

基于模板分解和积分图像的快速 Kirsch 边缘检测

邵平^{1,2} 杨路明¹

摘要 将 Kirsch 算子的模板分解为差值模板和公共模板, 然后通过相邻差值模板的差异比较, 找出边缘强度最大的方向, 并计算出相应的边缘强度值, 避免了将 8 个方向的边缘强度全部算出, 减少了 Kirsch 算子的模板与原图像的卷积运算. 公共模板和原图像的卷积则利用灰度信息处理时得到的积分图像来加速. 实验证明应用这种快速算法的 Kirsch 边缘检测, 运算量比当前主流快速算法 (FKC 算法) 有较大幅度的减少. 另外, 运用模板分解和积分图像减少卷积运算的思路具有一定通用性, 实例说明此思路可用于一些其它边缘检测和空域滤波算法中.

关键词 边缘检测, Kirsch, 模板分解, 积分图像
中图分类号 TP391

Fast Kirsch Edge Detection Based on Templates Decomposition and Integral Image

SHAO Ping^{1,2} YANG Lu-Ming¹

Abstract Templates of Kirsch operators are decomposed into difference templates and a common template. By a contrast between every two-neighbor difference templates, the direction of maximum edge intensity is found and the corresponding value of edge intensity is worked out. Thus it is no longer necessary to compute the edge intensity in eight directions, greatly reducing the convolution between templates of Kirsch operators and original image, and at the same time accelerating the convolution between the common template and original image by integral image that has been worked out in gray information processing. Using such a fast algorithm, Kirsch edge detection is made much less time-consuming than that of the current mainstream fast algorithm (FKC algorithm). The validity is proved by experiments. The idea of reducing convolution with templates decomposition and integral image has some universality. Examples show that this idea can also be applied in other edge detection algorithms and space filters.

Key words Edge detection, Kirsch, templates decomposition, integral image

1 引言

图像的边缘特征在图像处理和目标识别领域有着重要的用途. 而 Kirsch 算子在图像边缘检测中也有着广泛的应用^[1~5]. Kirsch 算子由 8 个 3×3 的模板 (卷积核) 构成, 每两个模板之间的夹角为 45 度^[6,7]. 由于它采用 8 个模板来处理一幅图像的每一个像素, 运算量大, 且图像处理和目标识别任务也越来越复杂, 因此其应用范围在一定程度上受到限制. 例如在人脸检测或识别系统的训练过程中, 大量图像或图像窗口需要处理, 运用 Kirsch 算子提取图像或图像窗口的边缘特征所花费的时间不可忽视.

目前, 利用 Kirsch 算子进行边缘检测的快速算

法较少^[8,9]. 文献 [9] 提出的 FKC 算法, 有效地解决了经典算法运算量过大的问题, 是当前的主流快速算法. 它采用移行推算的办法来计算 8 个方向的边缘强度, 然后选取最大值作为最终边缘强度. 移行推算先用一个模板与图像进行卷积, 算出与该模板方向相应的边缘强度, 其余 7 个方向均采用前一方向边缘强度的计算结果来推算下一相邻方向的边缘强度.

积分图像是一种用于快速计算图像窗口的灰度值总和或灰度均值的一种图像中间表示, 有较为广泛的应用^[10~12]. 除此之外, 积分图像还能用于快速计算图像窗口的灰度方差, 以及实现图像预处理中的快速灰度分布标准化和目标任务识别中的快速模板匹配^[13,14], 因此在图像灰度信息处理时往往先计算积分图像, 以提高计算效率.

