

文章编号:1001-9081(2005)04-0786-04

## 一种基于彩色信息和抽样检测的视频分割方法

刘鸿伟<sup>1</sup>, 刘 克<sup>2</sup>

(1. 北京科技大学 信息工程学院, 北京 100083; 2. 国家自然科学基金委员会 信息科学部, 北京 100085)  
(idn\_lhw@tom.com)

**摘 要:**提出了一种基于彩色信息和抽样检测的视频分割算法。该算法把每一帧视频图像划分为  $M \times N$  个互不重叠的像素块(信息检测窗),并用矩阵表示,对  $M \times N$  个信息检测窗进行抽样检测,针对检测结果,快速重建背景模型,依据彩色信息和灰度信息分割运动目标。实验结果表明,该算法与已有算法相比,具有分割速度快、适应性强、运动目标分割准确等特点。

**关键词:**自适应背景模型;信息检测窗;运动目标分割;彩色信息

**中图分类号:** TP391.41 **文献标识码:** A

## Efficient video object segmentation algorithm based on color information and sampling detection

LIU Hong-wei<sup>1</sup>, LIU Ke<sup>2</sup>

(1. School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China;  
2. Department of Information Sciences, National Natural Science Foundation of China, Beijing 100085, China)

**Abstract:** Based on the characters of static scenes and block processing techniques, a video segmentation algorithm based on color information and sampling detection was proposed. In the algorithm, every frame was divided into many small regular blocks(Information Detecting Window, IDW) and showed by matrix. According to the detecting results with sampling IDWs, the algorithm quickly rebuild background model and segment moving object with the RGB color and grey scale information. It is shown that the algorithm combines the advantages of less runtime, stronger adaptability and accuracy with comparison to other background subtraction algorithms.

**Key words:** adaptive background model; information detection window; moving object segmentation; color information

### 0 引言

视频分割旨在分割出视频序列中的运动对象,并沿时间轴跟踪运动对象的演进,是最新发展起来的视音编码标准 MPEG-4、MPEG-7 中一项关键技术。

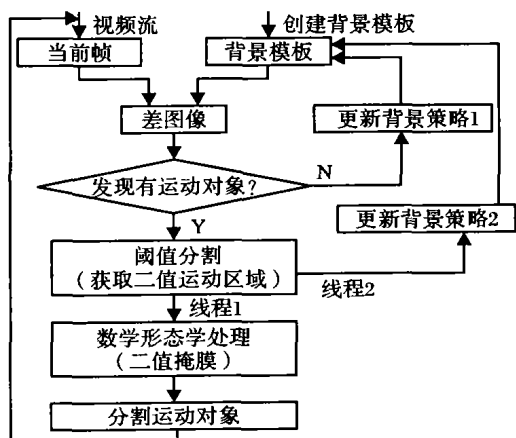


图1 视频分割流程

本文提出了一种抽样检测信息窗的分割算法,为了使检测结果不受随机因素的影响,提高检测速度,考虑把背景模板和当前帧划分为多个信息检测窗,抽样检测信息窗,判断是否存在运动目标区域。根据信息窗的检测结果再决定相应的背

景模板更新策略。图1为视频分割处理流程。

### 1 基于彩色信息和抽样检测的视频分割算法

#### 1.1 信息检测窗的划分

变化检测方法,主要依据是差图像中亮度、色度的变化信息。基于像素的帧差计算虽然简单,但对噪声比较敏感,所以本文把场景划分为  $M \times N$  个互不重叠的像素块( $M$  是行数,  $N$  是列数),即  $M \times N$  个信息检测窗。具体方法如下:把场景  $P_i$  按从左到右,从上到下的空间顺序分成一系列  $k \times k$  ( $k \geq 8$ ) 大小的图像子块,如图2所示,每一帧图像可以用矩阵表示:

$$G_i = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} \quad (1)$$

同样将差分场景  $DP_i$  按同样的方式划分子块并用矩阵表示。基于像素块的帧差计算虽然稍显复杂,但具有抑制噪声的能力,并为以后的模板匹配跟踪打好基础。

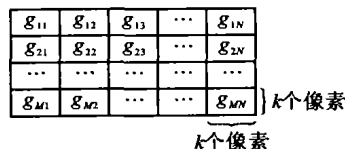


图2 信息检测窗的划分

收稿日期:2004-09-03

**作者简介:**刘鸿伟(1975-),女,河北保定人,硕士研究生,主要研究方向:运动检测跟踪、图像分析与处理、多媒体信息处理;刘克(1965-),男,河南人,教授,博士生导师,主要研究方向:智能信息处理、计算机网络应用、多媒体实时监控、面向对象程序设计。

