗余信息

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2018)04-0012-05

The Improved DV-Hop Localization Algorithm Based on Secondary Correction Error

MA Xiao-xu, LIU Wen-ju, WANG Ze

(College of Computer Science and Software, Tianjin Polytechnic University, Tianjin 300387, China)

Abstract: Due to the low localization accuracy of the DV-Hop localization algorithm, this paper proposed an improved DV-Hop localization algorithm based on secondary correction error(Secondary Correction Error DV-Hop, SCEDV-Hop). Firstly, multi communication radius is adopted in SCEDV-Hop to refine the hop value between nodes. Moreover, orthogonal polynomial fitting method, a mathematical model, is taken advantage of to correct the distance error. Then subtraction first and then square, a strategy, is introduced to solve the equations. At the same time, the distance and distance error are taken as weighting factors to construct the weighted matrix to solve the unknown node coordinates. Finally, the redundant information obtained by solving the equations is employed to refine the coordinates of unknown nodes. Simulation experiment result showed that the SCEDV-Hop can effectively decrease positioning error and increase the positioning accuracy.

Key words: DV-Hop localization algorithm; orthogonal polynomial fitting; distance error; weighted matrix; redundant information

1 引言

近年来无线传感器网络技术已广泛应用于军事安全、智能交通、医疗卫生和智能家居与消费类电子等诸多领域. 传感器节点定位技术的研究是必不可少的, 节点需要根据有限的位置已知节点的信息来确定自身位置. 根据定位过程中是否测量节点间的

实际距离, 节点定位算法可分为基于距离(Range-based)定位算法和距离无关(Range-free)定位算法^[1]. 其中, 无需测距定位算法因其具有低能耗、低成本和无需额外硬件设施支撑等优势受到人们关注. 本文旨在深入研究 DV-Hop 算法这一种的距离无关定位算法, 并针对不足提出有效的改进算法.

2 DV-Hop 定位算法

2.1 算法描述

DV-Hop 算法由 Niculescu 等人提出,算法具体分为以下三个阶段。

阶段一,信标节点广播自身信息,网络中的全部节点得到与信标节点间最小跳数。信标节点算得自身平均每跳跳距值,并广播给全部节点:

$$\text{HopSize}_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (1)$$

式中, (x_i, y_i) 和 (x_j, y_j) 分别为信标节点 i 和信标节点 j 的坐标, h_{ij} 为信标节点 i 和信标节点 j 之间的跳数。

阶段二,未知节点找到距离自身最近的信标节点,用其平均每跳距离乘以两节点间最小跳数,算出估计距离值:

$$d_{ik} = \text{HopSize}_i \times h_{ik} \quad (2)$$

阶段三,未知节点首先逐一获取与各信标节点的估算距离。然后,再使用三边法或极大似然估计法去确定未知节点的坐标。

2.2 DV-Hop 定位算法的误差分析

DV-Hop 算法误差主要存在于两个方面:

(1) 计算两节点间的跳数时无论相距多远,在其通信范围内的都将跳数计为 1,所以利用跳数来估算节点间距离会产生较大误差。文献[2]提出基于跳数区域划分的改进方法,同时引入 RSSI 测距技术和限跳机制,但所需的 RSSI 硬件设备增加了节点开销。文献[3-4] 利用多级通信半径,细化修正跳数,从而求得的距离更加精准,但是两个算法都没有考虑转播过程中的精化跳数值。

(2) 采用最小二乘原理在求解未知节点坐标时会积累误差,从而导致了定位结果的二次误差。文献[5]利用二次曲线算法代替极大似然估计法。文献[6]分别将混合遗传微粒群算法用于校正未知节点位置,但它增加了计算时间和算法复杂度^[7]。文献[8]引入坐标修正策略,文献[9]采取加权最小平方解方程组求解并修正未知节点的坐标。但是两种算法都忽略了估算节点间距离阶段产生的误差,定位精度提高不明显。

以上文献的改进算法都不同程度上提高了定位精度,但仍存在不足,本文在此基础上提出基于二次修正误差的 SCEDV-Hop 算法。

3 正交多项式

通过实验观测到一组数据对 $(x_i, y_i) (i = 1, 2, \dots, m)$, 利用最小二乘法求一个 $n (n < m)$ 阶曲线多项式为:

$$\varphi(x) = a_0 x_0 + a_1 x_1 + \dots + a_n x_n \quad (3)$$

使已知各数据点到曲线(4) 的偏差平方和函数

$$J(a_0, a_1, \dots, a_n) = \sum_{i=1}^m \delta_i^2 = \sum_{i=1}^m (\varphi(x_i) - y_i)^2 \quad (4)$$

