

# 基于群代理的 QoS-AP 算法优化 QoS 组播路由

李睿<sup>1</sup> 余勇<sup>2</sup>

<sup>1</sup>(阿坝师范高等专科学校网络管理中心 四川 汶川 623002)

<sup>2</sup>(河南经贸职业学院信息管理系 河南 郑州 450053)

**摘要** 针对互联网组播应用中多约束服务质量(QoS)组播路由优化问题,提出一种基于群代理的融合蚁群(ACO)算法与粒子群优化(PSO)算法的 QoS-AP 算法。首先根据 QoS 约束,产生多个组播模型。然后利用 ACO 算法对每个模型和模型中的属性进行评估并放置信息素。再根据信息素值,利用 PSO 算法调整粒子代理的运动方式来重组组播树。经过多次迭代,最后形成一个满足 QoS 的最优组播树。通过仿真实验,与现有的 PSOTREE、TGBACA 算法进行比较。结果表明,该算法能够找出更好的组播树模型,不仅能够满足 QoS 约束,而且还最大限度地减少了树的成本。

**关键词** 多约束 QoS 组播路由 群代理 蚁群算法 粒子群算法

中图分类号 TP3 文献标识码 A DOI:10.3969/j.issn.1000-386x.2015.09.031

## OPTIMISING MULTICAST ROUTING OF QOS WITH SWARMING AGENT-BASED QOS-AP

Li Rui<sup>1</sup> Yu Yong<sup>2</sup>

<sup>1</sup>(Network Management Center, ABA Teacher's College, Wenchuan 623002, Sichuan, China)

<sup>2</sup>(Department of Information Management, Henan Economic and Trade Vocational College, Zhengzhou 450053, Henan, China)

**Abstract** For the optimisation issue of multicast routing of quality of service (QoS) with multi-constraint in Internet multicast applications, we propose a QoS-AP algorithm, which is based on swarming agents and integrates ant colony optimisation (ACO) and particle swarm optimisation (PSO). First, according to QoS constraints it generates several multicast models. Secondly, it evaluates every model and the attribute of each model using ACO algorithm, and places the pheromones as well. Thirdly, according to the pheromone value, it reconstructs the multicast tree by adjusting the motion ways of particle agent through PSO. After several times of iteration, finally it forms an optimal multicast tree satisfying the QoS. Through simulation experiment, QoS-AP is compared with existing PSOTREE and TGBACA algorithms. Results show that, the algorithm is able to find better multicast trees model, and can meet the QoS constraints as well as minimise the cost of the tree.

**Keywords** Multicast routing of QoS with multi-constraint Swarming agents Ant colony optimisation Particle swarm optimisation

## 0 引言

随着网络多媒体技术的飞速发展,实时多媒体服务应用越来越广泛,如视频会议、在线游戏、远程教育等互联网业务。这些服务往往需要底层网络提供组播功能。组播是指从一个源传输数据包到多个目的地。这些实时应用对服务质量(QoS)参数(如带宽、丢包率、延迟、抖动)有严格的要求。QoS 路由的关键问题是建立满足 QoS 参数的组播树。然而,多约束条件下构建组播树是 NP 完全问题<sup>[1]</sup>。因此,需要通过启发式智能优化求解该问题。

文献[2]提出一种基于蚁群算法 ACO 的分布式智能路由算法,通过感知节点能量、链路质量和链路生存性等网络状况来寻找路径,该算法可以找出一条近似最优路径。文献[3]提出一种基于蚁群算法的组播树生长算法(TGBACA),通过树的生长方式建立一颗组播树,并通过 ACO 算法进行优化。然而,这些算法都存在收敛速度慢等缺点。文献[4]提出一种基于树结

构演化的粒子群优化 PSO 算法,粒子由以组播树为分量的向量构成,表示问题的一个可行解,通过粒子的飞行来学习领域结构,从而找出全局最优解。文献[5]提出一种粒子群(PSO)树形优化算法(PSOTREE),直接在组播树上进行树形优化,并引入粒子群优化(PSO)算法求解此问题,一定程度上控制了树形变换的优化方向和进化速度。然而,这些算法的性能一定程度取决于粒子的数目。