## 1.2 背景模型的建立和自动更新

为了检测场景中的运动物体,建立一个有效的背景模型是非常重要的。大多摄像头输入的图像是由  $R$ 、 $G$  和  $B$  三个分量组成,假设  $R$ 、 $G$  和  $B$  三个颜色分量和亮度分量相互统计独立,对静态背景中某一个像素进行观察统计,则四个分量均呈高斯分布。

基于背景像素点的统计特性,本文提出了一种基于彩色信息的自适应背景模型,创建方法如下:选择一帧不含任何运动对象的初始背景帧,划分为  $M \times N$  个像素块,每个像素块的大小为  $k \times k (k \geq 8)$  个像素点,并参考式(1)表示成矩阵形式,以统计的方法给像素块中每个像素点建模:设  $\mu'_{m,n}(i,j)$ 、 $\mu''_{m,n}(i,j)$ 、 $\mu^b_{m,n}(i,j)$  分别表示背景模型中第  $m$  行第  $n$  列像素块中的第  $(i,j)$  像素点的  $R$ 、 $G$ 、 $B$  序列均值; $\sigma'_{m,n}(i,j)$ 、 $\sigma''_{m,n}(i,j)$ 、 $\sigma^b_{m,n}(i,j)$  分别为背景模型中第  $m$  行第  $n$  列像素块中的第  $(i,j)$  像素点的  $R$ 、 $G$ 、 $B$  标准偏差; $P'_{i(m,n)}(i,j)$ 、 $P''_{i(m,n)}(i,j)$ 、 $P^b_{i(m,n)}(i,j)$  为第  $t$  帧视频第  $m$  行第  $n$  列像素块中的第  $(i,j)$  像素点的  $R$ 、 $G$ 、 $B$  值,其中第  $(i,j)$  像素点表示该像素块中第  $i$  行第  $j$  列的像素点(块内像素点的扫描顺序为从左到右、从上到下,  $i, j = 0, \dots, k$ )。背景模板初始化时首先把初始背景帧的像素点  $RGB$  值放入  $\mu'_{m,n}(i,j)$ 、 $\mu''_{m,n}(i,j)$ 、 $\mu^b_{m,n}(i,j)$ , 标准偏差  $\sigma'_{m,n}(i,j)$ 、 $\sigma''_{m,n}(i,j)$ 、 $\sigma^b_{m,n}(i,j)$  初始为 0。

本文采用当前帧中判断为背景的像素点与背景模板的加权平均更新背景。令  $P^k_{i(m,n)}(i,j) (k = r, g, b)$  为当前帧(第  $t$  帧)某像素点的  $RGB$  值,  $\mu^k_{m,n}(i,j) (k = r, g, b)$  为背景模板该像素点的序列均值,  $\sigma^k_{m,n}(i,j) (k = r, g, b)$  为背景模板该像素点的标准偏差,更新背景模板基本依据式(2)、(3):

$$\mu^k_{m,n}(i,j) = \alpha * P^k_{i(m,n)}(i,j) + (1 - \alpha) * \mu^k_{m,n}(i,j) \quad (2)$$

$$\sigma^k_{m,n}(i,j) = \alpha * (P^k_{i(m,n)}(i,j) - \mu^k_{m,n}(i,j)) + (1 + \alpha) * \sigma^k_{m,n}(i,j) \quad (3)$$

其中,权值  $\alpha$  的大小影响更新速度,若  $\alpha$  的取值不当,则更新后的背景质量不佳,并且背景模型与将要比较的当前帧中的背景像素相差很大,有可能把当前帧中的背景误判为前景目标,同时也会使更新速度变慢。经过验证,  $\alpha$  取 0.05 ~ 0.01 较佳,而且此时的更新速度对光照变化不敏感。

