

·信号与信息处理·

一种图像相关匹配算法的改进及实现

宋刘非, 郑 丽

(东北电子技术研究所, 辽宁 锦州 121000)

摘要:序贯相似性检测算法(SSDA)是针对传统模板匹配算法提出的一种高效的图像匹配算法,对SSDA算法进行改进,利用序列来代替固定阈值,采用隔点采样以及粗匹配和精匹配相结合,进一步提高了计算效率。并在此基础上,采用Visual C++ 6.0对该算法进行了编程实现。实验证明,该算法比传统的SSDA算法计算速度更快、匹配误差更小。

关键词: Visual C++; SSDA; 序贯相似性检测算法; 图像匹配

中图分类号: TP317.4

文献标识码: A

文章编号: 1673-1255(2013)-04-0061-04

Improvement and Implementation of Image Correlation Matching Algorithm

SONG Yi-fei, ZHENG Li

(Northeast Research Institute of Electronics Technology, Jinzhou 121000, China)

Abstract: A high efficient sequential similarity detection algorithm (SSDA) is proposed compared with the traditional mould board matching algorithm. SSDA is improved such as using sequence instead of fixed threshold, dot sampling, coarse and fine matching combination so as to increase calculation speed further. Based on this, Visual C++ 6.0 is used to implement the algorithm programming. Experimental results show that the algorithm has faster calculation speed and less matching error comparing with traditional algorithm.

Key words: Visual C++; sequential similarity detection algorithm (SSDA); image matching

图像匹配技术是根据已知的图像模块(模板图)在另一幅图像(搜索图)中寻找相应或相近模块的过程,它是计算机视觉和模式识别中的基本手段。已在卫星遥感、空间飞行器的自动导航、机器人视觉、气象云图分析及医学X射线图片处理等许多领域中得到了广泛的应用^[1]。研究表明,图像匹配的速度主要取决于匹配算法的搜索策略。传统的模板匹配算法的基本搜索策略是遍历性的,为了找到最优匹配点,传统方法均必须在搜索区域内的每一个像素点上区域相关匹配计算,图像相关匹配的数据量和计算量很大,匹配速度较慢。为了提高模板图像匹配效率,提出了一种快速图像匹配算法——序贯相似性检测算法(简称SSDA)^[2]。对SSDA算法作

进一步讨论,并在此基础之上提出一种新的图像相关匹配算法。

1 序贯相似性检测算法

传统模板匹配算法在每个待匹配位置上都要做 M^2 (M 为模板的大小,如图1所示)次相关计算,计算量很大。为了减少在每个待匹配位置上的计算量,人们提出了序贯相似性检测算法。序贯相似性检测算法是对传统模板匹配算法的改进,其要点如下。

设模板 T 叠放在搜索图 S 上平移,模板覆盖下的那块搜索图设为子图 $S^{i,j}$, i,j 为这块子图的左上角像素在 S 图中的坐标,则 i 和 j 的取值范围为 $1 \leq i、$

$j \leq N - M + 1$ 。

(1)定义绝对误差值

$$\epsilon(i, j, m_k, n_k) = |S^{i,j}(m_k, n_k) - \hat{S}^{i,j}(i, j) - T(m_k, n_k) + \hat{T}| \quad (1)$$

其中, $\hat{S}^{i,j}(i, j)$ 为模板覆盖下的搜索子图的像素平均值, \hat{T} 为模板位图像素的平均值。

$$\hat{S}^{i,j}(i, j) = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^M S^{i,j}(m, n) \quad (2)$$

$$\hat{T}(i, j) = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^M T(m, n) \quad (3)$$

(2)取一个不变的阈值 T_k

(3)在子图中 $S^{i,j}(m, n)$ 随机选取像素点, 计算它同 T 中对应点的误差值 ϵ , 然后把这点的差值同其他点对的差值累加起来, 当累加 R 次误差超过 T_k , 则停止累加, 并记下累加次数 R , 定义 SSDA 的检测曲面为

$$I(i, j) = \left\{ r \mid \min_{l \leq r \leq m^i} [\epsilon(i, j, m_k, n_k) \geq T_k] \right\} \quad (4)$$

(4)把 $I(i, j)$ 值最大的点定为匹配点, 因为在这一点上需要很多次累加才能使总误差 $\sum \epsilon$ 超过 T_k 。

SSDA 算法以随机不重复的顺序选取像素对, 利用上述步骤进行计算, 计算部分误差和 ϵ , 只要总误差超过阈值 T_k , 停止本次计算, 进入下一个位置的测试。

2 改进算法设计思路及软件实现

为了提高算法的计算效率, 对 SSDA 算法进行了改进, 利用序列来代替固定阈值以达到阈值的目的。采用隔点采样以及先粗后细的匹配搜索策略, 可以进一步提高计算效率^[5]。

