

文章编号:1001-9081(2007)02-0342-04

基于分簇的无线传感器网络节能路由算法

王春雷¹, 柴乔林¹, 王 华², 于磊磊¹, 刘 鑫¹

(1. 山东大学 计算机科学与技术学院, 山东 济南 250061;

2. 山东中医药大学 网络中心, 山东 济南 250014)

(wangchunlei@mail.sdu.edu.cn)

摘 要:针对传感器网络能量有限的特点,提出一种基于分簇的节能路由算法—CESR 算法,算法采用逐步求解的方式,将整个网络分成若干个交叠簇,整个网络有簇间和簇内两级拓扑。节点在选择路由时先根据簇间拓扑选择簇路由,当数据经过某个簇进行转发时,边界节点根据簇内拓扑产生簇内路由,算法尽量选择消耗能量值最小的路径转发数据,并且能够避免低能量节点,算法存储开销和控制消息相对较少。

关键词:无线传感器网络;簇;簇间路由;控制消息;OMNET++ 仿真

中图分类号: TP393 **文献标识码:** A

Cluster-based energy-saving routing algorithm for wireless sensor network

WANG Chun-lei¹, CHAI Qiao-lin¹, WANG Hua², YU Lei-lei¹, LIU Xin¹

(1. School of Computer Science & Technology, Shandong University, Jinan Shandong 250061, China;

2. Network Center of Shandong University of Traditional Chinese Medicine, Jinan Shandong 250014, China)

Abstract: A new Cluster-based Energy Saving Routing (CESR) for energy-limited wireless sensor network was proposed. In this protocol, the network was divided into overlapping clusters; each node only knew the node connectivity within its cluster and the cluster connectivity of the whole network. The global path was selected according to the cluster level topology. The node in the overlap region computed the local path and decided how to send the data to the next cluster. The algorithm tried to find the path with the minimum power consumption and can avoid the node with low energy. The storage and communication overhead was reduced.

Key words: wireless sensor network; cluster; cluster-level routing; control message; OMNET++ simulation

0 引言

随着微机电技术、计算机技术和通信技术的发展与融合,具有通讯能力的微型传感器出现,它们被布置在一个特定的区域内,形成一个无线网络,相互协作,感知、采集、处理区域内感知对象的信息,这就是传感器网络。传感器网络可以广泛应用在如国防军事、环境监测、交通管理、医疗卫生、反恐救灾等领域,成为近几年国内外研究的热点^[1]。

与传统网络相比,传感器网络具有以下特点:1)节点分布极其稠密且数目很大,每个节点维护全局信息是不可能的;2)节点的能量、存储空间及计算能力等资源非常有限;3)传感器节点布置完毕后,除了少数节点需要移动以外,大部分节点都是静止的^[2]。因此,传统的路由算法并不适合于无线传感器网络,必须针对其特性研究新的路由算法。

由于无线传感器网络通常工作在人无法接近或者高危险区域,使得随时更换节点能源是非常困难的,因此路由算法的节能性就成为研究人员关注的焦点,并提出了多种新颖的路由算法。

1 现有算法分析

先对无线传感器网络模型进行描述,无线传感器网络的

拓扑可看作是由节点和链路构成的图,记为 $G = (V, E)$ 。其中, V 表示节点的集合, E 表示边的集合。节点 x 和 y 之间存在边 (x, y) , 意味着节点 x 和节点 y 是可以互相通信的 1 跳邻居节点。 $R_x = x, n_0, n_1, \dots$, S 表示节点 x 到 Sink 节点的一条路径, R_* 为节点 x 到 Sink 节点的所有路径的集合。

1.1 MTPR 算法

MTPR (Minimum Total Transmission Power Routing) 算法基本思想是选择消耗能量值最小的路径传送数据。令 $R_d = n_0, n_1, \dots, n_d$ 表示从源节点 n_0 到目的节点 n_d 的路径(无线传感器网络中 n_d 为 Sink 节点), $E(n_i, n_j)$ 为节点 n_i 将数据传给 n_j 所要消耗的能量,则路径 R_d 能量总消耗值计算公式为:

$$P(R_d) = \sum_{i=1}^{d-1} E(n_i, n_{i+1})$$

MTPR 算法选择的路径 R_{MTPR} 满足:

$$R_{MTPR} = \min_{R_j \in R_*} P(R_j)$$

MTPR 算法虽然选择消耗能量值之和最小的路径,但是有可能该路径中的某个节点剩余能量值已经很小,造成该节点过度使用^[3]。

1.2 MMBCR 算法

MMBCR (Minimum Max Battery Cost Routing) 算法能够避

收稿日期:2006-08-17;修订日期:2006-11-01 基金项目:山东省信息产业攻关项目(2003118)