本文提出一种基于群代理的融合蚁群(ACO)算法和粒子群优化(PSO)算法的多约束 QoS 组播路由优化算法(QoS-AP)。利用 ACO 算法中的蚂蚁对每个组播树模型和模型中的属性进行评估并放置信息素。PSO 算法产生多个移动代理,粒子代理根据模型和属性的信息素值,调整自己的运动方向,将多个模型中优秀的属性重新组合成一个新模型。经过多次迭代,最后形成一个满足 QoS 的最优组播树模型。仿真实验表明,本文算法

收稿日期:2014-04-21。四川省教育厅重点课题(13ZA0038)。李睿,工程师,主研领域:计算机网络,物联网,无线传感器网络。余勇,讲师。

能够找出更好的组播树模型,不仅能够满足 QoS 约束,而且还最大限度地减少了树的成本。

## 1 数学模型

将网络模型看成一个链接图  $G = (V, E)$ , 其中  $V$  是节点的集合,  $E$  是节点之间边的集合(网络链接)。设定  $n = |V|$  为网络节点的数目,  $l = |E|$  为网络链接的数目。如果存在从节点  $u \in V$  到节点  $v \in V$  的连接  $e = (u, v)$ , 则也存在一个从节点  $v$  到节点  $u$  的连接  $e' = (v, u)$ 。模型中, 有 4 个非负实数函数与连接  $e (e \in E)$  相关<sup>[6]</sup>: 成本函数  $C(e): E \rightarrow R^+$ , 延迟函数  $D(e): E \rightarrow R^+$ , 丢包率函数  $L(e): E \rightarrow R^+$ , 带宽函数  $B(e): E \rightarrow R^+$ 。

组播树<sup>[7]</sup>  $T(s, M)$  是  $G$  生成的源节点  $s \in V$  和目的节点集  $M \subseteq V - \{s\}$  的一个链接子图。 $m = |M|$  为组播目的节点的数目,  $M$  为目的节点集合,  $\{s\} \cup M$  为组播组。另外,  $T(s, M)$  中可能含有中继节点, 这些节点在组播树中, 但不在组播组中。 $P_T(s, d)$  表示在树  $T$  中, 源节点  $s$  到目的节点  $d \in M$  的唯一路径。

组播树  $T(s, M)$  的总成本为树上所有链接的成本之和, 其表达式为:

$$C(T(s, M)) = \sum_{e \in T(s, M)} C(e) \quad (1)$$

路径  $P_T(s, d)$  的总延迟为路径  $P_T(s, d)$  上所有链接延迟的总和, 其表达式为:

$$D(P_T(s, d)) = \sum_{e \in P_T(s, d)} D(e) \quad (2)$$

路径  $P_T(s, d)$  的总丢包率表达式为:

$$L(P_T(s, d)) = 1 - \prod_{e \in P_T(s, d)} (1 - L(e)) \quad (3)$$

路径  $P_T(s, d)$  的带宽为路径中所有链接的最小可用带宽, 其表达式为:

$$B(P_T(s, d)) = \min\{B(e), e \in P_T(s, d)\} \quad (4)$$

组播树  $T(s, M)$  的延迟抖动为从源节点到目的节点路径延迟的差, 其表达式为:

$$DJ(T(s, M)) = \sqrt{\sum_{d \in M} (D(P_T(s, d)) - delay\_avg)^2} \quad (5)$$

式中,  $delay\_avg$  表示在这条路径上源节点到目的节点的平均延迟。

设定  $\Delta$  为延迟约束,  $\zeta$  为丢包率约束,  $\beta$  为从源节点到目的节点  $d$  路径的带宽约束,  $\delta$  为延迟抖动约束。则多约束的最小成本组播问题可定义为最小化  $C(T(s, M))$  :

$$\begin{cases} D(P_T(s, d)) \leq \Delta \forall d \in M \\ L(P_T(s, d)) \leq \zeta \forall d \in M \\ B(P_T(s, d)) \geq \beta \forall d \in M \\ DJ(T(s, M)) \leq \delta \end{cases} \quad (6)$$