本文采取两种背景更新策略,一种是用当前帧的所有像素点的值直接更新背景模板,另外一种是用当前帧中被判断为背景点的像素点依据式(2)和式(3)更新背景模板。

## 1.3 光照突变检测和前景像素点的检测

环境光照的缓慢变化以及视频捕获过程中的随机噪声,一般都可以通过式(2)和式(3)的背景更新融合到背景模板中。根据实验分析发现,光照的突变将引起场景中几乎所有像素点亮度值的较大改变,要么都增大、要么都减小,因此可以利用这些特征来随机选取当前帧中一部分像素点进行光照突变检测。

$$\Delta(P) = \left[ \frac{(P'_{i(m,n)}(i,j) - \mu'_{m,n}(i,j))^2}{\sigma'^2_{m,n}(i,j)} + \frac{(P''_{i(m,n)}(i,j) - \mu''_{m,n}(i,j))^2}{\sigma''^2_{m,n}(i,j)} + \frac{(P^b_{i(m,n)}(i,j) - \mu^b_{m,n}(i,j))^2}{\sigma^b^2_{m,n}(i,j)} \right] \quad (8)$$

根据虚警率设置判断阈值  $T_p$  (取值 100 ~ 200), 进行如下判决:

$$V_{i(m,n)}(i,j) = \begin{cases} 0 & \text{若 } \Delta(P) \geq T_p \\ 255 & \text{其他} \end{cases} \quad (9)$$

式(9)中  $V_{i(m,n)}(i,j)$  所有对应于前景目标的像素值为 0 (黑色), 对应于背景的像素值为 255 (白色), 如图 5(b) 所示, 这时获得初期运动区域二值图像。此时应依据式(2)和式

为了提高检测速度,我们采取的方法是:先对当前帧(第  $t$  帧)中的像素块和像素块中的像素点进行隔行或隔列(本文采取隔行)扫描,依据式(4)计算:

$$C_{i(m,n)}(i,j) = 0.299 * P'_{i(m,n)}(i,j) + 0.587 * P''_{i(m,n)}(i,j) + 0.114 * P^b_{i(m,n)}(i,j) \quad (4)$$

其中,  $C_{i(m,n)}(i,j)$  表示第  $t$  帧视频第  $m$  行第  $n$  列像素块中第  $(i,j)$  像素点的亮度值,  $P'_{i(m,n)}(i,j)$ 、 $P''_{i(m,n)}(i,j)$ 、 $P^b_{i(m,n)}(i,j)$  为第  $t$  帧视频该像素点的  $RGB$  值。计算当前帧各扫描点的亮度值  $C_{i(m,n)}(i,j)$ , 并依据公式(4)计算背景模板中相对应点的亮度值  $BC_{i(m,n)}(i,j)$ , 统计每一个像素块中  $|C_{i(m,n)}(i,j) - BC_{i(m,n)}(i,j)| > c$  的像素点的个数, 其中  $c$  是判断阈值, 实验证明其取值为 5 ~ 15, 若  $a\%$  的像素块中  $b\%$  的扫描点满足  $|C_{i(m,n)}(i,j) - BC_{i(m,n)}(i,j)| > c$ , 则可以断定, 此时一定是环境光照发生了突变;  $a$  和  $b$  是调节系数, 它们的取值可依据场景的情况及检测光照突变的速度与误差进行选取, 一般来说, 场景中黑色的部分较多,  $a$  和  $b$  的取值较小, 若加快检测光照突变的速度,  $a$  和  $b$  选取为 95 左右。一旦检测到环境光照发生了突变, 立即更新背景模板, 把背景模板中的  $RGB$  序列均值全部以当前帧像素点的  $RGB$  值代替, 而标准偏差以 0 取代, 从而实现背景的及时更新(第一种更新策略); 若只是非常少的像素点差值  $|C_{i(m,n)}(i,j) - BC_{i(m,n)}(i,j)| > c$ , 且成堆现象很分散, 则可以判定没有任何情况发生, 用当前帧所有像素点依据式(2)和式(3)更新背景模型; 若只是较少部分像素块的像素点的差值  $|C_{i(m,n)}(i,j) - BC_{i(m,n)}(i,j)| > c$ , 而且像素块具有较明显的成堆性, 则可以认为环境光照没有突变, 而是有运动目标出现, 此时应进行下面的前景像素点检测。

