

# 基于 CUDA 与粒子滤波的多特征融合视频目标跟踪算法<sup>①</sup>

刘 伟, 孟朝晖, 薛东伟

(河海大学 计算机与信息学院, 南京 211100)

**摘 要:** 针对复杂环境中非线性运动目标跟踪, 单一特征无法满足对目标的准确描述, 造成不能准确跟踪的问题, 提出了一种基于粒子滤波与多种特征自适应融合的跟踪方法. 该方法先对目标区域提取轮廓方向分布与颜色分布, 根据自适应规则融合后, 然后与粒子滤波理论相结合, 实现对各种复杂环境中视频运动目标的有效跟踪. 同时, 通过使用 CUDA(Compute Unified Device Architecture)加速, 实现了目标跟踪的实时性. 实验结果表明, 该方法可对非线性、非高斯的运动目标进行有效的跟踪, 对目标的遮挡与暂时消失, 背景焦距的拉伸与背景颜色的变换, 有很强的鲁棒性和实时性.

**关键词:** 视频目标跟踪; 粒子滤波; 轮廓方向分布; 颜色分布; CUDA

## Multi-Feature Fusion Video Object Tracking Algorithm Based on CUDA and Particle Filter

LIU Wei, MENG Zhao-Hui, XUE Dong-Wei

(Computer Science and Technology Institute of Hehai University, Nanjing 211100, China)

**Abstract:** To solve the problem that single feature can't describe the nonlinear moving targets accurately under complex environment, this paper proposed a adaptive method which is based on multiple features and particle filter. It first extracted contour direction distribution and color distribution from the target area. And then with help of particle filter theory, it integrated these features to track nonlinear moving targets from the video. In the end, CUDA can provide powerful parallel computing capacity so that this proposed method can be accelerated and meet the need of real-time monitoring. The experiment results of this paper indicates that the proposed method has strong robustness and high accurateness for non-linear and non-Gaussian moving targets. Moreover, it can cope with the situations of targets occlusion or temporarily disappearing, focal length transformation and background changing.

**Key words:** video target tracking; particle filter contour; direction distribution; color distribution; CUDA

## 1 引言

视频目标跟踪是图像和视频处理领域的一个重要研究方向, 广泛应用于智能导航、交通监视、人机交互、视频会议和军事制导等多个领域, 是当前计算机视觉领域的研究热点和难点之一. 视频目标跟踪的目的是通过对传感器拍摄到得视频序列进行分析, 计算出目标在每一帧图像上的二维位置, 并将图像序列中不同帧中同一目标关联起来得到一个运动目标的完整的运动轨迹. 粒子滤波器依据是蒙特卡洛理论, 它的

优点是对状态模型的依赖性不强, 能够有效地处理非线性和非高斯条件下动态系统的状态估计问题, 在实际应用中十分广泛.

同时, 建模特征的选取也很大程度决定跟踪性能, 这些特征可以是颜色分布、形状、轮廓以及点特征等等. 各种特征都有自身的优点与局限性, 比如颜色分布特征对运动目标的形变和旋转具有一定的鲁棒性, 但是在背景和目标具有相似颜色特征时, 颜色特征分布则不能把目标和场景很好地区分开来; 虽然

<sup>①</sup> 收稿时间:2013-04-17;收到修改稿时间:2013-05-20

形状特征能够克服颜色变化的影响,但是对于复杂环境,跟踪性能不高.针对视频跟踪环境复杂,跟踪目标交互、目标暂时消失、目标背景变化等情况,单一的特征很难完成持续跟踪,本文提出了一种多特征融合建模的方法,通过自适应融合算法,综合几种特征的优势,极大地提高了系统的鲁棒性.同时为了保证跟踪的实时性,本算法采用 CUDA 加速,实验证明,本算法在各种复杂环境下的跟踪比单一的跟踪特征鲁棒性更高,性能更加优越.

## 2 目标特征选取

由于单一特征无法准确描述运动目标,有效完成目标跟踪的,而采用多特征,可以优势互补,能够更加准确的描述运动目标,在复杂环境中,跟踪的鲁棒性更高.实验中,采用颜色分布与轮廓方向分布对目标建模,很好的解决了目标遮挡、目标暂时消失、背景环境复杂的问题.下面是颜色分布与轮廓方向分布的提取方法.