值最小,问题转化为求 $J(a_0, a_1, \dots, a_n)$ 极小值点,则对 a_i 求偏导数令其为 0,整理得到:

$$\begin{cases} a_0 \sum_{i=1}^m \varphi_0(x_i) \varphi_0(x_i) + a_1 \sum_{i=1}^m \varphi_1(x_i) \varphi_0(x_i) + \dots a_n \sum_{i=1}^m \varphi_n(x_i) \varphi_0(x_i) = \sum_{i=1}^m y_i \varphi_0(x_i) \\ a_0 \sum_{i=1}^m \varphi_0(x_i) \varphi_1(x_i) + a_1 \sum_{i=1}^m \varphi_1(x_i) \varphi_1(x_i) + \dots a_n \sum_{i=1}^m \varphi_n(x_i) \varphi_1(x_i) = \sum_{i=1}^m y_i \varphi_1(x_i) \\ \vdots \\ a_0 \sum_{i=1}^m \varphi_0(x_i) \varphi_n(x_i) + a_1 \sum_{i=1}^m \varphi_1(x_i) \varphi_n(x_i) + \dots a_n \sum_{i=1}^m \varphi_n(x_i) \varphi_n(x_i) = \sum_{i=1}^m y_i \varphi_n(x_i) \end{cases} \quad (5)$$

将式(5) 转化为式(6) 的形式

$$\begin{bmatrix} (\varphi_0, \varphi_0) & \dots & (\varphi_n, \varphi_0) \\ \vdots & \dots & \vdots \\ (\varphi_0, \varphi_n) & \dots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (y, \varphi_0) \\ \vdots \\ (y, \varphi_n) \end{bmatrix} \quad (6)$$

式(6) 的系数矩阵是一个对称的正定矩阵,所以它存在唯一解。文献[10] 选取代数多项式作为基函数拟合。通过解 $n+1$ 的线性方程组解得 $a_k (k = 0, 1, \dots, n)$ 。当给定数据点个数 m 较大或增加拟合阶数 n 时,系数矩阵容易呈现“病态” 且不稳定。本文在此基础上提出正交多项式最小二乘法拟合方法,有效的避免这种“病态” 发生。构造一个正交多项式族,递推公式如下:

$$\begin{cases} \varphi_0(x) = 1 \\ \varphi_1(x) = (x - \alpha_0) \varphi_0(x) \\ \vdots \\ \varphi_{k+1}(x) = (x - \alpha_k) \varphi_k(x) - \beta_k \varphi_{k-1}(x) \end{cases} \quad (7)$$

其中

$$\alpha_k = \frac{(x \varphi_k, \varphi_k)}{(\varphi_k, \varphi_k)}$$

$$\beta_k = \frac{(\varphi_k, \varphi_k)}{(\varphi_{k-1}, \varphi_{k-1})} \quad (8)$$

使得方程组(6)的系数矩阵中非对角元素全变为0,可简化为

$$\begin{bmatrix} (\varphi_0, \varphi_0) & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (y, \varphi_0) \\ \vdots \\ (y, \varphi_n) \end{bmatrix} \quad (9)$$

解得多项式系数

$$a_k = \frac{(y, \varphi_k)}{(\varphi_k, \varphi_k)} = \frac{\sum_{i=1}^m y \varphi_k(x_i)}{\sum_{i=1}^m \varphi_k(x_i) \varphi_k(x_i)} \quad (10)$$

代入式(3),得正交多项式.

4 SCEDV-Hop 算法

本文提出 SCEDV-Hop 算法首先采用多通信半径进行通信,细化节点间跳数值,同时利用正交多项式拟合方法修正未知节点和信标节点间估算距离误差.最后采取先相减然后再平方的策略求解定位方程组,求解未知节点坐标.最后利用解得的冗余信息第二次修正未知节点的位置坐标,使定位结果更加精准.

