

# 自适应移动宽线检测方法

曲智国<sup>1</sup> 谭贤四<sup>1</sup> 李志淮<sup>2</sup> 王红<sup>1</sup> 林强<sup>1</sup>

(空军预警学院三系 武汉 430019)<sup>1</sup> (空军预警学院雷达士官学校 武汉 430019)<sup>2</sup>

**摘要** 为消除基本宽线检测算子中的冗余计算量,提高其运算速度,提出了一种快速的宽线算子实现方法——自适应移动宽线算子。基本宽线算子采取逐像素移动的方式来检测图像中的宽线特征;自适应移动宽线算子与之不同,在检测宽线像素时能够根据当前像素类型自适应地确定模板移动的步长,从而加快模板的移动速度,较好地消除了基本算法中的冗余运算。利用测试图像进行的实验的结果表明,自适应移动宽线算子在取得相当检测性能的同时,提高了基本宽线算子的运算速度。

**关键词** 宽线特征,宽线算子,快速实现,冗余计算

**中图分类号** TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.03.045

## Adaptive Moving Wide Line Detection Algorithm

QU Zhi-guo<sup>1</sup> TAN Xian-si<sup>1</sup> LI Zhi-huai<sup>2</sup> WANG Hong<sup>1</sup> LIN Qiang<sup>1</sup>

(Department No. 3, Air Force Early Warning Academy, Wuhan 430019, China)<sup>1</sup>

(Radar Sergeant School, Air Force Early Warning Academy, Wuhan 430019, China)<sup>2</sup>

**Abstract** In this paper, a fast implementation of wide line detector was proposed to eliminate its computation redundancy and improve its computational speed, namely adaptive moving wide line detector. Instead of moving the circular mask pixel by pixel as done in the basic implementation, the adaptive moving wide line detector determines its step adaptively according to the current type of pixel under test. In this way, the computational redundancy can be decreased to a large extent so as to accelerate the detector. Simulated and real images were adopted for performance test of the proposed adaptive moving wide line detector. Experimental results demonstrate that the fast implementation accelerates the wide line detector significantly while keeping its detection performance unaffected.

**Keywords** Wide line feature, Wide line detector, Fast implementation, Computational redundancy

宽线是图像中一种常用的线特征,宽度一般小于 10 个像素,具有三角型或方波型屋脊的灰度结构,在图像匹配、目标识别等任务中具有广泛的应用。对于宽线特征,传统的边缘算子处理往往会在宽线两侧检测到两条相邻的边缘,使得特征的数量增加,且在匹配时容易混淆,这对图像匹配任务是不利的。宽线检测算子可以很好地解决这一问题。

目前,学者们提出了一些宽线检测方法,大致可以分为:中心线检测法<sup>[1-4]</sup>和区域检测法<sup>[5-7]</sup>。中心线检测法一般通过检测屋脊(Ridge Detection)得到宽线的中心线,而区域检测法则首先将宽线结构的所有像素检测出来,然后再根据需要确定其中心线。在区域检测法中,Liu 等人<sup>[6-7]</sup>提出的 INF(Isotropic Nonlinear Filtering)宽线算子的性能较好。在本质上,该方法的思想和 SUSAN 算子<sup>[8]</sup>相似,将一圆形模板在图像中逐像素滑动,在每个像素处计算该像素的 USAN(Uni-value Segment Assimilating Nucleus)值(响应值),然后将 US-

AN 值与某一几何门限进行比较,判断当前像素是否为宽线上的像素。但是,该方法在实现过程中存在着冗余计算,影响了算法的运算速度。为了消除算法中的冗余计算,降低宽线算子的运算量,借鉴串匹配 KMP 算法<sup>[9]</sup>和 BM 算法<sup>[10]</sup>根据“部分匹配”信息自适应地改变子串移动量的思想,提出了一种宽线算子快速实现方法——自适应移动宽线算子,简称为 AM-INF 算子(Adaptive Moving INF Detector)。该方法在检测当前像素类别的同时,根据当前像素类别自适应地确定下一步模板移动的步长,从而消除基本 INF 算子中冗余响应的计算,提高了算子的速度。

## 1 INF 宽线算子的冗余性分析

### 1.1 算法思想

假设宽线为方波型的灰度结构(见图 1),将圆形模板放置在图像上并滑动以检测宽线像素,其与宽线的不同位置关

到稿日期:2017-03-28 返修日期:2017-06-11 本文受国家自然科学基金(61401504),博士后科学基金资助面上项目(2014M562562)资助。  
曲智国(1982—),男,博士后,主要研究方向为预警监视、图像处理和目标识别等,E-mail:green20001@sina.com(通信作者);谭贤四(1967—),男,教授,博士生导师,主要研究方向为预警探测体系论证等,E-mail:Tanxs\_2015@163.com;李志淮(1983—),男,博士,讲师,主要研究方向为预警监视、目标检测识别等,E-mail:lizhihuai1983@163.com;王红(1962—),女,教授,博士生导师,主要研究方向为预警监视、预警探测装备作战运用等,E-mail:Wangh\_572g@163.com;林强(1971—),男,教授,硕士生导师,主要研究方向为预警监视、空管雷达建设等,E-mail:linqiang@163.com。