## 2 QoS-AP 算法(基于群代理的融合 ACO 和 PSO 算法优化 QoS 组播路由)

代理是一种能与外部环境和其它代理相互作用来实现特定目标的独立实体<sup>[8]</sup>。ACO 算法中, 系统将蚂蚁作为代理, 蚂蚁代理通过信息素互相交流来协调行动。PSO 算法中, 粒子代理也是通过与其它代理进行交互来协调它们的行为, 并将它们的信念(belief)与网络中最成功的信念相适应<sup>[10]</sup>。

### 2.1 ACO 和 PSO 算法

蚁群(ACO)算法是一种受蚂蚁自然行为中的合作和适应机制而启发的智能优化算法<sup>[11]</sup>。蚂蚁找出的路径的信息素值与路径所对应的解的质量成正比。每条边路径的信息素值还会决定蚂蚁选择下一步路径的概率。经过一定次数的迭代后, 最终蚂蚁会找出一条最优或近似最优解的路径。

粒子群优化(PSO)算法是一种受鸟类集群捕食而启发的基于种群搜索的自适应优化算法, 目的是找出一个适应度函数, 在一个给定的空间(搜索空间)中搜索全局最优解<sup>[12]</sup>。PSO 算法中粒子的运动方式会受到两个随机加权因子的影响<sup>[13]</sup>: 1) 个性认知, 指粒子自身具有的思考行为, 能够引导粒子向其之前发现的最佳位置运动; 2) 社会认知, 指粒子之间的相互交流和合作, 能够引导粒子向群体粒子中发现的最佳位置运动。

迭代中, 粒子的速度和位置更新公式为:

$$v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \cdot rand() (p_{id}^t - x_{id}^t) + c_2 \cdot Rand() (p_{gd}^t - x_{id}^t) \quad (7)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (8)$$

式中,  $w$  是惯性权重;  $v_{id}^t$  指在第  $t$  次迭代中, 第  $i$  个粒子在  $d$  维空间中的运动速度;  $x_{id}^t$  指在第  $t$  次迭代中, 第  $i$  个粒子在  $d$  维空间中的位置;  $c_1$  和  $c_2$  分别是个体认知和社会认知的学习因子;  $p_{id}$  指粒子之前发现的最好位置,  $p_{gd}$  指粒子群发现的全局最好位置;  $rand()$  和  $Rand()$  是  $[0, 1]$  区间内的随机数。其中,  $c_1$ 、 $c_2$  和  $w$  需要预先设定。

### 2.2 群集代理体系结构

自然界中存在很多群集行为, 如昆虫、鸟群、鱼群等, 它们通过某种方式自行组织, 在运动中形成整体动态平衡<sup>[14]</sup>。

首先, 随机生成  $n$  个组播树模型, 每个模型中, 生成  $m$  个密匙作为属性用来表示每个组播树模型中  $m$  个目的地<sup>[14]</sup>。因此, 这种模型的结构是由以下方程定义:

$$T_i = \{a_{i1}, a_{i2}, \dots, a_{im}\} \quad i = 1, 2, \dots, n \quad (9)$$

式中,  $T_i$  表示组播树模型  $i$ ,  $a_{ij}$  表示模型  $i$  的第  $j$  个属性,  $n$  是所有模型的数量。

在每个模型中, 粒子代理可以检测本地属性并放置信息素。这里有两种信息素: 一种是模型信息素, 另一种是模型内部的属性信息素。若一个模型含有较高的信息素, 则其成为最优模型的概率就高, 或这个模型中的一些属性具有较高的概率被最优模型所包含, 这也会吸引更多的粒子代理向这个模型移动。在算法迭代运行后, 粒子代理会聚集到最优模型上, 并利用原有模型中较好的属性来重组模型<sup>[13]</sup>。

### 2.3 提出的 QoS-AP 算法

最初, 随机生成  $n$  个候选模型, 并放置到大小为  $k \times l$  搜索空间中的网格中, 每个模型对应一个网格。然后, 产生  $n$  个粒子代理, 均匀分布在搜索空间中, 每个粒子代理也分别对应一个网格。开始迭代运行算法, 在每次迭代中, 每个粒子代理在组播树成本方面, 评估其所在网格中树模型的适应度值。利用 ACO 算法, 每个粒子代理根据评估值, 放置信息素到数模型中的每个属性中。