作者简介:王春雷(1982-),男,山东聊城人,硕士研究生,主要研究方向:无线传感器网络;柴乔林(1956-),男,山东青岛人,教授,主要研究方向:计算机网络;王华(1980-),女,山东临朐人,助教,主要研究方向:计算机网络;于磊磊(1982-),男,山东省聊城人,硕士研究生,主要研究方向:无线传感器网络 QoS;刘鑫(1982-),男,山东济南人,硕士研究生,主要研究方向:无线传感器网络拥塞控制。

开剩余能量值小的节点,用 $P_i(t)$ 表示在 t 时刻节点 n_i 的剩余能量,则对于路径 R_d 定义权值 $P(R_d) = \min_{n_i \in R_d} P_i(t)$,MMBCR 算法选择路径 R_{MMBCR} 满足 $P(R_{MMBCR}) = \max_{R_j \in R_*} P(R_j)$ 。MMBCR 算法能够避开能量值很小的节点,增加了网络的公平性,但是该算法却不能保证每次传输数据消耗的能量最少,而有可能增加网络的能量消耗,从而导致网络的生存周期缩短^[4]。

1.3 CMMBCR 算法

CMMBCR (Conditional MMBCR) 综合了以上两种算法,其思想如下:节点 x 向 Sink 节点发送数据时,搜索所有能够达到 Sink 节点的路径,若路径上节点的剩余能量都很充足,则按照 MTPR 算法选择消耗能量值之和最小的路径。而如果路径上节点的剩余能量值都较小,就按照 MMBCR 算法选择路径。这一过程可以通过为网络节点增设阈值来实现,如果某一节点的剩余能量小于该阈值,则在选择路由时尽量避开该节点,来延长整个网络的生存周期。该算法既考虑了总的传输能量又考虑到了网络中节点的剩余能量^[5]。

但是,上述算法都很难在大规模无线传感器网络中实现,主要问题在于节点发送数据之前先确定数据要沿哪条路径进行转发,而节点在搜索转发路径时,需要知道整个网络的拓扑结构,以及每个节点的剩余能量值^[6],而网络中每个节点的能量都在不断变化,要得到每个节点确切的剩余能量值是很困难的。当然节点可以向其他节点广播自己的剩余能量值,但这种方式会增加大量的控制消息,而且网络规模越大,控制消息越多,同样会缩短网络的生存周期。因此,针对大规模无线传感器网络,提出了一种基于分簇的节能路由算法 (Cluster-based Energy Saving Routing, CESR),该算法是 CMMBCR 的近优算法,而存储开销和控制消息却相对少得多。

2 算法设计

算法的基本思想是采用逐步求解的方式,先将整个网络分成若干个重叠簇,把一个簇看成一个整体,每个簇都有一个能量度量值(算法中采用簇内节点的平均能量值),这样簇之间也形成了一个网络拓扑结构,整个网络存在两级拓扑,簇间拓扑,及簇内拓扑。节点在选择路由时先根据簇间拓扑选择簇路由,确定发送的数据要经过哪几个簇进行转发,当数据经过某个簇时,重叠节点根据簇内拓扑,确定如何将信息传送到下一个簇,即产生簇内路由。

算法的前提是将网络分成若干个相互重叠的簇,若两个簇之间有重叠节点,则认为两个簇是连通的,簇之间的数据通过重叠节点进行转发,通常两个簇之间有多个重叠节点,这样就形成了多网关机制,重叠节点共同承担转发数据的任务。簇头节点只负责维护簇内拓扑信息,以及收集簇内节点的剩余能量值,而不转发任何数据,这就避免了传统的算法只通过簇头转发数据,而造成簇头节点能量消耗过快,而频繁更换簇头节点的缺陷。具体的重叠簇分簇算法参考文献[7]。

分簇工作完成之后,算法可以归结为以下三个问题:1) 簇头如何计算簇的剩余能量值;2) 节点如何选择簇间路由;3) 节点如何选择簇内路由。

2.1 计算簇的剩余能量值

当分簇稳定之后,簇内的每个节点都向自己的簇头报告自己的剩余能量值,当簇头节点收集了簇内各节点的剩余能量值之后,计算其平均值,作为簇的能量值的标识,然后向其他所有簇广播,簇头收集到其他簇的信息之后,形成簇间拓