4.1 计算节点间最小跳数

信标节点不同时间的通信半径 R_i 和所发送的跳数值 e_i 根据式(11)得到.

$$e_i = \begin{cases} \frac{1}{n} & i = 1 \\ \vdots \\ \frac{j}{n} & i = j \\ \vdots \\ 1 & i = n \end{cases} \quad R_i = \begin{cases} \frac{1}{n}R & i = 1 \\ \vdots \\ \frac{j}{n}R & i = j \\ \vdots \\ R & i = n \end{cases} \quad (11)$$

第一步:初始化,信标节点并以泛宏方式向网络广播含有自身坐标、编号和跳数 e_1 等信息的数据包,此时通信半径为 R_1 . 邻居节点接收到数据包,与该信标节点的最小跳数记为 e_1 ,并统计记录所有距离自己 R_1 内的信标节点编号.为了减小通信开销,暂时不进行下一步转发.

第二步: t 时间过后,信标节点第二次同样广播数据包,此时通信半径为 R_2 . 如果第一步已经接收了该信标节点的信息,则保留原最小跳数;否则,记与该信标节点的最小跳数为 e_2 ,并记录所有距离自己 R_2 内的信标节点编号.同样暂时不进行下一步转发.

第三步:直至第 $(n-1)$ 步重复上述步骤. 第 n 步:信标节点最后一次广播数据信息,此时通信半径为 R . 如果前 $(n-1)$ 步已经接收了该信标节点的信息,则保留原最小跳数;否则,记与该信标节点的最小跳数为 e_n ,并将跳数值加1后,以通信半径 R 进行转发. 第 j 次泛洪接收节点转播信息过程中,同时转发记录的信标节点的编号信息. 如果接收节点与信标节点的编号对应上,则该信标节点需要将跳数值加 j/n 后再进行转播信息.

最后,所有未知节点都获得了与信标节点间最小跳数值记为 h .

4.2 正交多项式拟合修正距离误差

每个信标节点根据最小跳数计算自身平均跳距值,为了准确的拟合出跳数与距离误差,取所有信标节点平均跳距的平均值作为统一平均跳距,然后算出自身与其他信标节点间估计距离的误差:

$$\text{HopSize}_{\text{avg}} = \frac{\sum_{i=1}^n \text{HopSize}_i}{n} \quad (12)$$

$$\text{dist}_i = \text{HopSize}_{\text{avg}} \times h_{ij} - D_{ij} \quad (13)$$

同一跳数会存在多个值大小不一的距离误差,假设有 m 个跳数,一个跳数相应的距离差个数为 n 个,将这 n 个距离差累加求和再除以个数 n ,则获得每个跳数相应的平均距离差值:

$$\text{errdis}_m = \frac{\sum_{i=1}^n \text{dist}_i}{n} \quad (14)$$

将 m 个跳数和 m 个平均距离误差值作为已知数据 $(x_i, y_i) (i = 1, 2, \dots, m)$,利用最小二乘法拟合出跳数与距离误差对应关系的正交多项式(1).

未知节点利用已经获得的正交多项式,修正未知节点与信标节点之间的估算距离:

$$d_{ik} = \text{HopSize}_{\text{avg}} \times h_{ik} - \varphi(h_{ik}) \quad (15)$$

4.3 计算未知节点坐标

未知节点由式(15)获得与所有信标节点间的估算距离,采用改进的极大似然估计法确定未知节点坐标. 设 (x, y) 是未知节点坐标, (x_i, y_i) 是信标节点 i 的坐标,则:

$$\begin{cases} \sqrt{(x-x_1)^2 + (y-y_1)^2} = d_1 \\ \sqrt{(x-x_2)^2 + (y-y_2)^2} = d_2 \\ \vdots \\ \sqrt{(x-x_n)^2 + (y-y_n)^2} = d_n \end{cases} \quad (16)$$

为了使在方程中修正因子较小,有更好的定位精度,用前 $n-1$ 个方程分别减去第 n 个方程,然后两边平方^[9]. 简化整理,得到: $-2(x_i + x_n)x - 2(y_i + y_n)y + 2S = d_1^2 + d_n^2 - (E_1 + E_n)$ 其中 $S = x^2 + y^2$, $E_i = x_{i2} + y_{i2}$