图 1 描述了在  $5 \times 6$  网格中按顺序分布 30 个组播树模型。30 个移动粒子代理  $p_i$  随机分布在搜索空间中, 每个粒子代理对应一个模型。在搜索网格中粒子代理有固定的拓扑结构, 其邻居的最大数量为 8。在交互过程中, 如果粒子发现邻居中最优模型的适应度高于其目前所在的模型, 则将这个邻居模型位置

作为发现的最优位置。然后,应用 PSO 优化算法更新模型信息和属性信息素,并重新检测树模型和其中属性的适应度值。这个过程被迭代一定次数后,则会返回具有最大适应值的模型,作为满足所有约束的解决方案。所提出的 QoS-AP 算法程序如算法 1 所示。

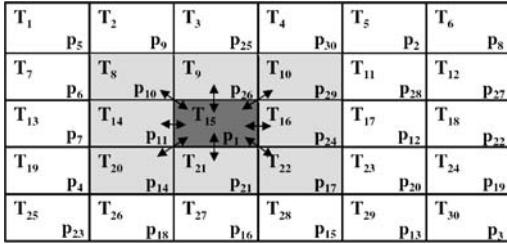


图1 粒子和模型代理分布

### 算法 1: QoS-AP 算法

输入:模型的数目  $n$ ,模型中属性数目  $m$ ,算法最大迭代次数  $Count_{max}$ 。

输出:最优组播树模型。

开始迭代;

/\* 利用 ACO 算法检测模型和属性的信息素 \*/

for  $i = 1$  to  $n$

增加粒子代理  $i$  所选择的模型的信息素;

for  $j = 1$  to  $m$

更新所有属性的信息素,增加代理  $i$  所选择的属性的信息素;

end for;

end for;

/\* 利用 PSO 算法调整粒子代理的运动方式 \*/

for  $i = 1$  to  $n$

更新模型和属性的信息素;

if (粒子当前位置模型的适应度值 < 最优邻居模型的适应度值)

则移动到这个邻居的模型中;

if(粒子最后位置模型中属性的适应度值 > 新位置)

则用最后位置高适应度属性替代目前位置低适应度属性;

if(新产生的模型适应度 > 之前模型的适应度)

则更新属性为高适应度;

end if;

知道迭代次数达到设置的最大迭代次数,迭代结束;

返回组播树模型。

组播树模型生成算法如算法 2 所示。

### 算法 2: 组播树模型( $T_i$ )生成程序

1. 开始;

2. 初始化组播树  $V_T$  从节点  $s$  开始;

3. 初始化节点  $s$  的延迟、丢包率和成本;

4. 设置当前节点  $cur\_node$  为节点  $s$ ;

5. 迭代开始

6. 判断当前节点不为空;

7. for all(当前节点  $cur\_node$  的所有邻居节点  $v$ )

8. if(当前节点到邻居节点  $v$  的带宽、延迟和丢包率均满足约束条件)

9. 则添加该邻居节点  $v$  到组播树;

10. end if;

11. end for all;

12. 从组播树  $V_T$  中随机选择一个节点作为当前节点  $j$ ;

13. 计算组播树总成本  $cost$ ;

14. 计算节点  $j$  成本  $costsofar(j)$ ;

15. 计算节点  $j$  信息素  $pheromone(j)$ ;

16. 计算节点  $j$  延迟  $delaysfar(j)$ ;

17. 计算节点  $j$  丢包率  $losssofar(j)$ ;

18. if(节点  $j$  不是目的节点)

19. 则设置节点  $j$  为当前节点  $cur\_node$ ;

20. else

21. 从  $V_T$  中重新选择一个节点作为当前节点  $cur\_node$ ;

22. for all(不是目的节点且不输入组播树的节点  $u$ )

23. 添加节点  $u$  到组播树,并更新组播树成员信息素  $pheromone(u)$ ;