扑,然后向簇内所有节点广播,这样网络中每个节点都有了簇拓扑图。由于节点的剩余能量值在不断变化,所以簇的剩余能量值也会随时变化,这就需要簇头节点周期性地收集簇内所有节点的剩余能量值,从而重新度量簇的能量值。当剩余能量值比较大时,局部节点能量的变化不会对整个网络造成太大的影响,所以这时更新周期可以长一些,只有当网络中每个节点的剩余能量值都比较小时,信息传输会对网络的能量分布造成比较明显的影响,这时更新周期相应地延长。

簇的剩余能量值计算完成之后,这时如果把簇看成一个整体,那么所有的簇就形成了一个簇的拓扑图,例如图1中(a)中所示为一个网络拓扑图,有A,B,C,D,E五个簇,括号中的数字代表每个簇内节点的平均剩余能量值,那么图1(a)中的网络所对应的簇拓扑图如图1(b)所示。

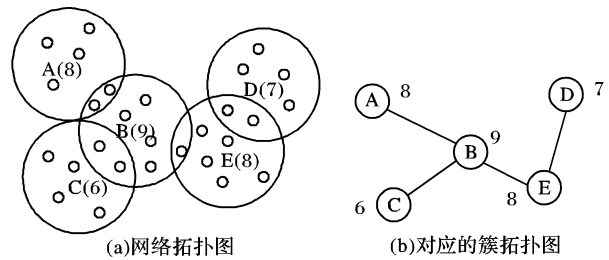


图1 簇间拓扑

2.2 簇间路由

当节点有数据要向 Sink 节点发送时,该节点首先要计算簇间路由,即该消息要经过哪几个簇进行转发。由于簇拓扑图中的边没有权值,所以簇间路由算法采用 MMBCR 算法,尽量选择剩余能量值较大的簇转发数据,下面采用改进的 Bellman-Ford 算法计算簇路由。算法描述如下:

网络拓扑图 $G(V, E)$, 对 $v \in V$, $P(v)$ 表示 v 的剩余能量值, $V[G]$ 表示图中顶点的个数, $E[G]$ 表示图中边的条数。寻找路径 $s = v_0, v_1, v_2, \dots, v_{k-1}, v_k = t$ (t 为 Sink 节点所在的簇) 使 $\min_{i=1}^{k-1} P(v_i)$ 最大。

```

for (each vertex  $v \in V[G]$ ) //算法初始化
{
  //对于图中的每个顶点
  If ( $edge(s, v) \in E[G]$ ) //如果  $s$  到  $v$  存在边
  {
     $d[v] = INFINITE$ ;  $n[v] = s$ ;
  }
  else  $\{d[v] = 0; n[v] = NULL\}$ ;
}
 $d[s] = INFINITE$ ; // #define INFINITE 65536
for ( $i = 1; i < |V[G]|; i++$ )
{
  for (each edge  $(u, v) \in E[G]$  and  $u \neq s$ )
  {
    //对网络中的每条边
    if ( $d[v] < \min(d[u], P[u])$ )
    {
      //如果  $d[v]$  比  $d[u], P[u]$  都小
       $d[v] = \min(d[u], P[u])$ ;
       $n[v] = u$ ;
    }
  }
}
 $i = n[t]$ ;
while ( $n[i] \neq s$ )
{
  //Path 为从  $s$  到  $t$  的路由
  push(Path,  $n[i]$ );  $i = n[i]$ ;
}

```

当计算好簇路由之后,形成一个簇路由表,数据包携带簇路由信息向目标传送。

2.3 簇内路由

簇内路由决定一个数据包在簇内经过哪些节点进行转发,当边界节点收到来自其他簇的数据包之后,首先根据数据包携带的簇路由表,确定该数据包将要转发到哪一个簇,将该簇与下一个簇的一个重叠节点作为目标节点,对簇内的网络拓扑运用 CMMBCR 算法,计算数据包的转发路径,然后将数据包沿此路径进行转发。如图 2 所示,数据包要经过簇 A、B、C 进行转发,节点 1 接收到来自 A 簇的数据包后,首先根据数据包中的簇路由信息,确定要将数据包转发到 C 簇,节点 1 根据簇内拓扑图选择一个 B 与 C 的重叠节点作为目的节点(节点 4),然后利用 CMMBCR 算法计算到节点 4 的路径 1,2,3,4,数据包沿此路径进行转发到达节点 4,节点 4 以同样的方法将数据包向下一个簇进行转发,直到转发到目标节点所在的簇(目标节点一般为 Sink 节点)。一般情况下,两个簇的重叠节点不止一个,这时在选择重叠节点进行转发数据时,选择剩余能量值最大的一个。