本文提出一种基于模板分解和积分图像的快速 Kirsch 边缘检测算法 (Templates decomposition and integral image based Kirsch algorithm, TDI-IKA), 该算法将 Kirsch 算子的 8 个方向模板分解为差值模板和公共模板, 然后对各差值模板进行比较, 找出边缘强度的最大方向, 并计算出相应的边缘强度值, 避免了将各方向模板作用于图像后的边缘强

收稿日期 2006-6-26 收修改稿日期 2007-2-12
Received June 26, 2006; in revised form February 12, 2007
广西教育厅科研项目 (200508208), 玉林师范学院重点科研项目 (2006YJZD03) 资助
Supported by Research Project of Department of Education of Guangxi (200508208), Key Research Project of Yulin Normal College (2006YJZD03)
1. 中南大学信息科学与工程学院 长沙 410000 2. 玉林师范学院物理与信息科学系 玉林 537000
1. Department of Information Science and Engineering, Central South University, Changsha 410000 2. Department of Physics and Information Science, Yulin Normal College, Yulin 537000
DOI: 10.1360/aas-007-0795

度全部算出,大大减少了模板与图像的卷积运算.公共模板和图像的卷积则利用图像灰度信息处理时得到的积分图像来加速.理论分析表明,如果积分图像已知(在图像灰度信息处理过程中积分图像已算出),TDIIKA 算法的加法次数仅为 FKC 算法的 62%,乘法次数仅为 FKC 算法的 67%,仿真实验结果也证明,TDIIKA 算法的检测时间仅为 FKC 算法的 50%~70%.因此,结合图像灰度和梯度信息进行目标检测时,TDIIKA 算法能有效地提高 Kirsch 边缘检测速度.并且,本文提出的运用模板分解和积分图像减少卷积运算的思路具有一定通用性,可应用于其它边缘检测和空域滤波算法中.

本文组织如下:第 2 节简要介绍 Kirsch 算子和积分图像;第 3 节阐述 TDIIKA 算法及其运算量,并讨论积分图像在 TDIIKA 算法中的作用;第 4 节给出一个综合应用实例;第 5 节举例说明运用模板分解和积分图像减少卷积运算思路的通用性;第 6 节为实验结果,第 7 节是总结.

2 Kirsch 算子和积分图像

2.1 Kirsch 算子

设图 1 所示的 $M_k(k=0,1,2,\dots,7)$ 为 Kirsch 算子的 8 个 3×3 模板,各 M_k 均可看成是 3×3 的矩阵,每个模板对应的边缘方向如图 2 所示.将 M_k 分别与图像中每一个可能的 3×3 窗口进行卷积,然后取最大输出值作为相应窗口中央像素点的边缘强度,并把取得最大值的模板对应的 k 作为边缘方向.

M_0	M_1	M_2	M_3
$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$
M_4	M_5	M_6	M_7
$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$

图 1 Kirsch 算子在 8 个方向上的模板

Fig. 1 Templates of Kirsch operators in eight directions

假设图像中任意一点坐标为 (i, j) ,与其邻域构成的 3×3 窗口 P 如图 3 所示, P 也可看成是 3×3 的矩阵,其中 $p(i+n, j+n)$ ($n=-1, 0, 1$) 表示相应点的灰度值.设“*”表示卷积运算,并令 q_k 为点 (i, j) 经过 Kirsch 算子的 M_k 模板作用后的边缘强度,即

$$q_k = M_k * P \quad (1)$$

则点 (i, j) 的最终边缘强度 $q(i, j) = \{\max(q_k) \mid k=0, 1, 2, \dots, 7\}$,相应的边缘方向 $d(i, j) = \{k \mid \max(q_k)\}$,其中 \max 表示取最大值(以下同).