### 2.1 轮廓方向分布

轮廓方向分布是利用核概率密度估计对目标轮廓上的每个点相对质心的方向,离散化后的分布,具体计算方法如下:

(1) 在当前图像中选择目标区域,并得到单独的目标区域图像,用边缘算子对目标区域图像进行边缘化处理,提取目标的边缘轮廓点序列.设边缘轮廓点集合为:

$$E = \left\{ \begin{matrix} x_1, x_2, \dots, x_s \\ y_1, y_2, \dots, y_s \end{matrix} \right\} \quad (1)$$

式中  $s$  表示轮廓上点的数量.

(2) 求出轮廓的质心  $G = (\bar{x}, \bar{y})$ :

$$\bar{x} = \frac{1}{s} \sum_{i=1}^s x_i, \bar{y} = \frac{1}{s} \sum_{i=1}^s y_i \quad (2)$$

(3) 对轮廓上的每一点求方向,得到轮廓方向集合

$$\theta = \{ \theta_i = \arctan(\frac{x_i - \bar{x}}{y_i - \bar{y}}), i = 1, 2, \dots, s \} \quad (3)$$

(4) 把轮廓方向离散化为  $m$  个方向,即

$\{\theta_i\} \xrightarrow{f} \{1, 2, \dots, m\}$ , 即用 1 表示  $\theta_i$  在  $0 \sim \frac{\pi}{m}$  范围内,

$m$  表示  $\theta_i$  在  $\frac{m-1}{m}\pi \sim \pi$  内的范围内.

(5) 利用核密度估计,将目标轮廓方向分布离散化为  $m$  维,定义量化函数为  $M(p_i): R^2 \rightarrow \{1, 2, \dots, m\}$ , 表示把轮廓点  $p_i$  根据其轮廓方向分配相应的数字. 设给定的当前目标状态为  $X_t$ , 目标的边缘分布  $FF = \{F_u\}_{u=1,2,\dots,m}$ , 其定义为:

$$F_u = \frac{1}{s} \sum_{i=1}^s \delta(M(p_i) - u) \quad (4)$$

其中  $s$  为边缘点的个数,  $\delta(\cdot)$  为 Delta 函数.

### 2.2 颜色方向分布

采用带空间位置信息的颜色直方图,在实验中,将颜色信息在 HSV 颜色空间划分成为  $m_1 = 10 \times 10 + 10$  个等级.假设目标的状态为  $X_t$ , 其中目标区域的中心为  $g = (x, y)$ , 半径为  $h$ , 目标区域内像素的位置为  $x_i = (x_i, y_i)$ , ( $i = 1, 2, \dots, k$ ),  $k$  为目标区域内像素的总数,则目标区域内的颜色分布可以表示为

$$G_u = \frac{\sum_{i=1}^k k \left( \left\| \frac{x - x_i}{h} \right\| \right) \delta[b(x_i) - u]}{\sum_{i=1}^k k \left( \left\| \frac{x - x_i}{h} \right\| \right)}, u = 1, 2, \dots, m_1 \quad (5)$$

其中,  $\delta(\cdot)$  为 Delta 函数, 函数  $b(x_i)$  为位于  $x_i$  处的像素在直方图上颜色等级索引的映像,  $u$  为直方图中颜色等级索引,  $k(\cdot)$  为权值函数.

## 3 多特征融合粒子滤波算法

粒子滤波器包括预测与更新是其基本的两个步骤,实现这两个步骤的前提是目标的状态转移概率模型与观测概率模型.

### 3.1 粒子滤波算法

粒子滤波是指通过寻找一组在空间中传播的随机样本,对概率密度函数  $p(X_k | Z_k)$  进行近似,以样本均值代替积分运算,从而获取状态的最小方差估计的过程.定义  $X_k^i$  表示  $k$  时刻第  $i$  个粒子,  $\omega_k^i$  表示该粒子在第  $k$  时刻的权值,则利用一组带权值的随机变量  $\{X_k^i, \omega_k^i\}_{i=1}^N$  统计逼近后验概率函数的表达式为

$$p(X_k | Z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(X_k - X_k^i) \quad (6)$$