2.1 系统流程图

这种 SSDA 改进算法的实现难度要高于模板匹配算法, 流程图如图 1 所示。

2.2 系统实现

设搜索子图和模板图像的相似程度的计算公式为

$$D(x, y) = \frac{1}{M^2} \sum_{i=0}^M \sum_{j=0}^M |S(x+i, y+i) - T(i, j)| \quad (5)$$

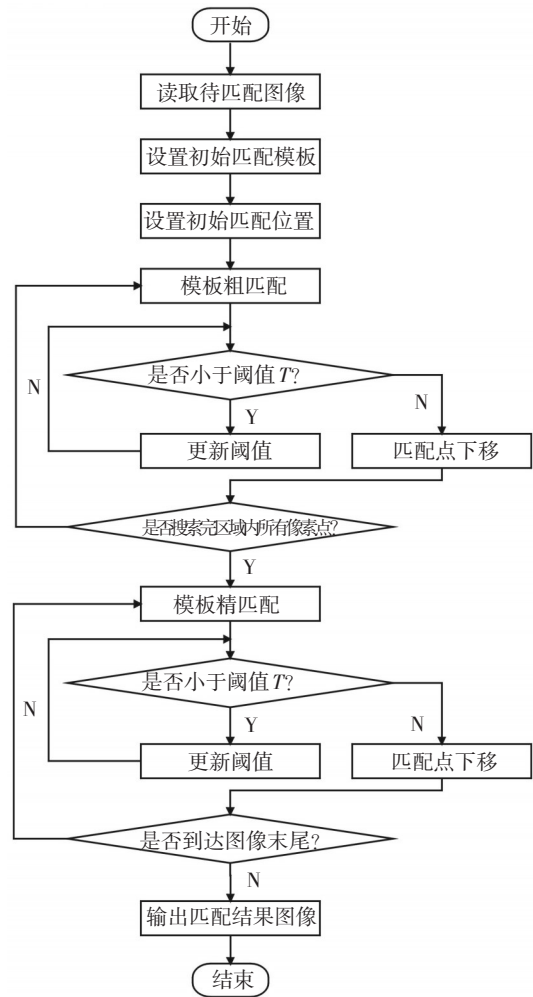


图1 SSDA 改进算法的流程图

(1)粗匹配过程

①在搜索区域内随机选取一点 (x, y) , 利用式 (5) 计算模板图像与初搜索子图中隔 m 行和 n 列 (通过反复实验得知, 每行和每列中参与计算的像素点数要超过 15 个像素点) 对应像素点的 $D(x, y)$ 作为初始阈值。

②从初始点 $(0, 0)$ 开始, 同样计算模板图像与搜索子图隔 m 行和 n 列像素点的 $D(x, y)$, 如果在计算过程中 $D(x, y)$ 大于阈值 T , 则停止本次累加计算, 转入下一像素点。如果计算完所有像素点后 $D(x, y)$ 小于阈值 T , 则记录此位置为相关匹配点, 并用 $D(x, y)$ 值来更新阈值 T 。其中阈值变化如式 (6) 所示

$$T = \begin{cases} T & T > D(x, y) \\ D(x, y) & T \leq D(x, y) \end{cases} \quad (6)$$

粗匹配过程的部分程序代码如下

Pt $(x, y) = \text{RandomPos}()$; // 得到搜索区域内随机点

```

T = CompDis ( x , y ); //计算点 ( x , y ) 处的相关系数作为
阈值初始值(隔点采样)
for ( i = 0; i < SearchHeight; i ++ )
{
for ( j = 0; j < SearchWidth; j ++ )
{
temp = 0; //存储相关系数值
for ( m = 0; m < ImageHeight; m = m + 2ImageHeight/15; m +
+ )
{
for ( n = 0; n < ImageWidth; n = n + 2ImageW idth /15; n +
+ )
{
temp=temp+|ImageData[m][n]-match2ImageData[i+m][i+n]|;
}
if ( temp > T ) //累加值大于阈值时停止循环
Break;
}
if ( temp < T )
{ //更新阈值 T 及粗相关匹配点
T = temp;
TargetX = j;
TargetY = i;
}
}
}

```

这样,在计算完搜索区域内所有像素点时即可得到粗相关匹配点。算法在运行过程中阈值是自适应单调递减序列,阈值逐渐逼近最佳阈值,从而使非匹配区域经过少量的计算就超过阈值而被丢弃。于是采用随机选取初始点和采用隔点采样的方式,可以将算法的复杂度降低为 $1/mn$,并且不会影响算法的精度。