系分别如图 1 中的 a-e 所示。

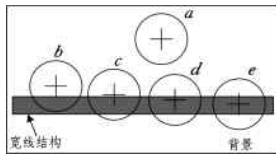


图 1 宽线模型及位于不同位置的模板示意图

Fig.1 Illustration of wide line model and circular masks

与 SUSAN 算子类似,通过比较模板中心像素(核 Nucleus)和模板其他像素的灰度大小即可确定 USAN 区域<sup>[8]</sup>,即模板中与中心像素灰度值相同或相近的像素组成的区域:

$$s(x, y, x_0, y_0, t) = \begin{cases} 1, & \text{if } |I(x, y) - I(x_0, y_0)| \leq t \\ 0, & \text{if } |I(x, y) - I(x_0, y_0)| > t \end{cases} \quad (1)$$

其中,  $(x_0, y_0)$  为圆形模板中心像素的坐标,  $(x, y)$  为模板内其他像素的坐标,  $I(x, y)$  为像素  $(x, y)$  的灰度,  $t$  为灰度阈值,  $r$  为模板半径。

根据式(1)可确定模板中的每个像素与中心像素是否属于同一类别(宽线像素或背景像素),并计算 USAN 区域的大小,即算子的初始响应值:

$$m(x_0, y_0) = \sum_{\substack{x_0-r \leq x \leq x_0+r \\ y_0-r \leq y \leq y_0+r}} s(x, y, x_0, y_0, r) \quad (2)$$

基于初始响应值,宽线像素可根据式(3)进行检测:

$$L(x_0, y_0) = \begin{cases} g - m(x_0, y_0), & \text{if } m(x_0, y_0) < g \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

其中,  $g$  为几何门限,一般设为  $g = m_{\max} / 2$ ;  $m_{\max}$  为 USAN 区域可能取得的最大值。

### 1.2 冗余性分析

假设图像大小为  $M \times N$ , 圆形模板半径为  $r$ , 基本 INF 宽线算子的实现过程为:将圆形模板放置在图像上,对模板中心所在的像素,采用式(1)一式(3)计算其响应值并判定像素类别(宽线像素或背景像素),逐个像素地滑动模板直至访问完图像中的所有像素。为讨论方便,将利用式(1)一式(3)对每个像素的计算称为宽线响应值计算(Response Calculation, RC),其时间复杂度为  $O(r^2)$ ,则基本 INF 宽线算子的总运算量为  $k(k=MN)$  次 RC。

事实上,模板 USAN 区域的像素类别相同,模板 USAN 区域不仅反映了中心像素的类别,同时也表明了 USAN 区域其他像素的类别。如图 2 所示,当圆形模板的中心  $O$  位于均匀背景像素、宽线特征上时,由式(3)可判断中心像素的类别,同时也可判断 USAN 区域其他像素的类别,两种情况下的类别分别有背景像素和宽线像素。但是,基本 INF 算子的基本实现过程仅利用 USAN 区域判断中心像素的类别,忽略了对

模板中形成 USAN 区域的其他像素类别信息的利用,在移动模板时每次仅移动一个像素,导致实现过程中包含冗余计算量。显然,如果能够利用 USAN 区域中像素类别相同这一信息,可以省略对 USAN 区域内其他像素的 RC 计算从而直接判断其像素类型,进而加快圆形模板的移动,不再是每次移动一个像素。本文正是通过减少冗余的 RC 计算次数来设计快速算法,即使  $O(kr^2)$  中的  $k$  值小于  $MN$ 。

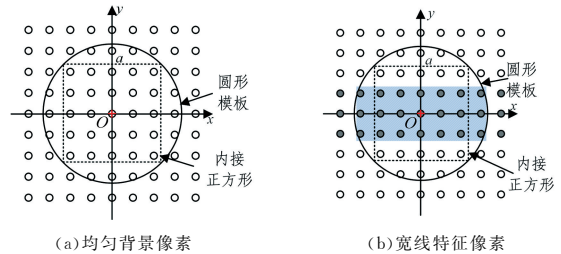


图 2 不同像素类型的 USAN 区域示意图

Fig.2 USAN area formed by different types of pixels

## 2 AM-INF 宽线算子

在串匹配算法 KMP<sup>[9]</sup> 和 BM<sup>[10]</sup> 中,为了减少子串移动次数和字符的比较次数,在移动子串时利用已经得到的“部分匹配”信息将子串向右移动尽可能多的字符,而不是仅移动一个字符,从而降低算法的运行时间。借鉴这一思想,AM-INF 宽线算子根据当前像素类型标记 USAN 区域所有其他像素的类型,省略对这些像素的 RC 计算,并自适应地确定滑动步长(Step Length, SL)移动模板。若模板中心像素为背景像素,则 SL 的取值可以大一些,以加快模板的移动速度,如图 2(a)所示,原则上 SL 最大可取  $r$ ,实现中为避免锯齿效应,SL 取值为圆形模板内接正方形边长的一半,即  $SL = a/2 = \sqrt{2}r/2$ 。若模板的中心像素为宽线特征,则 SL 的取值可以小一些,以防止漏检宽线像素。