其中  $\sum_{i=1}^N w_k^i = 1$ ;  $\delta(\cdot)$  表示 Delta 函数. 权值更新公式

为:

$$w_k^i \propto w_{k-1}^i \frac{p(Z_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, Z_k)} \quad (7)$$

其中  $q(X_k^i | X_{k-1}^i, Z_k)$  为重要密度, 为了方便计算可以令

$$q(X_k^i | X_{k-1}^i, Z_k) = p(X_k^i | X_{k-1}^i) \quad (8)$$

将式(8)代入式(7)得

$$w_k^i \propto w_{k-1}^i p(Z_k | X_k^i) \quad (9)$$

将权值归一化处理, 获取加权粒子集合  $\{X_k^i, \omega_k^i\}_{i=1}^N$ , 从而求得最小均方差估计为

$$\bar{X}_k \approx \sum_{i=1}^N w_k^i X_k^i \quad (10)$$

### 3.2 状态转移概率模型

对于视频目标来说, 在相邻帧之间, 目标运动的机动性不是很大, 可以采用二阶常速模型来描述其运动规律. 本文中, 采用一个矩形来表示一个目标模板, 用  $p = [x, y, h_x, h_y]$  来表示, 其中  $x$  和  $y$  分别为矩形的中心,  $h_x$  与  $h_y$  分别表示为矩形的半长轴与半宽轴. 这样目标状态可以表示为:

$$X = [x, x', y, y', h_x, h_y]^T \quad (11)$$

其中  $x'$  与  $y'$  分别表示为目标中心在图像  $x$  和  $y$  方向的速度. 在建立状态模型是, 对  $(h_x, h_y)$  采用随机扰动模型, 这样, 目标的状态随时间的变换过程可以描述为:

$$X_t = \begin{bmatrix} 1 & \Delta R & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta R & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} X_{t-1} + \begin{bmatrix} \frac{\Delta R^2}{2} & 0 & 0 & 0 \\ \Delta R & 0 & 0 & 0 \\ 0 & \frac{\Delta R^2}{2} & 0 & 0 \\ 0 & \Delta R & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} U_t \quad (12)$$

其中  $\Delta R$  为采样周期,  $U_t$  是一个多变量的高斯白噪声.

### 3.3 观测概率模型

对视频目标跟踪, 需要选择一个合适的特征来描述运动目标, 然后在每帧图像中确定目标的位置. 本文选择目标的轮廓方向分和颜色分布来作为描述目标的特征; 通过比较目标样本区域和参考目标的轮廓方向分布和颜色分布的 Bhattacharyya 距离, 然后利用自适应融合算法, 将得到的值, 进行加权相加, 建立系统观测概率模型.

#### 3.3.1 目标的轮廓方向分布信息与颜色信息

在视频的初始帧, 手动选择要跟踪的目标的参考

模型  $X_0$ , 利用式(9)和式(10)计算出参考模型的轮廓方向分布  $R = (R_i)_{i=1,2,\dots,m}$  和颜色分布  $S = (S_i)_{i=1,2,\dots,m_1}$ . 在第  $k$  帧, 计算目标状态  $X_k$  的第  $i$  个采样  $X_k^i$  所对应的图像区域的轮廓方向分布  $F = (F_i)_{i=1,2,\dots,m}$  与颜色分布  $G = (G_i)_{i=1,2,\dots,m_1}$ . 然后可以通过 Bhattacharyya 距离计算对应分布之间的相似度.

一、样本  $X_k$  与参考模型  $X_0$  的轮廓方向分布相似程度可以表示为

$$d_s[R, F] = \sqrt{1 - \rho[R, F]} \quad (13)$$

其中  $\rho[R, F]$  即为 Bhattacharyya 系数

$$\rho[R, F] = \sum_{u=1}^m \sqrt{R_u F_u} \quad (14)$$

根据得到的 Bhattacharyya 距离, 可以设计轮廓方向分布的观测似然函数:

$$p_s(Z_t | X_t) = \frac{1}{\sqrt{2\pi\sigma_s}} \exp\left(-\frac{d_s^2[R, F]}{2\sigma_s^2}\right) \quad (15)$$

其中,  $\sigma_s^2$  为高斯方差, 式(15)越大, 说明候选目标与目标模板的轮廓越相似, 候选目标是真实目标的可能性越大.