(2) 精匹配过程

① 假设在粗匹配过程中得到的粗匹配相关匹配点为 $P_t(x, y)$,利用式(5)计算搜索子图与模板图像所有对应点的 $D(x, y)$ 作为初始阈值。

② 以粗匹配相关点为中心,在 $2n \times 2m$ 的矩形区域内寻找最优匹配点,其匹配过程中利用式(5)计算矩形区域内所有像素的 $D(x, y)$,阈值和最佳匹配的更新与粗匹配过程相同。

精匹配的部分算法如下:

```

T = CompDis ( TargetX , TargetY ); //计算点 ( TargetX , Tar-
getY ) 处的相关系数作为阈值初始值
for ( i = TargetY - ImageHeight/15; i < TargetY2pos + Image-
Height/15; i ++ )

```

```

{ //在区域内查找最优匹配点
for ( j = TargetX - ImageWidth/15; j < TargetX2pos + Image-
Width; j ++ )
{
temp = 0; //存储相关系数值
for ( m = 0; m < ImageHeight; m = m + 2ImageHeight/15; m ++ )
{
for ( n = 0; n < ImageWidth; n = n + ImageWidth/15; n ++ )
{
temp = temp + |ImageData[m][n] - match2ImageData[i + m][j + n]|;
}
if ( temp > T ) //累加值大于阈值时停止循环
Break;
}
if ( temp < T )
{
//更新阈值 T 及粗相关匹配点,和粗匹配更新过程相同
}
}
}
ShowMsg ( ); //输出匹配结果

```

这样,遍历完 $2n \times 2m$ 矩形区域内所有点后,即可最终得到精相关匹配点。由于精匹配过程中搜索子图与模板图像所有对应像素都参与计算,达到了精确的目的,同时,把计算区域限定在 $2n \times 2m$ 的矩形区域,从而整体上减少了计算量。

2.3 实验结果分析

为了验证所采用算法的高速性,对传统的归一化算法、固定阈值的 SSDA 算法以及自适应阈值的 SSDA 算法在处理时间上做了比较。实验环境为: P43.0 G 处理器, 1G 内存, WINDOWS XP 系统, VC++ 6.0 编程软件。软件系统运行界面如图 2 所示。



图2 软件运行界面

实验采用图3作搜索图 S 。在搜索图 S 中截取 50×40 的子图像作为模板图,如图4所示,其中模板图左上角位于搜索图的 $(230, 260)$ 坐标点上。处理结果见表1。



图3 搜索图

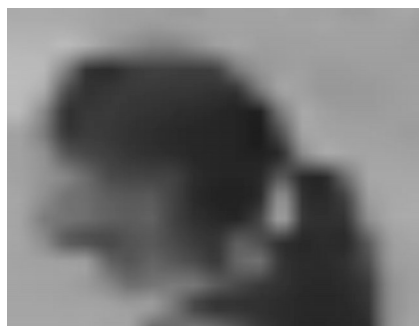


图4 模板图



图5 匹配结果

表1 各种算法处理时间比较

算法	匹配位置	匹配时间/ms
归一化算法	(230,260)	2 250
固定阈值SSDA算法	(230,260)	724
自适应阈值SSDA算法	(230,260)	232

实验结果表明,在不存在失真和噪声的情况下,这种图像相关匹配的改进算法表现出良好的性能,自适应阈值法采用单调递减阈值序列以及粗细分层匹配方法,粗匹配过程中采用隔点采样,降低了计算量,精匹配过程又保证了算法的精度。

3 总结

自适应阈值的SSDA算法总的思想都是从减少搜索点数和比较次数以降低配准时间,从而达到快速匹配的目的。由于已成功应用于实际项目中,表明了该算法的可行性与有效性。

参考文献

- [1] 牛蕾,倪林. 基于ROI的压缩域多谱段遥感图像的检索[J]. 中国图像图形学报,2005,10(10):1212-1217.
- [2] 龚声蓉,刘纯平,王强. 数字图像处理与分析[M]. 北京:清华大学出版社,2006.
- [3] Boliek M, Chistopoulos C, Majani E. J PEG2000 Part I Final Committee Draft Version 1.0. ISO/IECJ TC 1/SC 29/WG 1 N1646R[Z],2000.
- [4] Barbara Zitova, Jan Flusher. Image registration methods: asurvey[J]. Image and Vision Computing, 2003(21): 977-1000.
- [5] BARNEA D I, SILVERMAN HE. A class of algorithms for digital image registration [J]. IEEE, 1972, 21(2): 179-186.
- [6] 刘晓光,陈曦,陈政伟,等. 基于图像灰度的SSDA匹配算法[J].航空计算技术,2010,40(1):54-57.
- [7] 何斌,马天予,王运坚,等. Visual C++数字图像处理[M]. 北京:人民邮电出版社.
- [8] 王立新,刘彤宇,李阳. SSDA图像匹配算法的研究及实现[J]. 光电技术应用,2005,20(3):53-55.