24. end for all;

25. end if;

26. 直到  $V_T$  包含组播组中所有节点,迭代结束;

27. if( $T_i$  满足延迟抖动  $DJ(T_i(s, M)) \leq \delta$ )

28. 则从树成本上计算信息素( $T_i$ );

29. 返回  $T_i$ ;

30. end;

在组播树模型生成程序中,首先初始化设置组播树  $V_T$  从源节点  $s$  开始,设置  $s$  作为当前节点。设置源节点  $s$  的延迟  $delaysfar(s)$  和丢包率  $losssofar(s)$  分别为 0 和 1。然后,从当前节点的邻居节点中选择一个符合 QoS 约束的节点  $j$ 。添加节点  $j$  到组播树  $V_T$  中,通过算法中步骤 16 和步骤 17 计算节点  $j$  的  $delaysfar(j)$  和  $losssofar(j)$ 。如果节点  $j$  不是目的节点,则将  $j$  作为当前节点,否则,从  $V_T$  中重新随机选择当前节点。通过步骤 23,对组播树中成员的信息素进行更新。重复以上过程,直到所有组成员被添加到组播树中。

本文算法中,当粒子移动到新的位置时,会带来上一个位置的属性和与其相关的适应度值。然后,将较好质量的属性代替新位置中较差质量的属性,则会产生一个比原来具有更高质量的新组合模型。在下次迭代中,新组合的模型将被代理人执行,并放置信息素。经过多次迭代后,则会产生最优模型。

## 3 仿真实验

本文利用 Visual C++ 编译器实现提出的算法,设定节点随机分配在一个  $4000\text{ m} \times 2400\text{ m}$  大小的矩形范围内。利用欧氏距离来确定每对节点之间的距离,利用 Waxman 算法<sup>[15]</sup> 构建网络拓扑结构。节点  $u$  和节点  $v$  之间的边链接存在的概率取与它们之间的距离。边存在的概率为  $P(u, v) = \beta \exp(-l(u, v)/\alpha L)$ , 其中  $0 < \alpha, \beta \leq 1$ ,  $l(u, v)$  是节点  $u$  到节点  $v$  之间的欧氏距离,  $L$  是网络中任意两个节点之间的最大距离。用  $\alpha$  的值来控制随机生成的网络拓扑结构中短链接的数量,  $\alpha$  的值越小,短链接的数量越多。用  $\beta$  来控制网络拓扑中链接的数量,  $\beta$  值越低,链接数量越多。本文仿真中,设定  $\alpha$  和  $\beta$  的值分别为 0.8 和 0.7。设置链接的延迟为 1 ~ 30 范围内的随机数,丢包率为 0.0001 ~ 0.01 范围内的随机数,带宽为 1 ~ 100 范围内的随机数。设置最大迭代次数  $Count_{max}$  为 30。

从网络拓扑中的节点集中随机选择源节点,并均匀选择目的节点。设置延迟约束  $\Delta$ 、延迟抖动约束  $\delta$  和丢包率约束  $\zeta$  分别为 120 ms、60 ms 和 0.05,随机产生组播应用中所需的带宽。在同样的仿真环境中,随机产生 30 个组播树模型,用于分析和比较 PSOTREE、TGBACA 算法和本文算法的性能。这 30 个组播树模型根据生成顺序分布到大小为  $5 \times 6$  的矩形网格中。在每种模型中执行程序 100 次,并得出实验平均值。

实验中, PSOTREE、TGBACA 和本文提出算法产生的组播树均满足延迟、延迟抖动、丢包率和带宽约束。图 2 描述了在有 100 个节点的网络中, 三种方案形成的组播树的成本与组播组大小的关系; 图 3 描述了在不同网络节点数量(10% 作为目的节点)下, 三种方案形成的组播树的成本; 图 4 描述了在不同网络节点数量(25% 作为目的节点)下, 组播树的成本。从图中可以看出, 本文算法生成的组播树成本小于 PSOTREE 和 TGBACA 算法。PSOTREE 算法通过消除定向循环来构建组播树, 然而, PSOTREE 算法不能够产生较好的树, 因为其在循环中删除的连接可能优于不在定向循环中的连接。TGBACA 算法根据信息素更新策略来构建最优组播树, 在每一代都更新链接上信息素产生最优树, 然而, 这种策略会加速收敛过程, 使结果容易陷入局部最优解。