### 2.1 算法实现

AM-INF 宽线算子与基本算子的模板移动顺序类似,从图像左上角开始依次从左向右、自上而下移动,不同的是根据当前像素类型和 USAN 区域自适应地确定模板移动步长(列自适应步长 CSL 和行自适应步长 RSL)。

根据模板的移动顺序确定自适应步长时只需关注模板的象限 IV 即可,如图 3、图 4 所示。若当前中心像素为宽线,则标记模板 USAN 区域为宽线特征;若当前中心像素为背景,则标记模板 USAN 区域为背景。根据象限 IV(包含  $x$  轴和  $y$  轴)、USAN 区域和 Non-USAN 区域的边界,来确定 CSL 和 RSL。

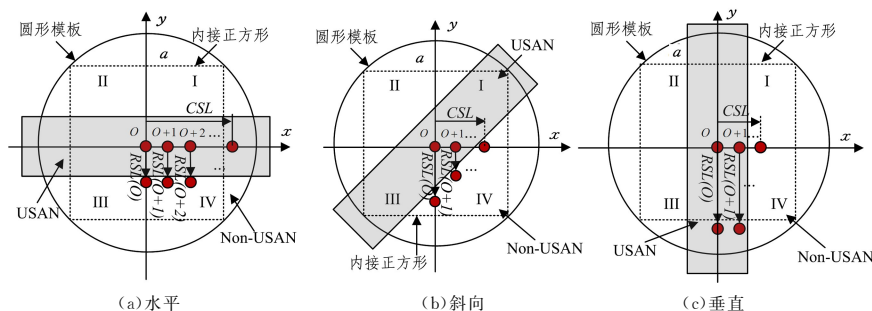


图 3 宽线像素的自适应步长 CSL 和 RSL 的确定

Fig.3 Determining of adaptive step length CSL and RSL for wide line pixels

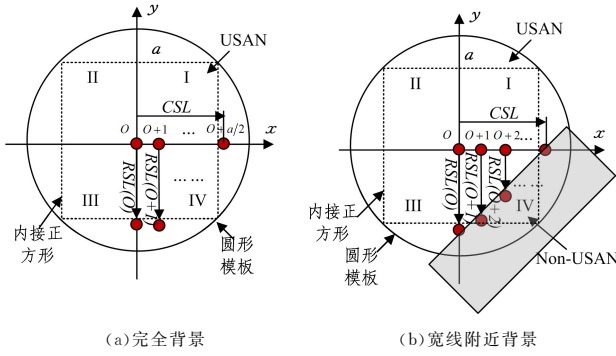


图 4 背景像素的自适应步长 CSL 和 RSL 的确定

Fig. 4 Determining of adaptive step length CSL and RSL for background pixels

1) 确定 CSL。在模板上沿当前中心像素  $O$  所在行找到 Non-USAN 区域内与 USAN 区域的交界处的像素(见图 3(b)、图 3(c)、图 4(b)),令其到中心像素  $O$  的水平距离为  $d_{cmin}$ ;若无交界(见图 3(a)、图 4(a)),则  $d_{cmin} = r$ 。若  $d_{cmin} > \sqrt{2}r/2$ ,则令  $CSL = \sqrt{2}r/2$ ;否则  $CSL = d_{cmin}$ 。

2) 确定 RSL。与 CSL 类似,不同的是其沿着列寻找 Non-USAN 区域内与 USAN 区域的交界处的像素(见图 3(a)、图 3(b)、图 4(b)),令其到中心像素的垂直距离为  $d_{Rmin}$ ;若无交界(见图 3(a)、图 4(a)),则  $d_{Rmin} = r$ 。若  $d_{Rmin} > \sqrt{2}r/2$ ,则令  $RSL = \sqrt{2}r/2$ ;否则  $RSL = d_{Rmin}$ 。

综上,CSL 和 RSL 可由下式确定:

$$\begin{cases} CSL = \min(d_{cmin}, \sqrt{2}r/2) \\ RSL = \min(d_{Rmin}, \sqrt{2}r/2) \end{cases} \quad (4)$$

假设图像的大小为  $M \times N$ ,AM-INF 宽线算子的实现过程如下:

步骤 1 初始化。定义  $RA[N]$  记录图像中一行像素的行自适应步长 RSL,初始化  $RA[j] = 1, j = 1, \dots, N$ ;定义  $RSL_{min}$  为一行像素的最小 RSL,初始化  $RSL_{min} = \sqrt{2}r/2$ ;定义矩阵  $BW[M][N]$  记录图像宽线检测结果,初始化为  $BW[i][j] = 0, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ 。

步骤 2 循环处理。从图像第一个像素开始,沿着自左向右、自上而下的顺序,根据步骤 3、步骤 4 移动模板,直至图像最后一个像素。

步骤 3 列自适应移动。假设模板放置在当前第  $i$  行的像素  $j$  上:1)若  $RA[j] > RSL_{min}$ ,则省略当前像素的 RC 计算,更新  $RA[j] = RA[j] - RSL_{min}$ ,模板右移 1 个像素;2)若  $RA[j] \leq RSL_{min}$ ,利用式(1)~式(3)计算 RC,求得像素  $j$  的类型及其 USAN 区域。若像素  $j$  为宽线特征,则标记  $BW[i +$

$p][j + q] = 1, (i + p, j + q) \in USAN \& 0 \leq p, q \leq r - 1$ 。利用式(4)确定像素  $j$  的列自适应步长 CSL,以及像素  $j$  和  $j + CSL - 1$  之间的像素的行自适应步长,并将其记录在 RA 中,  $RA[k] = RSL_k, k = j, j + 1, \dots, j + CSL - 1$ 。最后模板右移 CSL 个像素。

步骤 4 行自适应移动。当模板移动至第  $i$  行的最后一个像素时,搜索 RA 中的最小值,令  $RSL_{min} = \max(\min_{1 \leq j \leq N} RA[j], 1)$ ,然后把模板移动至  $i + RSL_{min}$  的第一个像素。

步骤 5 后处理。处理完所有像素后,进行形态学滤波、邻近连接等处理,从而得到最终的检测结果。

### 2.2 运算量分析

令  $M_{IV}^{USAN}$  为 USAN 区域位于模板象限 IV 内的像素个数,则 AM-INF 宽线算子相当于每  $M_{IV}^{USAN}$  个像素进行 1 次 RC 计算,即运算量约为基本 INF 宽线算子的  $1/M_{IV}^{USAN}$ 。但是,  $M_{IV}^{USAN}$  的取值取决于当前模板中心像素的类型和所处位置:位于宽线上时(见图 3),  $M_{IV}^{USAN} \in [1, \sqrt{2}rw/2]$ ;位于背景上且 USAN 区域包含模板的象限 IV 时(见图 4(a)),  $M_{IV}^{USAN} = r^2/2$ ;位于背景上且 USAN 区域不完全包含模板的象限 IV 时(见图 4(b)),  $M_{IV}^{USAN} \in (1, r^2/2)$ 。其中,  $w$  表示宽线的宽度。

由以上分析可知,相对于基本 INF 算子,AM-INF 宽线算子节省的运算量取决于图像内容。为了直观分析,假设  $K$  条宽线(长  $l$ 、宽  $w$ )随机分布在图像( $M \times N$ )中,且背景像素灰度起伏小于  $t$ 。考虑到 AM-INF 算子中模板的移动顺序,宽线附近的背景像素分布如图 5 所示,个数为  $r^2 + (l + w)r$ ,则图像中宽线像素总数量  $N_l = Klw$ ,宽线附近背景像素总数量  $N_{nl} = K[r^2 + (l + w)r]$ ,均匀背景像素个数  $N_b = M \times N - N_l - N_{nl}$ 。

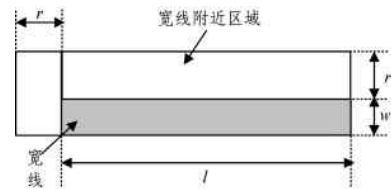


图 5 宽线附近区域示意图

Fig. 5 Image region near the wide lines

基于上述假设,AM-INF 算子与 INF 算子的运算量比较如表 1 所列。为简化计算,表 1 中  $M_{IV}^{USAN}$  不同位置的取值用其均值来近似:宽线像素  $M_{IV}^{USAN} = (1 + \sqrt{2}wr/2)/2$ ,宽线附近的背景像素  $M_{IV}^{USAN} = (r^2 + 2)/4$ 。另外,AM-INF 宽线算子增加了步长 CSL 和 RSL 的计算:在计算 RC 的每个像素位置时需计算 1 次 CSL 和若干次 RSL,其中 RSL 的计算次数取平均值来近似,即为  $(\sqrt{2}r + 2)/4$ 。