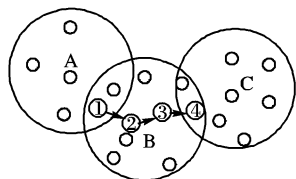


图 2 簇内数据转发

一个簇的一个重叠节点作为目标节点,对簇内的网络拓扑运用 CMMBCR 算法,计算数据包的转发路径,然后将数据包沿此路径进行转发。如图 2 所示,数据包要经过簇 A、B、C 进行转发,节点 1 接收到来自 A 簇的数据包后,首先根据数据包中的簇路由信息,确定要将数据包转发到 C 簇,节点 1 根据簇内拓扑图选择一个 B 与 C 的重叠节点作为目的节点(节点 4),然后利用 CMMBCR 算法计算到节点 4 的路径 1,2,3,4,数据包沿此路径进行转发到达节点 4,节点 4 以同样的方法将数据包向下一个簇进行转发,直到转发到目标节点所在的簇(目标节点一般为 Sink 节点)。一般情况下,两个簇的重叠节点不止一个,这时在选择重叠节点进行转发数据时,选择剩余能量值最大的一个。

3 算法分析与仿真实验

3.1 算法分析

在大型的无线传感器网络中,CESR 算法比 CMMBCR 算法有明显优势,其存储开销小,控制消息少。下面以一个实际的网络模型说明两种算法的差别,假如网络中有 300 个节点,网络分为 20 个簇,每个簇内平均 20 个节点(因为簇间有重叠,一个节点可能同时属于多个簇)。

对于 CESR 算法,每个节点需要存储簇间拓扑和所在簇的簇内拓扑两个拓扑图,但这要比存储整个网络拓扑图小得多。对上述网络中,每个普通节点需要存储有 20 个顶点的簇间拓扑,以及 20 个顶点的簇内拓扑。而对于 CMMBCR 算法每个节点都要存储有 300 个顶点的整个网络拓扑。

两种算法都周期性地更新网络拓扑图。在上述的网络中,对 CMMBCR 算法,在每次更新时,每个节点都需要向网络中其他所有节点广播自己的剩余能量值,所以会产生 300×300 条控制消息。而对于 CESR 算法,每个节点向簇头发布自己的剩余能量值,簇头收集簇内所有节点剩余能量之后,重新形成簇内拓扑,然后将簇内拓扑图在本簇内广播,所以每个簇内会产生 40 条控制消息,整个网络内会形成 20×40 条控制消息。簇头根据簇内节点的剩余能量值计算簇的平均剩余能量值,然后向其他的簇头广播,这样会形成 20×20 条控制消息,而每个簇头需要将形成的簇内拓扑向簇内所有节点广播,又会形成 20×20 条控制消息,所以对 CESR 算法,每次更新网络拓扑图时将会产生 1600 条控制消息,这要比 CMMBCR 算法中控制消息少得多。

由于 CESR 算法中,簇内消息不向其他簇传播,对节点级造成的移动影响限制在本簇区域,只有当节点移动改变了簇间的连接时才在簇间传播消息,这样限制了由于节点移动对网络拓扑结构的影响。

3.2 仿真实验

下面采用 OMNET++ 对算法进行仿真,并与 CMMBCR 算法进行比较。在 OMNET++ 实验环境中,设置网络图范围为 500×500 ,随机分布 71 个节点(1 个 Sink 节点),节点通信半

径为 50,每个节点的初始能量值为 2000(计算单位使用 OMNET++ 仿真环境规定),节点 n_i 与 n_j 传输数据消耗的能量 $E(n_i, n_j)$ 与其距离 $D(n_i, n_j)$ 的关系为: $E(n_i, n_j) = 0.01 \times D^2(n_i, n_j)$,采用的传输信道数据传输率为 250kb、出错率为 0、信道延迟为 0.01s,数据包长度为 128bit。实验中,从第 10s 开始,每 10s 为一个周期,每个周期内每个节点向 Sink 节点发送一个数据包,若某个节点的能量低于初始能量的 30%,则认为该节点为低能量节点,数据不再通过该节点转发,对于 CESR 算法采用了简单的基于地理位置的分簇方法。仿真结果如图 3 ~ 图 5 所示。