本文算法将两个组播树模型相融合, 把一个模型中较好的属性带到另一个模型中, 在每次迭代后产生一个新的组播树模型。经过多次迭代后, 算法会收敛到一个最优组播树模型。所以本文算法比 PSOTREE 和 TGBACA 算法具有更好的性能。

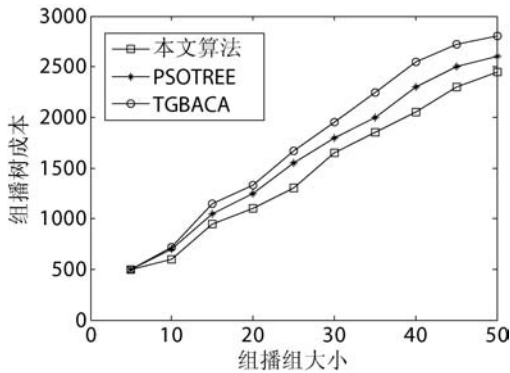


图 2 组播树成本和组播组大小的关系

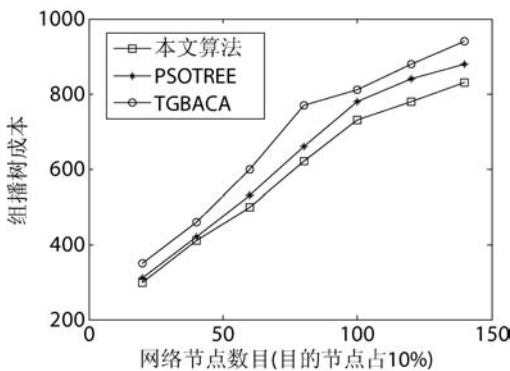


图 3 组播树成本和网络大小(目的节点占 10%)的关系

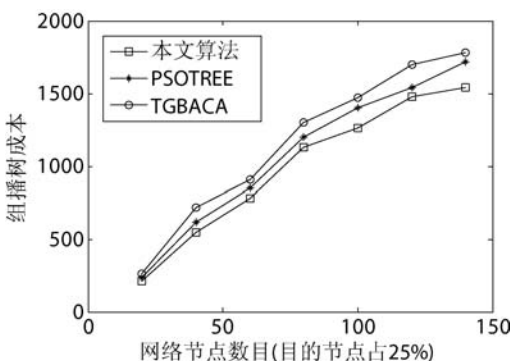


图 4 组播树成本和网络大小(目的节点占 25%)的关系

的节点)的网络中, 三种算法产生组播树的延迟和延迟抖动。其中, 本文算法与 PSOTREE、TGBACA 算法产生的组播树均满足延迟和延迟抖动约束。从图中可以看出, 在一些情况下, 本文算法的延迟和延迟抖动略高于 PSOTREE 和 TGBACA 算法, 然而相差不大。