表 1 AM-INF 算子与基本 INF 算子的运算量分析

Table 1 Complexity comparison of AM-INF detector and INF detector

算子	宽线 (RC)	宽线附近背景区域 (RC)	均匀背景区域 (RC)	步长计算	总运算量
INF	$N_l = Klw$	$N_{nl}$	$N_b$	—	$O(MNr^2)$
AM-INF	$N_l' = \frac{2Kl}{1 + \sqrt{2}r/2}$	$N_{nl}' = \frac{N_{nl}}{(r^2 + 2)/4}$	$N_b' = \frac{N_b}{(r^2/2)}$	$k \cdot (\sqrt{2}r/4 + 1.5)$ $k = (N_l' + N_{nl}' + N_b')$	$O(k(r^2 + r))$
运算量之比	$\frac{1 + \sqrt{2}wr/2}{2} : 1$	$\frac{r^2 + 2}{4} : 1$	$\frac{r^2}{2} : 1$	—	$K/MN$

由表1可知,AM-INF算子的总运算量是 $k=N_l'+N_{nl}'+N_b'$ ( $k \leq M \times N$ )次RC计算和步长计算。显而易见, $k$ 取值越小,AM-INF算子节省的RC次数越多,运算量越小,相比基本INF算子的加速比也就越大。极限情况下,若图像中不存在任何宽线特征且图像背景的灰度起伏小于 $t$ ,则有 $k=(MN)/(r^2/2)$ 成立,AM-INF算子的加速性能最优,忽略步长计算时间,最大加速比可达 $r^2/2$ 。

### 3 实验分析

为了测试AM-INF算子的性能,采用两组实验进行分析。实验1分析AM-INF算子的理想最大加速比;实验2分析AM-INF算子的检测性能和速度性能。实验环境为:P4 3.0GHz CPU,4GB内存,Windows XP系统,Matlab 2014。

#### 3.1 实验1

实验1分析并验证AM-INF相对基本INF算子的最大加速比 $r_{acc}$ 和最大RC节约倍数 $r_{save}$ 。

如图6(a)所示,实验图像灰度为单一值,大小为 $256 \times 256$ ,两种算子使用相同的模板,半径 $r$ 分别为7,9,11,13,15,图6(b)和图6(c)分别给出了 $r=7,15$ 时AM-INF算子在图像中的模板移动位置示意图,与基本INF算子逐像素移动不同,AM-INF算子采用自适应移动的方式,省略了部分像素的冗余RC计算。

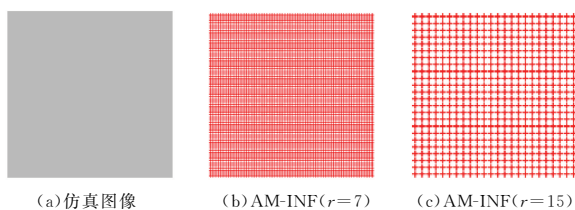


图6 AM-INF算子模板在图像中的采样位置

Fig. 6 Sampling location of AM-INF detector in picture

不同模板半径下,AM-INF算子相对INF算子的RC节约倍数 $r_{save}$ 和加速比 $r_{acc}$ 如图7所示。可以看出,在单一灰度图像中,模板半径 $r$ 越大,AM-INF算子相对基本INF算子节省的RC运算量越多,从而加速比也就越大。当 $r=15$ 时,AM-INF算子的RC最大节省倍数达到97,最大加速比 $r_{acc}$ 约为50倍。但是AM-INF算子的 $r_{acc}$ 小于 $r_{save}$ ,这是由算子的额外运算量导致的。

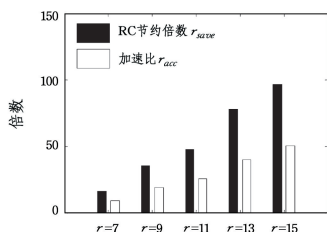


图7 两种算子在单一灰度图像上的性能比较

Fig. 7 Performance comparison of INF detector and AM-INF detector on single grayscale image

可见,对于单一灰度值图像,模板半径越大,每次移动的步长也就越大,节省的RC计算量就越大,AM-INF算子的加速性能也就越好,这与2.2节中关于运算量的理论分析一致。

#### 3.2 实验2

检测性能评价指标为<sup>[11]</sup>:

$$F = \frac{1}{\max(N_I, N_D)} \sum_{i=1}^{N_D} \frac{1}{1 + \xi d_i^2} \quad (5)$$

以基本INF算子得到的宽线中心线为基准,将AM-INF算子的检测结果与之比较。 $d_i$ 表示检测结果与基准结果对应像素点间的距离, $\xi \in (0, 1]$ 为惩罚系数,一般地, $\xi = 1/9$ , $N_I$ 和 $N_D$ 分别为基准结果和检测结果的像素个数。显然, $0 < F \leq 1$ ,其值越接近于1,检测性能越好。

评价加速性能的指标为RC的计算次数 $N_{RC}$ 和运算时间 $T_{run}$ ,同样以INF算子的实验结果为基准,计算AM-INF算子的RC节约倍数 $r_{save}$ 和加速比 $r_{acc}$ 。