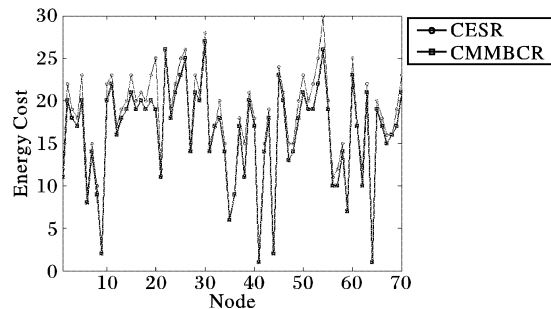


图 3 路径能量消耗

图 3 表示了某一周期内,每个节点向 Sink 节点发送数据所需要消耗的能量值,由图可以看出,CESR 算法中,节点向 Sink 节点发送一个数据包消耗的能量值略高于 CMMBCR 算法中消耗的能量值,但比较接近。

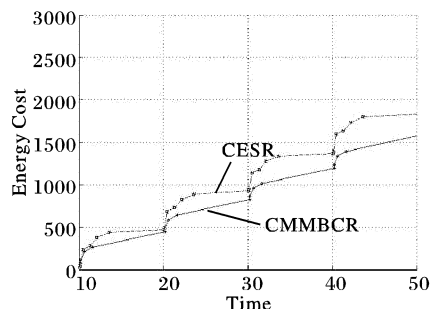


图 4 不考虑控制消息的能量消耗

图 4 显示了如果不考虑控制消息(即控制消息不消耗能量),50s 内两种算法能量消耗情况,可以看出,在不考虑控制消息的情况下,CESR 算法在能量消耗方面比 CMMBCR 算法要多一些,但是比较接近。由以上两图可以看出,在不考虑控制消息的情况下,CESR 算法是 CMMBCR 的近优算法,两种算法在节能方面都表现出良好性能。

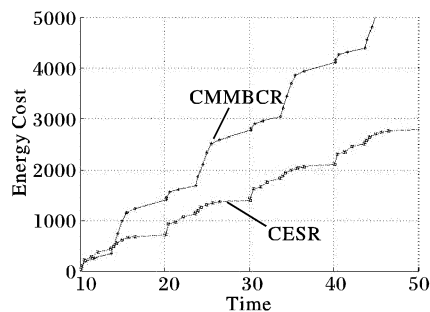


图 5 考虑控制消息的能量消耗

图 5 所示为考虑控制消息时两种算法的能量消耗,实验中,每传送完一轮数据之后都进行拓扑图的更新,由图可以看出,对于 CMMBCR 算法,其能量消耗呈现快速上升趋势,而 CESR 算法中由于控制消息少,其能量消耗是一种平缓上升

的趋势。由此可见,在考虑控制消息的情况下,CESR 算法在节能方面明显优于 CMMBCR 算法。

为了说明 CESR 算法与普通分簇算法的区别,在实验中与最小 ID 分簇算法 (Lowest-ID Cluster Algorithm) 进行了比较,同样是上述网络,采用最小 ID 分簇算法,相邻节点中具有最小 ID 的节点作为簇头,其一跳邻居节点成为该簇头所在簇的成员节点,并不再参与簇头选举过程^[8],所有数据都通过簇头进行转发,实验中不考虑分簇过程的开销。实验结果如图 6 所示。

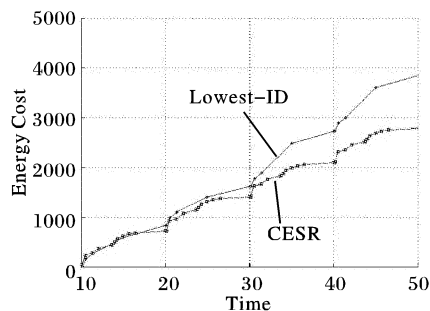


图6 能量消耗情况

由图 6 可以看出,一开始两种算法在能量消耗方面相差不多,但在 30s 之后,由于最小 ID 分簇算法中簇头负担过重,有的簇头节点能量不足,而进行了簇头的重新选举,比 CESR 算法消耗了更多的能量,而 CESR 算法中重叠节点共同承担了数据转发的任务,使整个网络负载平衡,没有出现重新选举簇头的情况。