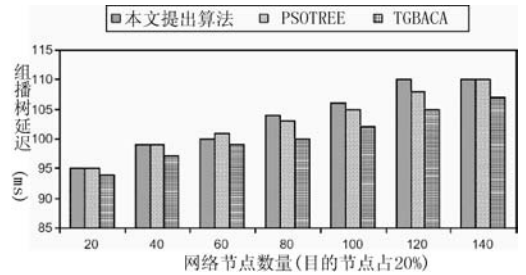


图 5 组播树延迟和网络大小(目的节点占 20%)的关系

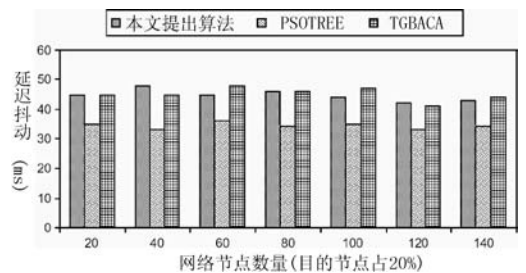


图 6 组播树延迟抖动和网络大小(目的节点占 20%)的关系

## 4 结 语

本文提出了一种基于群代理的融合 ACO 与 PSO 算法的 QoS 组播路由优化算法(QoS-AP)。利用 ACO 算法中的蚂蚁对每个模型和模型中的属性进行评估并放置信息素。PSO 算法产生多个移动代理, 粒子代理根据模型和属性的信息素值, 调整自己的运动方向, 将多个模型中优秀的属性重新组合成一个新模型。经过多次迭代, 最后形成一个满足 QoS 的最优组播树模型。通过仿真实验比较本文算法与现有的 PSOTREE、TGBACA 算法的性能。结果表明, 本文算法能够找出更好的组播树模型, 不仅能够满足 QoS 约束, 而且还最大限度的减少了树的成本, 并且在延迟和延迟抖动方面与现有算法差距不大。

## 参 考 文 献

- [1] 陈西宏, 刘少伟, 胡茂凯, 等. 基于蚁群遗传混合算法的 QoS 组播路由[J]. 计算机工程, 2011, 37(4): 99-101.
- [2] 许洪, 王华, 伊善文. 分布式蚁群 QoS 组播路由算法[J]. 计算机工程, 2010, 36(6): 111-113.
- [3] Di Caro G, Dorigo M. AntNet: Distributed Stigmergetic Control for Communications Networks[J]. Journal of Artificial Intelligence Research, 2011, 9: 317-365.
- [4] 马炫, 刘庆. 多组播路由问题的粒子群优化算法[J]. 计算机研究与发展, 2013, 50(2): 260-268.
- [5] 李帅, 王华, 王新军, 等. 树形变换的 PSO 组播路由算法[J]. 小型微型计算机系统, 2009(8): 1500-1506.
- [6] Zhu F, Wang H. A modified ACO algorithm for multicast state scalability problem based on multicast tree similarity[C]//Advanced Communication Technology (ICACT), 2011 13th International Conference on. IEEE, 2011: 972-976.



(a) 普通镜头 (b) 广角镜头 (c) 校正后图像

图 11 实景一



(a) 普通镜头 (b) 广角镜头 (c) 校正后图像

图 12 实景二



(a) 普通镜头 (b) 广角镜头 (c) 校正后图像

图 13 实景三

## 4 结 语

本文提出了一种改进传统半球面畸变的校正算法,采用可用于不同镜头的畸变校正模型,利用这种模型可以方便地得到各点的畸变率。该方法不需要为了确定校正参数的值而进行大量的迭代运算,仅仅使用一个包含对数运算和平方根运算的公式来计算畸变图像上每个像素点所对应坐标,能够减小计算复杂度。并且利用了畸变的对称性,当图像尺寸较大时,能够有效缩短计算时间,具有较好的通用性。另外,在模型中,我们给出了用于控制畸变校正程度的参数,以及参数的确定方法及相应的实验结果。该方法在进行图像校正后的畸变率误差大约还存在 1% 左右,比传统算法校正后余下的畸变率误差 5% 小,结果表明该算法提高了校正的精确度。虽然算法中的参数  $a_1$  可以方便的调节,但是为了得到更精确的校正效果时,仍然需要使用较为复杂的确定方法,怎样用更好的方法确定参数的精确取值是笔者需要进一步研究的内容。

## 参 考 文 献

- [ 1 ] Chou F, Cui Y. Line-based camera calibration with lens distortion correction from a single image[J]. *Optics and Lasers in Engineering*, 2013, 51(12): 1332-1343.
- [ 2 ] Li Zhaoxin. Distortion correction in wide-angle images for picture-based food portion size estimation [C]//*Bioengineering Conference*, 2012: 424-425.
- [ 3 ] 史泽林. 基于 BP 神经网络的大视场成像畸变校正方法[J]. *光学精密工程*, 2005(3): 348-353.
- [ 4 ] Henryk Blasinski, Wei Hai. Real-time, color image barrel distortion removal[C]//*IEEE International Symposium on Circuits and Systems*, 2012:1911-1914.