某一像素点的  $RGB$  分量均服从高斯分布, 在时间  $t$  的取值为  $X_t$ , 概率密度函数可以表示为:

$$\Pr(X_t) = \frac{\exp\left[-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1}(X_t - \mu)\right]}{(2\pi)^{\frac{3}{2}} |\Delta|^{\frac{1}{2}}} \quad (5)$$

由于  $R$ 、 $G$ 、 $B$  三个颜色分量相互统计独立:

$$\Delta = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_g^2 & 0 \\ 0 & 0 & \sigma_b^2 \end{bmatrix} \quad (6)$$

于是概率密度函数写为:

$$\Pr(X_t) = \prod_{i=1}^3 \frac{1}{\sqrt{2\pi}\sigma_i^2} \exp\left[-\frac{1}{2} \frac{(x_i - \mu)^2}{\sigma_i^2}\right] \quad (7)$$

用当前帧与背景模板相减, 得到差分图像, 然后对差分图像进行阈值分割。参考式(7)确定的像素点取值的概率分布, 计算  $\Delta(P)$ :

(3) 对背景模板进行自动更新(第二种更新策略)。

## 1.4 消除误判为前景点的阴影

上述提取的前景区域很可能包括一些被误判为前景点的物体阴影, 为了消除阴影, 不仅采用阈值  $T_p$  判断, 再增加一个阴影检测阈值  $T_s$ , 双阈值提取前景点。像素  $RGB$  值混合了亮度和色度信息, 对于前景物体, 像素点的  $RGB$  值不同于背景像素点的值, 而阴影像素点的值与相应的背景点的值却有相

似的色度。这是因为,阴影可以看作是图像的半透明区域,仍然保留着对应物体表面的光学特性,如模式、纹理、颜色等<sup>[7]</sup>。为了区分阴影和前景点,文献[7]设计了一种有效的计算颜色模型(如图3所示),计算像素 $X$ 的色度失真度可以表示为:

$$C(X) = \beta = \arccos\left(\frac{OI \cdot OI_b}{|OI| |OI_b|}\right) \quad (10)$$

其中 $I_b = [ER(X); EG(X); EB(X)]$ ,表示背景模型中RGB的序列均值。向量 $OI_b$ 表示期望的背景颜色向量。 $I = [R(X); G(X); B(X)]$ 表示像素在当前帧中的RGB值。

如果 $C(X) < T_c$ ,则断定该像素点在阴影区, $T_c$ 值可以根据经验设定(参考 $T_c = \pi/180$ )。此阈值判定与上述 $T_p$ 阈值相结合分割检测前景点,可以减少阴影误判。

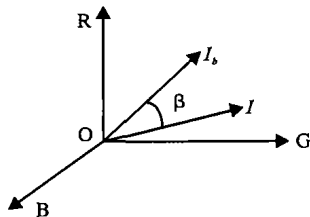


图3 计算颜色模型

### 1.5 分割运动目标

在1.3节中初步获得前景像素点二值区域,这时,运动区域可能会产生裂缝和空洞,需要应用形态学的开、闭运算和腐蚀、膨胀基本算子对该二值图像作进一步处理。然后再进行区域标注的连通检测处理,标记出各连通块,并将包含像素数目小于指定阈值的区域视为噪声块删去。