二、样本  $X_k$  与参考模型  $X_0$  的颜色分布相似程度可以表示为

$$d_c[S, G] = \sqrt{1 - \rho[S, G]} \quad (16)$$

其中  $\rho[S, G]$  即为 Bhattacharyya 系数

$$\rho[S, G] = \sum_{u=1}^{m_1} \sqrt{S_u G_u} \quad (17)$$

根据得到的 Bhattacharyya 距离, 可以设计颜色分布的观测似然函数:

$$p_c(Z_t | X_t) = \frac{1}{\sqrt{2\pi\sigma_c}} \exp\left(-\frac{d_c^2[S, G]}{2\sigma_c^2}\right) \quad (18)$$

其中,  $\sigma_c^2$  为高斯方差, 式(18)越大, 说明候选目标与目标模板的颜色越相似, 候选目标是真实目标的可能性越大.

#### 3.3.2 多特征信息融合

利用多种特征信息跟踪目标, 可以实现个特征信息的优势互补, 提高跟踪的鲁棒性. 本文中利用颜色分布与轮廓方向分布描述目标的观测信息. 对于给定目标状态为  $X_t$  时, 目标的观测似然函数表达式为:

$$p(Z_t | X_t) = \alpha p_s(Z_t | X_t) + \beta p_c(Z_t | X_t) \quad (19)$$

其中  $\alpha + \beta = 1$ ,  $p_s(Z_t | X_t)$  和  $p_c(Z_t | X_t)$  分别为轮廓

方向分布与颜色分布的观测似然函数, 分布由式(15)与式(18)得到.  $\alpha$  与  $\beta$  分别为两种特征融合时的权值, 当轮廓方向特征可靠时,  $\alpha$  越大, 反之亦然. 所以, 可以根据观测似然函数的值的大小, 设定一定规则, 让  $\alpha$  与  $\beta$  自适应调节, 达到最佳的跟踪效果.

用  $e_s$  与  $e_c$  分别表示轮廓方向与颜色特征信息的可信度, 分布由式(15)与式(18)得到, 然后将  $e_s$  与  $e_c$  模糊化为五个等级 {S1,S2,M,L1,L2},  $\alpha$  模糊化为九个等级 {S1,S2,S3,M1,M2,M3,L1,L2,L3} 将数字代表等级, 等级越大, 可信度越高. 根据以上信息, 建立如下模糊规则见图 1.

$\alpha$		$e_s$ 可信度值				
		S1	S2	M	L1	L2
$e_c$ 可信度值	S1	M2	M3	L1	L2	L3
	S2	M1	M2	M3	L1	L2
	M	S3	M1	M2	M3	L1
	L1	S2	S3	M1	M2	M3
	L2	S1	S2	S3	M1	M2

图 1 模糊规则  $\alpha$  的取值

#### 4 系统的实现与实验结果

本文使用的粒子滤波算法, 特别是引入多特征匹配后, 要进行的直方图匹配计算量成倍增加. 由于 GPU 通过增加并行处理单元和存储结器控制单元的方式提高处理能力和存储器带宽, 使得 GPU 在处理能力和存储器带宽上相对于 CPU 有明显优势, 从而为计算能力问题提供了新的解决方案.

##### 4.1 基于 CUDA 的程序实现

本实验中, 假设每一帧每一个目标会产生 100 个粒子, 每个粒子都需要与模板进行匹配计算, 对于两个特征就需要 200 次的匹配计算, 这个计算量是相当大的, 必须使用并行计算.

本文让每个粒子的每一个像素点由一个线程访问, 每一个粒子的颜色分布与轮廓方向分布由一个线程块负责计算, 由于目前的 CUDA 最多支持的线程数为 block 数\*thread 数=65536\*512 个(GeForce 8 系列), 这样如果目标大小为 655\*512 个像素,CUDA 完全可以一次全部载入全部的 100 粒子并对其进行颜色分布与轮廓方向分布就算. 具体流程如下:

Host 端:

- (1) 在第一帧手动获取的参考模板.
- (2) 利用 cudaMalloc 函数在显存开辟内存  $p$ , 保存

目标模板的数据.