##### 3.2.1 合成图像

合成图像共5幅,如图8所示。图8中宽线的宽度最大为6,因此两种INF算子的模板半径 $r=14$ <sup>[6-7]</sup>。

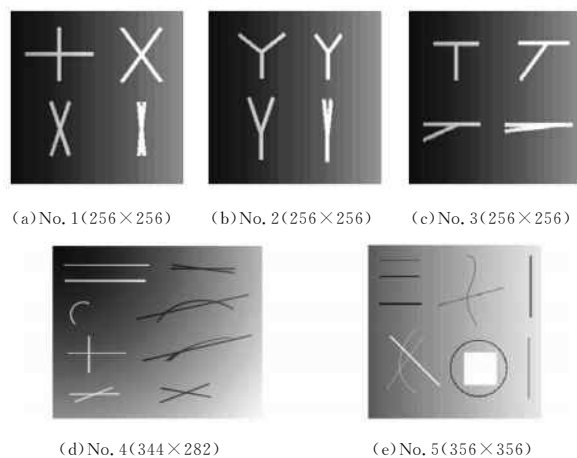
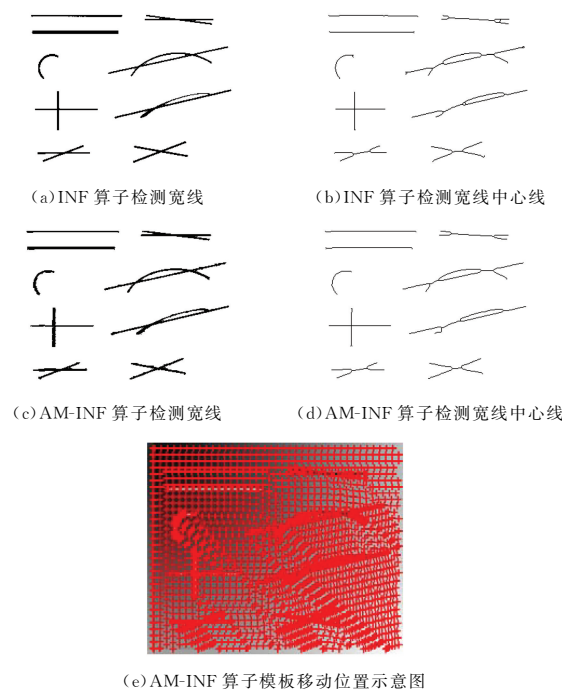


图8 合成图像

Fig. 8 Simulated images

图9给出了两种算子在第4幅图像上的检测结果示意。



(e) AM-INF算子模板移动位置示意图

图9 两种算子在图像No.4上的检测结果示意图

Fig. 9 Detection result of INF detector and AM-INF detector on simulated image No. 4

图 9(a)—图 9(d)分别为两种算子检测的宽线以及提取的宽线中心线,由图 9 可以看出,两种算子取得了类似的检测结果;图 9(e)为 AM-INF 算子在图像中的模板移动位置示意,由图可知 AM-INF 算子的模板在图像中的自适应移动情况,在均匀背景区域中,模板移动步长较大,以尽可能多地消除冗余 RC 计算,而在宽线特征附近的模板移动步长则较小,以确保不丢失宽线特征像素。

两种算子在 5 幅图像上的实验结果如图 10 所示。图 10(a)为算子的速度性能比较,图 10(b)为检测性能比较。可以看出,在仿真合成图像上,AM-INF 算子在保证检测性能与基本 INF 算子几乎相同的前提下,消除了基本 INF 算子的冗余 RC 计算,取得了较好的加速性能,RC 节约倍数  $r_{save}$  约为 40,加速比  $r_{acc}$  约为 20 倍。同时,比较两种快速算子在图像 No. 1—No. 5 上的性能可以发现,图像中均匀背景越多,AM-INF 算子的加速性能越好。其原因在于,当图像中均匀背景区域较多时,AM-INF 算子在移动模板时步长较大的次数也较多,节省的 RC 计算也就越多,从而加速性能就越好。

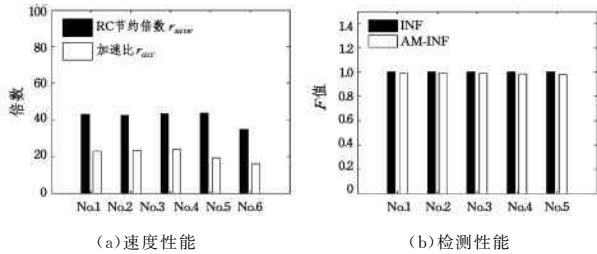


图 10 两种算子在合成图像上的性能比较

Fig. 10 Performance comparison of INF detector and AM-INF detector on simulated iamges

### 3.2.2 实际采集图像

实际采集图像共 6 幅,如图 11(a)—图 11(f)所示,根据宽线情况编号为 No. 1—No. 6。图中宽线的宽度最大为 6,两种 INF 算子的模板半径  $r=14$ <sup>[6-7]</sup>。