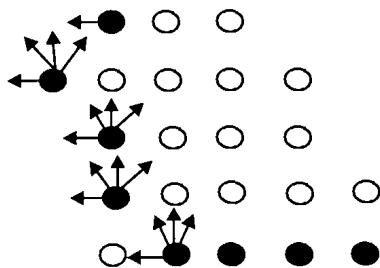


图4 连通检测算法

区域标注采用8-连通的情况,假设一幅二值图像从左到右,从上到下扫描,按此探测准则找到最左上方的边界点 $X_i$ ,如果这一点的像素值为0,以这个边界点起始,检查它的左、上、左上、右上四个近邻像素,分别为 $X_l$ 、 $X_u$ 、 $X_{lu}$ 、 $X_{ru}$ ,如果它们的值都是255,则给当前像素 $X_i$ 记一个新的标记 $F_i$ ;如果四个近邻像素其中只有一个为0,如 $X_l = 0$ ,就把该像素 $X_i$ 的标记 $F_i$ 赋给当前像素 $X_l$ ;如果其中有两个或多个值为0且具有相同的标记 $F_i$ ,就将该标记 $F_i$ 赋给当前像素 $X_i$ ;如果两个或多个值为0且具有不同的标记,就将其中任一标记 $F$ 赋给当前像素 $X_i$ 并表明这两个标记等价,即 $F = F_i$ 。在扫描结束时,所有像素值为0的点都已标记但有的是等价的。将所有等价的标记对归入等价组,对各个组赋以不同的标记,然后进行第二次扫描图像,将每个标记用它所在等价组的标记代替,连通检测算法如图4所示。记录每个前景区域的面积 $Area$ ,即前景像素点的数目。

当 $Area < Th$ 时,即认为是噪声块,把此区域融合到背景区域中,从而消除噪声块。其中, $Th$ 的选择与要检测的运动物体的大小有关,是经验值。经过去除噪声块处理后,得到最终的运动区域二值图像,即分割对象模板或者二值掩膜图像,如图5(c)所示。根据这个模板可最终分割出运动对象,如图5(d)所示。

## 2 试验结果与分析

为了验证基于RGB空间自适应背景模板的视频分割算法,将该算法运行在PIV 1.3G Windows 2000平台上,对连续200帧 $360 \times 288$ 图像序列进行处理,其中图像序列中的人分别在白天和傍晚做快速、缓慢,连续、间断的走动或跑动。图5给出了本算法的全部实验序列图像,其中图5(b)为初期 $T_p$ 阈值的分割效果,明显存在一些噪声点,图5(c)是经过形态学处理和区域标识检测后的结果,有效去除了噪声点,图5(d)是最后的运动对象分割结果。该算法的分割速度能达到15Frame/s,处理一帧图像所用的全部时间基本上在66ms左右,处理速度达到了视频对象分割的实时性要求。实验过程中经验值的选定相对比较简单,反复调整3、4次,即可获得满意的效果,其中选择噪声块的判断阈值 $Th$ 与运动目标在视频中所占的比例有关,其他均可参考上文给出的范围。本算法与已有的亮度变化检测算法相比,不仅可以保证目标分割的准确性和实时性,而且应用像素点的原始RGB值的检测受光照影响较小,可以降低阴影误判为前景点的几率。