写成矩阵形式 $\mathbf{A}\mathbf{X} = \mathbf{b}$,

$$\mathbf{A} = -2 \times \begin{bmatrix} x_1 + x_n & y_1 + y_n & -1 \\ x_2 + x_n & y_2 + y_n & -1 \\ \vdots & \vdots & \vdots \\ x_{n-1} + x_n & y_{n-1} + y_n & -1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ S \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} d_1^2 + d_n^2 - (E_1 - E_n) \\ d_2^2 + d_n^2 - (E_2 - E_n) \\ \vdots \\ d_{n-1}^2 + d_n^2 - (E_{n-1} - E_n) \end{bmatrix}$$

求解出

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A} \mathbf{b} \quad (17)$$

在求解未知节点坐标过程中,应充分考虑不同信标节点的影响,对于与自身位置越近,与自身距离误差越小的信标节点,未知节点应重点参考其坐标,给予其较大的加权值. 因此,未知节点到信标节点的加权值可设定:

$$w_{ki} = \left(\frac{1}{\varphi(h_{ik})} \right)^2 + \left(\frac{1}{D_{ki}} \right)^2$$

最终构造加权矩阵 $\mathbf{W}$ 为:

$$\mathbf{W} = \begin{bmatrix} w_{k1} & 0 & \cdots & 0 \\ 0 & w_{k2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{k,n-1} \end{bmatrix}$$

因此可以将式(17)进一步改进为

$$\mathbf{X} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b} \quad (18)$$

4.4 修正未知节点坐标

由式(18)解得未知节点坐标 (x, y) 和 S 后,若满足 $S = x^2 + y^2$,即为所求坐标;若不满足 $S = x^2 + y^2$,则令:

$$\begin{cases} x_1 = ax \\ y_1 = ay \end{cases} \quad (19)$$

代入 $S = x_1^2 + y_1^2$, 解出参数 a , 得到 x_1 和 y_1 的值,则最终修正得到的未知节点的坐标为:

$$\begin{cases} X = \frac{x_1 + x}{2} \\ Y = \frac{y_1 + y}{2} \end{cases} \quad (20)$$

5 仿真实验及分析

利用 MATLAB 对 SCEDV-Hop 算法与原 DV-Hop 算法、文献[8-9]中改进算法进行实验仿真,验证 SCEDV-Hop 算法的有效性和可行性. 设置节点随机均匀分布在 $100 \text{ m} \times 100 \text{ m}$ 的正方形区域内,进行 100 次仿真实验,实验结果取平均值. 本文选取未知节点平均相对定位误差作为算法性能评价指标,表示为:

$$\text{error} = \frac{\sum_{i=1}^{N-n} \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2}}{(N-n) \times R}$$

式中, N 为节点总数, n 为信标节点数, R 为节点的通信半径, (x_i, y_i) 和 (X_i, Y_i) 为未知节点 i 的估计坐标和真实坐标.

由上式可知,未知节点平均定位误差取决于节点总数,信标节点数和节点的通信半径的大小. 由此根据仿真实验结果讨论这三个方面对未知节点平均定位误差的影响.

节点总数为 100 个,其中信标节点比例为 10% 的节点随机分布如图 1 所示.

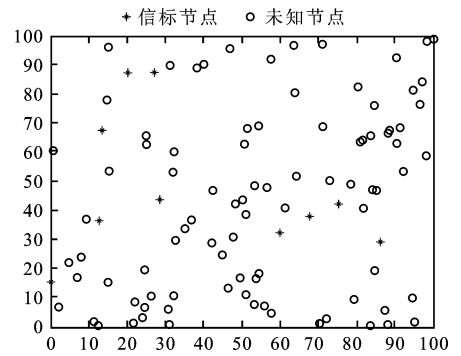


图 1 未知节点与信标节点位置分布

5.1 信标节点数对平均定位误差的影响

图 2 表示通信半径为 30 m, 节点总数为 300 个, 信标节点比例由 5% 变化至 30% 四种算法的平均定位误差曲线对比图. 如图所示, 随着信标节点个数的增加, 四种算法的平均定位误差都在减小, 且 SCEDV-Hop 算法当信标节点数比例达到 20% 以后, 较文献[9]优势显现, 这是由于信标节点增加使

得信标节点密度增大,获得的跳数值更加精准,同时跳数与距离误差的数据样本增多,拟合结果更加准确,更好的对距离误差进行修正.在节点总数和通信半径不变情况下,随着信标节点数目增加,可看出 SCEDV-Hop 算法较其他三种算法平均定位误差分别降低 38%,20%,11%.