- (3) 调用核函数:

Cal\_hist<<<dimGrid,dimBlock>>>(device\_address, p); 其中, dimGrid 定义了线程块 Block 数量为该模板宽的像素个数, dimBlock 规定了每个 block 中使用的线程数量, 此处取该模板长的像素个数.

以下操作对于视频每一帧循环:

- (4) 利用 cudaMalloc 函数在显存开辟直方图结构体数组 hist[i][j], 其中 i 为特征数, j 为每个目标的粒子数, 用来保存每一个粒子的颜色分布与轮廓方向分布.

- (5) 利用 cudaMalloc 函数在显存开辟目标粒子数组 particle[j], j 为每个目标的粒子数, 用来保存每一个目标的粒子.

- (6) 利用 cudaMalloc 函数在显存开辟数组 sort[j], j 为每个目标的粒子数, 用来保存每一个粒子与模板的相似度.

- (7) 调用 cudamMemcpy 函数将每一帧每一个目标的粒子复制到 particle[j]中.

- (8) 调用依次核函数:

Cal\_histparticle<<<dimGrid1,dimBlock1>>>(device\_address,particle);其中, dimGrid 定义了线程块 Block 数量为该粒子的宽的像素个数, dimBlock 规定了每个 block 中使用的线程数量, 此处取该粒子长的像素个数;

- (9) 核函数计算完成后, 将 sort[j]与 max 的值利用 cudaMemcpy 函数拷贝回主存的数组 sort1[j]与 max1 中.

- (10) 释放 GPU 上粒子所占的空间.

- (11) 然后利用式(19)计算总的观察似然函数.

- (12) 重采样.

- (13) 状态估计输出.

- (14) 返回第四步循环.

Device 端:

- (1) 定义共享内存变量 hist\_color 与 hist\_count, max;

(2) 计算每个线程访问的目标模板像素的索引值 (x,y), 分别计算模板的颜色直方图与轮廓方向直方图, 分别保存在 hist\_color 与 hist\_count 中.

(3) 计算每个线程访问粒子图像像素的索引值 (x,y), 分别计算粒子的颜色直方图与轮廓方向直方图, 保存在 hist[i][j]中;

(4) 计算每个粒子特征直方图与目标模板特征直方图的相似度, 保存在 sort[j], 并对 sort[j]进行排序, 保存值最大的粒子到 max 中.

## 4.2 实验结果

为了验证本文中所提方法的性能,选取了几组背景复杂,背景颜色与目标颜色相近的视频(其中有几组视频加入了镜头抖动,焦距拉伸变化,增加跟踪难度)进行实验.

本文实验包涵三个部分,全方位对本文提出的方法进行实验,并与其他方法进行优劣性比较.实验都是在第一帧手动选取目标,选取区域的大小跟视频中目标的大小相关,实验图像都是通过截屏软件截取.实验 1 是目标颜色与背景颜色相似的单目标跟踪,主要难度是目标颜色与背景颜色相似,并存在暂时遮挡现象.实验 2

是带有镜头抖动与焦距变换的双目标跟踪.实验 3 是在实验 1 的环境下,采用均值漂移算法进行跟踪.前面两个实验分为两部分,一个是基于单特征算法,一个是基于多特征融合算法.实验 3 是基于均值漂移算法.实验仿真环境为:主机配置为 CPU Dual-Core 2.66GHz、内存 2 GB, NVIDIA GeForce 210 显卡,采用 opencv 运行环境,实验的基本信息与处理时间见表 1.

为了直观的衡量算法的优劣,通过计算真实目标位置的质心与跟踪目标的质心的欧式距离,作为跟踪的误差.实验 1 与实验 3 的跟踪误差见图 7.实验 2 的跟踪误差见图 8.