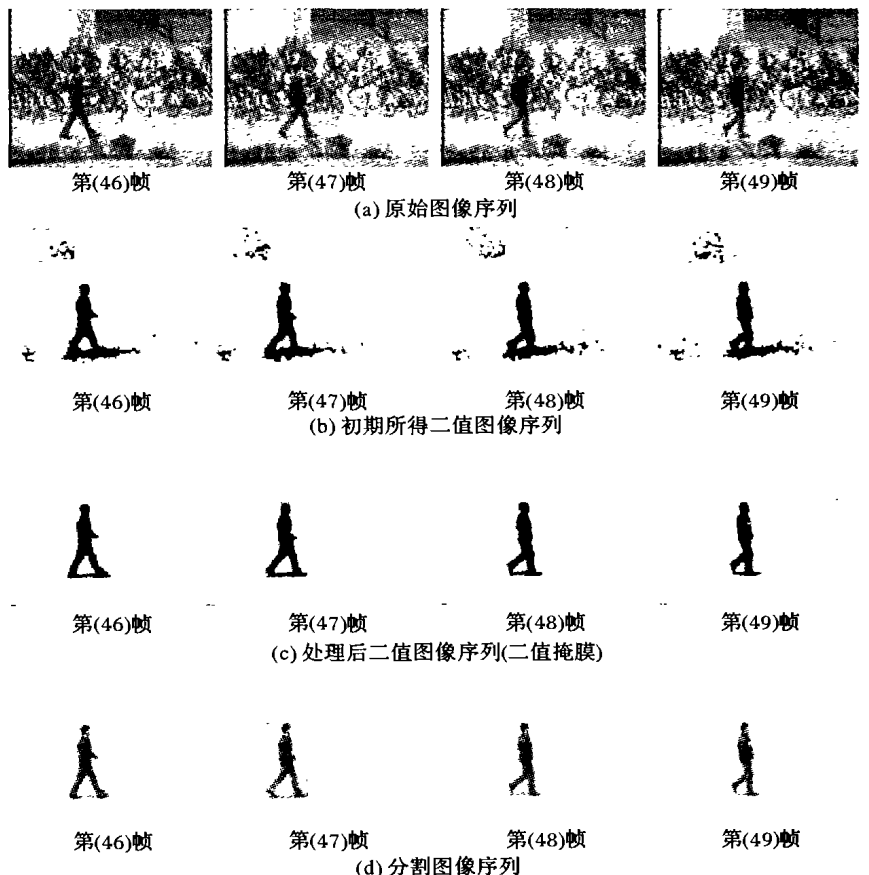


图5 序列图像运动目标检测结果

## 3 结语

本文针对静止背景的视频序列特性,提出了基于彩色信息和抽样检测的分割算法,该算法不同于通常背景差算法中采用亮度值的差异分割前景对象,而是根据序列图像中像素

点的 RGB 值的统计特性更新背景并分割运动物体。本算法一方面通过把视频图像划分为多个信息检测窗,使检测更趋于智能化,提高了检测速度;另一方面,采用基于 RGB 彩色信息的自适应背景模板,能精确的提取出运动目标,还具有很强的抗噪声能力。实验证明,该算法达到了准确分割运动目标的目的。

#### 参考文献:

- [1] ADIV G. Determining three-dimensional motion and structure from optical flow generated by several moving object[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985, 7(4): 384 - 401.
- [2] BOUTHEMY P, FRANCOIS E. Motion segmentation and qualitative dynamic scene analysis from an image sequence[J]. International Journal on Computer Vision, 1993, 10(2): 157 - 183.

- [3] CHOI JG, LEE WS, KIM SD. Spatio-temporal video segmentation using a joint similarity measure[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(2): 279 - 286.
- [4] ARCH T, KAUP A. Statistical model-based change detection in moving video[J]. Signal Processing, 1993, 31(3): 165 - 180.
- [5] XIAO DG, YU SS, ZHOU JL. Motion Tracking with Fast Adaptive Background Subtraction[J]. Wuhan University Journal of Natural Sciences, 2003, 1A(8): 035 - 040.
- [6] TSAIG Y, AVERBUCH A. Automatic segmentation of moving objects in video sequences: a region labeling approach[J]. IEEE Trans on Circuit and Systems for Video Technology, 2002, 12(7): 597 - 612.
- [7] 韩洪哲, 王志良, 等. 阴影消除的自适应背景建模[A]. 第二届全国智能视觉监控学术会议, 智能视觉监控研究进展——第二届全国智能视觉监控学术会议论文集[C]. 北京: 科学出版社, 2003. 35 - 39.

(上接第 738 页)

当  $(f_{\max} - f_i) \geq (f_{\max} - f_{avg})$  时,  $p_i = K_1 \times n \times (f_{\max} - f_i) / (f_{\max} - f_{avg})$

当  $(f_{\max} - f_i) < (f_{\max} - f_{avg})$  时,  $p_i = K_2$

公式中  $f_{\max}$  为当前代中适应度的最大值,  $f_{avg}$  为当前代中适应度的数学期望,  $f_i$  为当代中第  $i$  个个体的适应度,  $f_{\max} - f_{avg}$  表示当前代的适应度范围,  $f_{\max} - f_i$  表示个体的适应度与适应度最大值之间的差异,  $n$  为染色体的位串长,  $(f_{\max} - f_i) / (f_{\max} - f_{avg})$  表示第  $i$  个个体在当代中的优劣程度,  $K_1, K_2$  为调整系数。自适应变异算子可以在进化群体收敛到局部最优解附近时(即  $f_{\max} - f_{avg}$  较小),使变异率增大,增加种群的多样性;而当进化种群在解空间中较分散时(即  $f_{\max} - f_{avg}$  较大),使变异率变小,这样可以加快种群的收敛速度。