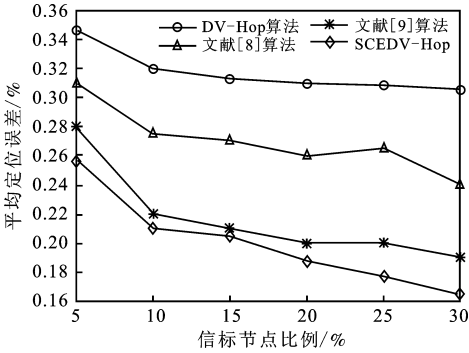


图2 平均定位误差与信标节点数的关系

5.2 通信半径对平均定位误差的影响

图3表示节点总数为300个,信标节点数为30个,通信半径由15 m变化为40 m的四种算法平均定位误差曲线对比图.如图所示,文献[9]算法当通信半径达到一定值后继续增大时,信标节点通信范围内未知节点增多,会增大估计距离的误差,从而平均定位误差先将后升. SCEDV-Hop 算法中修正了产生的误差,避免了文献[9]出现的定位误差上升的情况,但不断增大的通信半径使得信标节点间的跳数减小,获得的跳数与距离误差的数据样本减少,拟合结果对距离误差的修正效果有限,优化优势不再明显.在节点总数和信标节点数不变的情况下,随着通信半径的增加,可看出 SCEDV-Hop 算法较其他三种算法平均定位误差降低了38%,26%,27%.

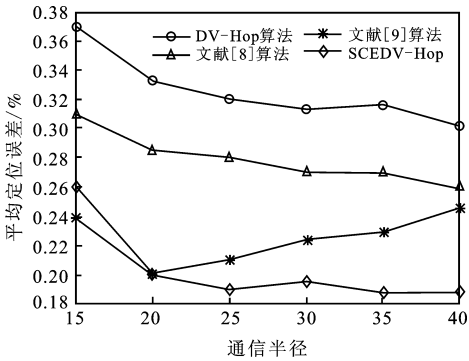


图3 平均定位误差与通信半径的关系

5.3 节点总数对平均定位误差的影响

图4表示通信半径为30 m,信标节点数为节点总数的10%,节点总数由100个增加到400个的四

种算法平均定位误差曲线对比图.如图所示,当节点总数不断增多时,三种算法的平均定位误差减小.这是由于当区域面积节点总数一定时,增加节点总数,会提高节点密度,提高网络连通度,从而使得定位精度提高.当节点总数不断增多,可看出 SCEDV-Hop 算法较其他三种算法平均定位误差降低了36%,18%,9%.

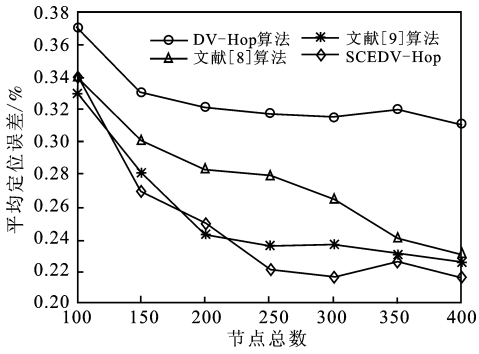


图4 平均定位误差与节点总数的关系

6 结束语

本文提出基于二次修正误差的 SCEDV-Hop 算法.采用多通信半径进行通信,细化节点间跳数值.同时采用正交多项式拟合方法修正节点间估算距离产生的误差.最后采取先相减然后再平方的策略求解未知节点坐标.并且利用解得的冗余信息再一次修正未知节点坐标.通过 Matlab 仿真实验,与经典的 DV-Hop 算法和其他两种算法作对比,充分验证了本文算法在确定未知节点坐标时可以有效降低定位误差,提高定位精确度.

参考文献:

[1] Woo H, Lee S, Lee C. Range-free localization with isotropic distance scaling in wireless sensor networks [C]// International Conference on Information Networking. Chiang Mai, Thailand, IEEE Computer Society, 2013:632-636.

[2] 夏少波, 邹建梅, 朱晓丽, 等. 基于跳数区域划分的 DV-Hop 改进算法[J]. 传感技术学报, 2014, 36(7): 964-969.