表 1 实验基本输入信息、参数信息及处理效率表

		视频信息	参数信息	跟踪处理时长	平均每秒处理帧数
实验一	多特征	大小: 40.3M, 640*480	粒子数: 100; 目标初始大小 51*141	55.34s	11.36
	单特征	帧速率: 30 秒/帧, 时长 21 秒	粒子数: 100; 目标初始大小 52*141	50.26s	12.53
实验二	多特征	大小: 33.9M, 640*480	粒子数: 100; 目标大小 29*75 与 28*70	57.50s	9.3
	单特征	帧速率: 30 秒/帧, 时长 18 秒	粒子数: 100; 目标大小 30*71 与 29*74	53.43s	10.10



图 2 实验 1 基于多特征融合的单目标跟踪算法  
(从左到右分别为第 1,15,30,40 帧跟踪结果)



图 3 实验 1 基于单特征(边缘方向)的单目标跟踪算法  
(从左到右分别为第 1,15,28,37 帧跟踪结果)



图 4 实验 2 基于多特征融合的双目标跟踪  
(从左到右分别为第 1,15,35,45 帧跟踪结果)



图 5 实验 2 基于单特征(基于颜色)的双目标跟踪  
(从左到右分别为第 1,15,35,43 帧跟踪结果)



图 6 实验 3 基于均值漂移单目标跟踪算法  
(从左到右分别为第 1,15,31,38 帧跟踪结果)

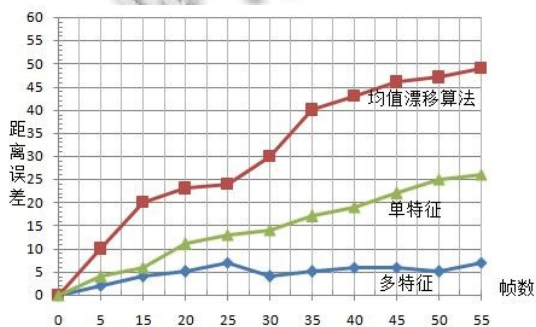


图 7 单目标跟踪误差表

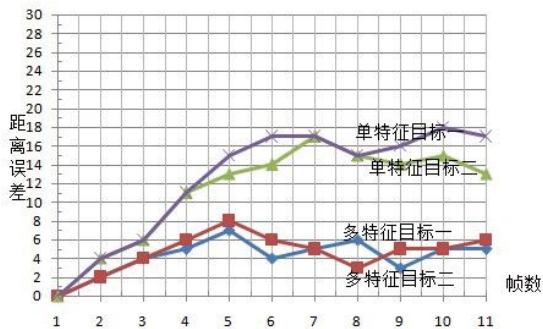


图 8 多目标跟踪误差表

### 5 结束语

本文主要针对单一特征无法准确描述目标这一实际,提出了多目标融合的粒子滤波算法,并通过实验验证,能够对目标准确而稳定的跟踪,具有很好的实用性。但是,如果遇到相似的两个目标,比如实验 1 至实验 2 中所用视频中,本方法并不能准确区分三个穿黑色衣服

的人。接下来的工作,可能集中在特征的优化以及根据粒子位置的反馈信息,控制摄像头的转动等方面。

### 参考文献

- 1 Vaswani N. Particle filtering for large dimensional state spaces with multimodal observation likelihoods. *IEEE Trans. on Signal Processing*, 2008, 56(2): 432-440.
- 2 王丹玲,鲁永泉,贾笑捷.实时粒子滤波跟踪算法及其实现. *系统仿真学报*,2009,21(18):5651-5655.
- 3 程建,周越,蔡念.基于粒子滤波的红外目标跟踪. *红外与毫米波学报*,2006,25(2):113-117.
- 4 刘惟锦,章毓晋.基于 Kalman 滤波和边缘直方图的实时目标跟踪. *清华大学学报:自然科学版*,2008,48(7):1104-1107.
- 5 李洪亮,侯朝楨,周绍生.一种高效的改进粒子群优化算法. *计算机工程与应用*,2008,44(1):14-16.
- 6 朱志宇.粒子滤波算法及其应用.北京:科学出版社. 2010:190-215.
- 7 胡士强,敬忠良.粒子滤波原理及其应用.北京:科学出版社.2010:102-148.
- 8 刘晓晨.基于贝叶斯理论的运动目标检测算法研究[硕士学位论文].南京:南京邮电大学,2012.
- 9 陈菲琪,吴晓丹.基于粒子滤波的多目标跟踪研究. *计算机仿真*,2010:147-247.
- 10 冯军华,张荣国,王元龙,刘焜.一种改进的快速轮廓线提取算法. *计算机与数字工程*,2010:25-225.