#### 2.7 算法停止准则

在仿真实验中,算法以预先设定的最大繁殖代数作为停止准则。合理地确定繁殖代数可以保证算法的效率和取得合理可靠的解。

### 3 实验分析

本文的算法已经编程实现,选择的编程环境是 Windows 和 VC 6.0。两个实验是针对一个有 60 个顶点的加权有向图设计的,不失一般性求顶点 1 到顶点 60 的最优路径。实验一为固定概率交叉算子和变异算子的遗传算法;实验二为用模拟退火算法改进的遗传算法。实验二所取各参数如下:群体规模为 100,退火初始温度  $T_0$  为 1000;温度冷却参数  $\alpha$  为 0.8。  $K_1, K_2$  调整系数取 0.618 和 0.382。实验结果如表 1 和图 1 所示。

表 1 遗传算法优化结果

迭代次数	最优路径	适应函数
5	1, 2, 3, 4, 19, 22, 27, 36, 40, 57, 60	0.0093
20	1, 4, 10, 16, 27, 36, 41, 51, 60	0.053
30	1, 4, 10, 16, 57, 60	0.074
50	1, 4, 14, 57, 60	0.086

实验结果表明,算法在最初几次迭代中的效果并不理想,个体的适应度也不高,但随着迭代次数的增加适应度高的个体将被优化出来,适应度也逐渐提高。从图 1 的比较中可以看出改进的遗传算法的适应度有着明显改善。

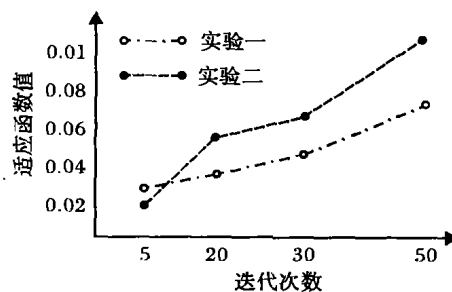


图 1 适应函数比较

### 4 结语

通过对传统遗传算法和模拟退火算法基本原理的研究,将模拟退火的思想引入遗传算法来解决应急决策系统中的最优路径问题,详细讨论了算法的实现和性能,提出了一种新的遗传算子,即插入式交叉算子和自适应变异算子,不但缓解了遗传算法的选择压力,而且增强了进化算法的全局收敛性,加快了算法的收敛速度。仿真实验证明该算法在解决决策支持系统中的最优路径是可行的,对提高应急决策的效率有一定的帮助。算法使得 SA 与 GA 结合,能很好的利用再生算子,力求保持种群的多样性,这是一个初步的比较直接的思想,算法的有效性还有待于在更广泛的应用问题上加以测试。

#### 参考文献:

- [1] 戴更新, 达庆利. 多资源组合应急调度问题的研究[J]. 系统工程理论与实践, 2000, 20(9): 52 - 55.
- [2] GOLDBERG DE. Genetic algorithms in search, optimization and machine learning[M]. New York: Addison-Wesley, 1999.
- [3] MAHFOUND SW, GOLBERG DE. A genetic algorithm for parallel simulated annealing[A]. Proceedings of International conference on Parallel Problem Solving from Nature[C]. Netherland, 1999.
- [4] 高淑萍, 刘三阳. 应急系统调度问题的最优决策[J]. 系统工程与电子技术, 2003, 10: 1022 - 1024.
- [5] REES J, KOEHLER GJ. Leadership and group search in group decision support system[J]. Decision Support System and Electronic Commerce, 2000, (30): 73 - 821.
- [6] 肖美华, 王命延. 基于遗传算法和模拟退火算法的布局问题研究[J]. 计算机工程与应用, 2003, 39(36): 70 - 72.
- [7] 高经纬, 张煦. 求解 TSP 问题的遗传算法实现[J]. 计算机时代, 2004, (2): 19 - 21.