[3] 邴晓瑛, 徐保国. 基于多通信半径的加权 DV-Hop 改进算法[J]. 计算机应用与软件, 2016, 33(7): 125-128.

[4] 景露露, 张玲华. 基于跳距优化的改进型 DV-Hop 定位算法[J]. 传感技术学报, 2017, 30(4): 582-586.

由于 Rocket-Chip 本身是面向 ASIC 电路的,因此生成 Vivado 的 FPGA 工程的时候时钟频率并不高. 根据 UCB 的流片结果,ASIC 的 Rocket-Chip 的频率是大于 200 MHz 的,所以我们在 vivado 上分析的更多是针对加速器进行.

通过综合及实现,将提供的 PLL 时钟频率设为 100 MHz 时,查看时序报告可得,仍有 2.404 ns 的余量,因此加速器实际最高频率为 131.65 MHz.

相应的,电路的资源消耗见表 2

表 2 FPGA 上资源消耗表	
LUTs	109 875
FFs	84364
DSPs	429
BRAMs	23

由于 FPGA 上的 Verilog 代码是由 Chisel 直接编译生成,因此消耗了大量的 LUTs. 接下来改进的展望是使用乘法器 IP,通过使用 DSP 来降低 LUT 的占用.

5 结束语

本文中,我们基于 Rocket-Chip 开源处理器提出了一种卷积神经网络的加速结构. 该结构通过构建一维卷积器的 PE 阵列,从而在横向,竖向和斜向上分别对卷积核,计算结果以及输入特征图像进行复用,从而减少访存的次数. 更进一步,优化了卷积层的计算顺序,尽可能使得输入特征图的利用率达到最高. 并且,通过 Chisel 语言编写的加速器在与 Rocket-Chip 结合后,生成的 C++ 模拟器上的仿真也证明了该结构的加速作用. 同时,该加速器的 FPGA 也证明了在资源消耗在合理的范围内,能获得良好的工作频率.

参考文献:

(上接第 16 页)

[5] Salama M, Kandil M. An Improved DV-hop localization algorithm based on modified hop-size[C]// Computer Applications & Research. Qingdao, China, IEEE, 2016:83-86.

[6] Mehrabi M, Taheri H, Taghdiri P. An improved DV-Hop localization algorithm based on evolutionary algorithms [J]. Telecommunication Systems, 2017, 64 (4) :1-9.

[7] 肖晓丽,李旦江,谭柳斌. 基于布谷鸟搜索算法的无线传感器网络节点定位[J]. 计算机工程与应用, 2017, 53(2):141-145.

[8] 邱奉美,李怀忠. 无线传感器网络 DV-Hop 定位算法的改进[J]. 计算机工程, 2014, 40(8):15-20.

[1] Waterman A, Lee Y, Patterson D A, et al. The risc-v instruction set manual, volume i: Base user-level isa [J]. Eecs Department, 2011, 7(9):475

[2] Asanovic K, Avizienis R, Bachrach J, et al. The rocket chip generator[R]. EECS Department, University of California, Berkeley, 2016.

[3] Chen Y H, Krishna T, Emer J S, et al. Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks[J]. IEEE Journal of Solid-State Circuits, 2017, 52(1): 127-138.

[4] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[5] 陆志坚. 基于 FPGA 的卷积神经网络并行结构研究[D]. 哈尔滨:哈尔滨工程大学, 2013.

[6] Chen Y H, Emer J, Sze V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks[C]//Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on. Seoul, South Korea,IEEE, 2016: 367-379.

[7] 方睿,刘加贺,薛志辉,等. 卷积神经网络的 FPGA 并行加速方案设计[J]. 计算机工程与应用, 2015, 51 (8):32-36.

作者简介:

杨维科 男,(1993-),硕士研究生. 研究方向为可重构处理器的系统架构. E-mail:wakefielddyoung@sjtu.edu.cn.

贺光辉 男,(1980-),博士,副研究员. 研究方向为通信系统、VLSI 等.

景乃锋 男,(1982-),博士,副研究员. 研究方向为计算机体系结构.

作者简介:

马肖旭 女,(1990-),硕士研究生. 研究方向为计算机网络与安全. E-mail:1656871571@qq.com.

刘文菊 女,(1963-),教授. 研究方向为计算机网络与安全、企业信息化.

王 赟 男,(1976-),教授. 研究方向为计算机网络与安全、企业信息化.