在另一次实验中,随机放置了 1000 个节点,节点初始能量为 2000,对 CESR 算法,采用了最简单的基于地理位置的分簇算法进行分簇,总共分为 40 个簇,由于簇之间有重叠,每个

簇内平均有 40 个节点。整个仿真过程分为若干个周期,在每个周期中,网络中的每个节点都向 Sink 节点发送一个数据包,每个周期结束,重新计算簇的平均剩余能量值。如果网络中任意一个节点能量耗尽时,则结束仿真过程,经过仿真,在第一个节点能量耗尽时,Sink 节点接收到 39000 个数据包,而采用 CMMBCR 算法只收到 13000 个数据包,由此可见 CESR 算法在大型传感器网络中具有更明显的优势。

参考文献:

- [1] 李建中, 李金宝, 石圣飞. 传感器网络及其数据管理的概念、问题与进展[J]. 软件学报, 2003, 14(10): 1717 - 1727.
- [2] AKYILDIZ IF, SU W, SANKARASUBRAMANIAM Y, et al. A Survey on Sensor Networks[J]. IEEE Communications Magazine, 2002, 40(8): 102 - 114.
- [3] SINGH S, WOO M, RAGHAVENDRA CS. Power-aware routing in mobile ad hoc networks[J]. Proceedings of MobiCom[C], 1998. 181 - 190.
- [4] LEE M, JING XP. Energy-efficient Routing Protocols in Wireless Ad hoc Networks[R]. Research Project Report. Spring, 2001.
- [5] TOH CK. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks[J]. IEEE Communication Magazine, 2001, 39(6): 138 - 147.
- [6] MISRA A, BANERJEE S. MRPC: Maximizing network lifetime for reliable routing in wireless environments[J]. IEEE Wireless Communications and Networking Conference(WCNC)[C]. Orlando, Florida, 2002. 800 - 806.
- [7] 郑少仁, 王海涛, 赵志峰, 等. Ad Hoc 网络技术[M]. 北京: 人民邮电出版社, 2005.
- [8] CHIAN C, GERLA M, ZHANG L. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel[J]. In Proceedings of IEEE SICON'97[C]. 1997. 197 - 211.

(上接第 341 页)

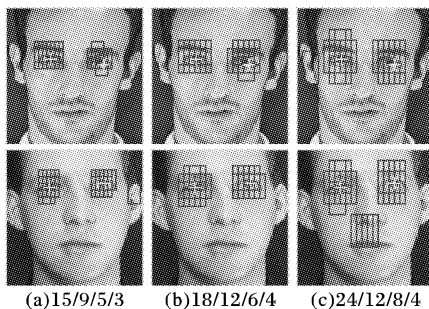


图6 不同参数设置的实验结果



图7 候选点对比示意图

与传统算法相比,本算法的候选点更接近真实的眼睛中心,这样邻接的点对就会被快速排除而为后面的眼睛对筛选缩小了范围。从计算时间上分析,实验表明整个算法的主要时间开销集中在眼睛定位上,这主要是由于眉眼区域初定位

时只要遍历一次图像便可得到积分图,为以后的计算节省了时间。实验结果表明,对于条件较好的人脸正面图像,本算法的定位准确率可以达到 96% 以上,在没能成功定位的图像中,主要原因包括光照条件太差、戴眼镜反光、眼睛闭上或太小等干扰,这时采用传统的对称变换方法也不能成功定位。

参考文献:

- [1] 林维训, 潘纲, 吴朝晖, 等. 脸部特征定位方法[J]. 中国图象图形学报, 2003, 8(8): 849 - 859.
- [2] REISFELD D, WOLFSON H, YESHURUN Y. Context-Free attentional operators: the generalized symmetry transform[J]. International Journal of Computer Vision, 1995, 14(2): 119 - 130.
- [3] REISFELD D, YESHURUN Y. Preprocessing of face images: detection of features and pose normalization[J]. Computer Vision and Image Understanding, 1998, 71(3): 413 - 430.
- [4] 周杰, 卢春雨, 张长水. 基于方向对称变换的人脸定位方法[J]. 电子学报, 1999, 27(8): 12 - 15.
- [5] 刘文予, 潘峰. 离散对称变换在人脸图像眼睛定位中的应用[J]. 红外与毫米波学报, 2001, 20(5): 375 - 380.
- [6] PAUL V, MICHAEL J. Rapid object detection using a boosted cascade of simple features[J]. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition[C]. Kauai, Hawaii, USA, 2001. 511 - 518.