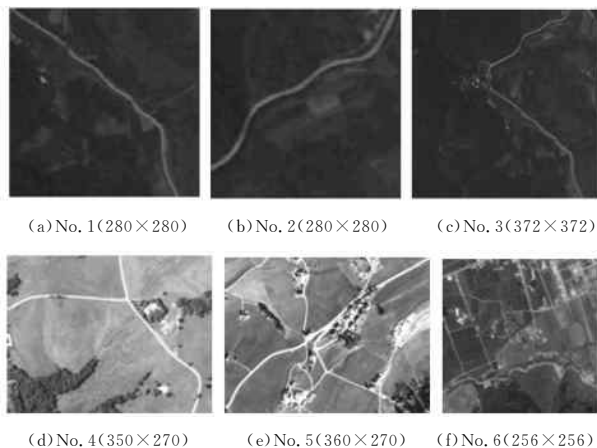


图 11 实际采集图像

Fig. 11 Real images

图 12 给出了两种算子在第 1 幅图像上的检测结果示意,图 12(a)—图 12(d)分别为两种算子检测的宽线以及提取的宽线中心线;图 12(e)给出了 AM-INF 算子在图像中的模板移动位置示意图。两种算子的实验结果如图 13 所示,

图 13(a)为算子的速度性能比较,图 13(b)为检测性能比较。

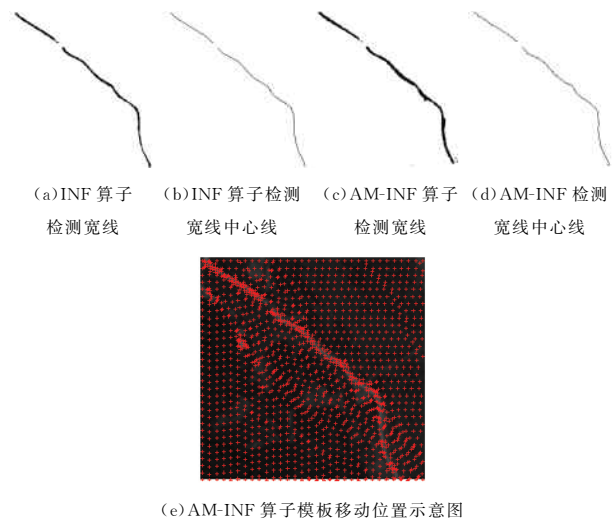


图 12 两种算子在实际图像 No. 1 上的检测结果示意

Fig. 12 Detection result of INF detector and AM-INF detector on real image No. 1

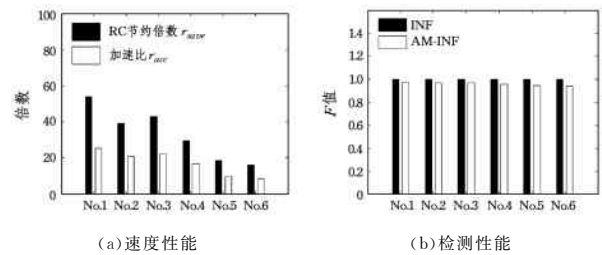


图 13 两种算子在实际采集图像上的性能比较

Fig. 13 Performance comparison of INF detector and AM-INF detector on real iamges

可以看到,在实际采集图像上取得了与仿真图像上相同的结论:与基本 INF 算子相比,AM-INF 算子同样表现出了较好的速度优势以及几乎相同的检测性能。

综上所述,AM-INF 算子根据图像当前像素提供的信息自适应地确定模板移动步长:在背景区域,加大步长以减少冗余计算;在宽线特征区域,缩短步长以保证检测性能。总之,通过自适应移动模板,AM-INF 算子较好地减少了基本 INF 算子中的冗余 RC 运算,在保证检测性能几乎不降低的同时,提高了运算速度。

**结束语** 通过设计基本 INF 宽线算子中圆形模板的移动方式,提出了一种宽线算子的快速实现方法——AM-INF 算子,较好地降低了基本 INF 算子的冗余 RC 运算量。实验分析表明,与基本 INF 宽线算子相比,本文提出的 AM-INF 算子取得了明显的速度优势,而且检测性能几乎不受影响。

### 参考文献

[1] LINDBERG T. Edge Detection and Ridge Detection with Automatic Scale Selection[J]. International Journal of Computer Vision, 1998, 30(2): 117-156.  
 [2] JACOB M, UNSER M. Design of Steerable Filters for Feature Detection Using Canny-Like Criteria[J]. IEEE Transactions on