- [ 5 ] Henryk Blasinski, Wei Hai. FPGA architecture for real time barrel distortion correction of colour images[C]//*IEEE International Conference on Multimedia and Expo*. 2011: 1-6.
- [ 6 ] 刘亚强, 陈文艺. 桶形畸变图像的一种校正方法[J]. *西安邮电学院学报*, 2012, 17(2): 27-36.
- [ 7 ] 史延东, 刘海清, 宁飞. 大视场景物非线性畸变校正的仿真[J]. *计算机仿真*, 2011, 28(7): 268-271.
- [ 8 ] 刘军, 胡振伟, 张达, 等. 基于控制点鲁棒提取的视觉成像畸变校正方法[J]. *东北大学学报: 自然科学版*, 2010(3): 330-333.
- [ 9 ] 韩广良, 宋建中. 一种基于畸变等效曲面的图像畸变校正[J]. *光学技术*, 2005(1): 122-124.
- [ 10 ] 崔洪州, 孔渊. 基于畸变率的图像几何校正[J]. *应用光学*, 2006(3): 183-185.
- [ 11 ] Fr'ed'eric Devernay, Olivier Faugeras. The distortion correction of large view wide-angle lens for image mosaic based on OpenCV [C]//*International Conference on Mechatronic Science, Electric Engineering and Computer*. 2011: 1074-1077.
- [ 12 ] Won Jun Kim, Changick Kim. An efficient correction method of wide-angle lens distortion for surveillance systems[C]//*IEEE International Symposium on Circuits and Systems*, 2009: 3206-3209.
- [ 13 ] Smith P W, Johnson K B, Abidi M A. Efficient techniques for wide-angle stereo vision using surface projection models[J]. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, 6(1): 113-118.
- [ 14 ] 杨广全. 鱼眼图像校正算法[J]. *影像技术*, 2010(3): 16-18.
- [ 15 ] 田原姬, 黄合成. 基于 CCD 摄像机成像的径向畸变研究[J]. *世界科技研究与发展*, 2008(2): 168-170.

## (上接第 130 页)

- [ 7 ] Sahoo S P, Ahmed S, Patel M K, et al. A Tree Based Chemical Reaction Optimization Algorithm for QoS Multicast Routing[M]//*Swarm, Evolutionary, and Memetic Computing*. Springer International Publishing, 2013:68-77.
- [ 8 ] 王建生, 曹叶文. 基于动态组播代理的移动组播协议[J]. *计算机工程*, 2010, 36(1): 87-90.
- [ 9 ] 梁淑萍, 毛力, 马亦先. 基于蚁群算法的 Ad Hoc 网络 QoS 组播路由研究[J]. *微电子学与计算机*, 2011, 28(7): 164-167.
- [ 10 ] Patel M K, Kabat M R, Tripathy C R. A swarm intelligence based algorithm for qos multicast routing problem[M]//*Swarm, Evolutionary, and Memetic Computing*. Springer Berlin Heidelberg, 2011:38-45.
- [ 11 ] 蒲汛, 于显平, 卢显良. 改进 Pareto 蚁群算法的多 QoS 组播路由策略[J]. *小型微型计算机系统*, 2012, 33(12): 2575-2579.
- [ 12 ] Abdel-Kader R F. An Improved Discrete PSO with GA Operators for Efficient QoS-Multicast Routing[J]. *International Journal of Hybrid Information Technology*, 2011, 4(2): 23-37.
- [ 13 ] Abdel-Kader R F. Hybrid discrete PSO with GA operators for efficient QoS-multicast routing [J]. *Ain Shams Engineering Journal*, 2011, 2(1): 21-31.
- [ 14 ] Wang H, Meng X, Li S, et al. A tree-based particle swarm optimization for multicast routing [J]. *Computer Networks*, 2010, 54(15): 2775-2786.
- [ 15 ] Sahoo S P, Ahmed S, Patel M K, et al. A Tree Based Chemical Reaction Optimization Algorithm for QoS Multicast Routing[M]//*Swarm, Evolutionary, and Memetic Computing*. Springer International Publishing, 2013:68-77.