- Pattern Analysis and Machine Intelligence, 2004, 26(8): 1007-1019.
- [3] EBERLY D, GARDNER R, MORSE B, et al. Ridges for Image Analysis[J]. Journal of Mathematical Imaging and Vision, 1994, 4(4): 353-373.
- [4] AGGARWAL N, KARL W C. Line Detection in Images through Regularized Hough Transform [J]. IEEE Transactions on Image Processing, 2006, 15(3): 582-591.
- [5] LI S X, CHANG H X, ZHU C F. Fast Curvilinear Structure Extraction and Delineation Using Density Estimation [J]. Computer Vision and Image Understanding, 2009, 113(6): 763-775.
- [6] LIU L, ZHANG D, YOU J. Detecting Wide Lines Using Isotropic Nonlinear Filtering[J]. IEEE Transactions on Image Processing, 2007, 16(6): 1584-1595.
- [7] LIU L, NGADI M O, PRASHER S O, et al. Objective determination of pork marbling scores using the wide line detector [J]. Journal of Food Engineering, 2012, 110(3): 497-504.
- [8] SMITH S M, BRADY J M. Susan—a New Approach to Low Level Image Processing [J]. International Journal of Computer Vision, 1997, 23(1): 45-78.
- [9] KNUTH D E, MORRIS J H, PRATT V R. Fast Pattern Matching in Strings[J]. SIAM Journal of Computing, 1977, 6(2): 323-350.
- [10] BOYER R S, MOORE J S. A fast string searching algorithm [J]. Communications of the ACM, 1977, 20(10): 762-772.
- [11] ABDOU I E, PRATT W K. Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors [J]. Proceedings of the IEEE, 1979, 67(5): 753-763.

(上接第 276 页)

### 参考文献

- [1] HAUTIERE N, TAREL J P, HALMAOUI H, et al. Enhanced fog detection and free-space segmentation for car navigation[J]. Machine Vision and Applications, 2014, 25(3): 667-679.
- [2] TANG Z X. Image Dehazing Based on Haziness Analysis[J]. International Journal of Automation and Computing, 2014, 11(1): 78-86.
- [3] OAKLEY J P, SATHERLEY B L. Improving image quality in poor visibility conditions using a physical model for contrast degradation[J]. IEEE Transactions on Image Processing, 1998, 7(2): 167-179.
- [4] TAN R. Visibility in bad weather from a single image [C]// Proceeding of IEEE Conference on Computer Vision and Pattern Recognition. Washington, DC: IEEE Computer Society, 2008: 2347-2354.
- [5] HE K, SUN J, TANG X. Single image haze removal using dark channel prior[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR 2009). IEEE, 2011: 1956-1963.
- [6] LEVIN A, LISCHINSKI D, WEISS Y. A Closed Form Solution to Natural Image Matting[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2006: 61-68.
- [7] HE K, SUN J, TANG X. Guided Image Filtering [M]// Computer Vision — ECCV 2010. Berlin: Springer Berlin Heidelberg, 2010: 1397-1409.
- [8] NARASIMHAN S G, NAYAR S K. Contrast Restoration of Weather Degraded Images[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2003, 25(6): 713-724.
- [9] ZHOU L, BI D Y, HE L Y. Single color image dehazing using variational partial differential equation [J]. Optics and Precision Engineering, 2015, 23(5): 1465-1472. (in Chinese)
- 周理, 毕笃彦, 何林远. 融合变分偏微分方程的单幅彩色图像去雾[J]. 光学精密工程, 2015, 23(5): 1465-1472.
- [10] SCHECHNER Y Y, NARASIMHAN S G, NAYAR S K. Instant dehazing of images using polarization[C]// Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001 (CVPR 2001). IEEE, 2001: 325-332.
- [11] SCHECHNER Y Y, NARASIMHAN S G, NAYAR S K. Polarization-based vision through haze [J]. Applied Optics, 2003, 42(3): 511-525.
- [12] YANG J, YIN W, ZHANG Y, et al. A fast algorithm for edge-preserving variation multichannel image restoration [J]. SIAM Journal on Imaging Science, 2009, 2(2): 569-592.
- [13] BRESSON X, CHAN T F. Fast minimization of the vectorial total variation norm and application to color image processing [J]. Inverse Problems and Imaging, 2008, 2(4): 455-484.
- [14] AUJOL J F, KANG S H. Color image decomposition and restoration [J]. Journal of Visual Communication and Image Representation, 2006, 17(4): 916-928.
- [15] DUVAL V, AUJOL J F, VESE L. A projected gradient algorithm for color image decomposition [J/OL]. <https://www.math.u-bordeaux.fr/~jaujol/PAPERS/CMLA2008-21.pdf>.
- [16] ZACH C, POCK T, BISCHOF H. A Duality Based Approach for Realtime TV-L1 Optical Flow[C]// Proceedings of DAGM'07. Berlin, Germany: Springer, 2007.
- [17] POCK T, URSCHLER M, ZACH C, et al. A Duality Based Algorithm for TV-L1-Optical-Flow Image Registration [M]// Medical Image Computing and Computer-Assisted Intervention—MICCAI 2007. Berlin: Springer Berlin Heidelberg, 2007: 511-518.