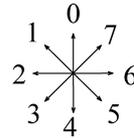


图 2 边缘方向 k

Fig. 2 Edge direction k

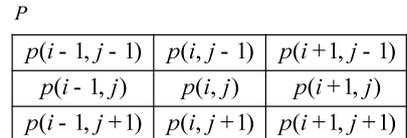


图 3 图像的 3×3 窗口示意图

Fig. 3 Sketch map of 3×3 window in an image

2.2 积分图像的概念

积分图像(Integral image)^[11]中任意一点 (i, j) 的值 $ii(i, j)$ 表示图 4 所示的原图像相应斜线区域的灰度值总和,即

$$ii(i, j) = \sum_{i' \leq i, j' \leq j} p(i', j') \quad (2)$$

其中 $p(i', j')$ 表示原图像中一点 (i', j') 的灰度值. $ii(i, j)$ 可用式 (3) 和 (4) 迭代计算得到

$$s(i, j) = s(i, j-1) + p(i, j) \quad (3)$$

$$ii(i, j) = ii(i-1, j) + s(i, j) \quad (4)$$

其中 $s(i, j)$ 表示一列的积分,且 $s(i, -1) = 0, ii(-1, j) = 0$.求积分图像,只需扫描原图像所有像素一遍,计算量很小.图 5 所示的窗口 W 的灰度值总和,不管窗口 W 的大小如何,均可以用积分图像的 4 个相应点 $(i_1, j_1), (i_2, j_2), (i_3, j_3), (i_4, j_4)$ 快速计算出来,即窗口 W 的灰度值总和为: $ii(i_4, j_4) + ii(i_1, j_1) - (ii(i_2, j_2) + ii(i_3, j_3))$.

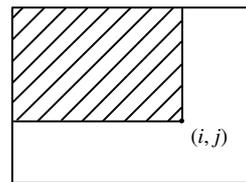


图 4 积分图像示意

Fig. 4 Sketch map of integral image

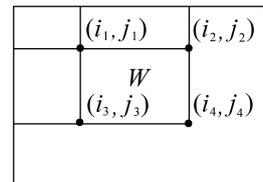


图 5 积分图像用法示意

Fig. 5 Sketch map of using integral image

3 TDIIKA 算法及其运算量分析

3.1 TDIIKA 算法原理

处理一幅 $N \times N$ 的图像, Kirsch 经典算法所需的加法运算次数为 $56N^2$ 次,乘法运算次数为 $16N^2$ 次; FKC 算法所需加法运算次数为 $21N^2$ 次,乘法运算次数为 $3N^2$ 次^[9],它们获取边缘强度最大值所需的数值比较次数均为 $7N^2$ 次.

以上两种算法的共同思想是,对图像中的每一点,均计算出 Kirsch 算子 8 个模板作用后的边缘强度,然后通过比较取最大值输出.本文提出的 TDIIKA 算法,不需将 Kirsch 算子 8 个模板作用后的边

缘强度全部算出, 仅需比较各模板作用于图像的边缘强度差异, 然后根据差异找出其中边缘强度最大的方向, 并计算相应的边缘强度值.

图 6 为将原模板进行分解得到的 8 个差值模板 $M_{dk}(k = 0, 1, 2, \dots, 7)$ 和公共模板 M_C (注: M'_C 是公共模板的另一种表示方法, 将在 3.3 节讨论). 它们仍可看成 3×3 的矩阵, 且具有每一个元素均为 1 或 0 的特点. 它们与图像进行卷积运算, 实际只需对图像中相关点的灰度值求和, 避免了乘法运算.

M_{d0}	M_{d1}	M_{d2}	M_{d3}	M_{d4}
1 1 1 0 0 0 0 0 0	1 1 0 1 0 0 0 0 0	1 0 0 1 0 0 1 0 0	0 0 0 1 0 0 1 1 0	0 0 0 0 0 0 1 1 1
M_{d5}	M_{d6}	M_{d7}	M_C	M'_C
0 0 0 0 0 1 0 1 1	0 0 1 0 0 1 0 0 1	0 1 1 0 0 1 0 0 0	1 1 1 1 1 1 1 1 1	1 1 1 1 0 1 1 1 1

图 6 Kirsch 的差值模板和公共模板

Fig. 6 The difference and common templates of Kirsch

令 $q_{dk} = M_{dk} * P$, 式 (1) 可分解为

$$q_k = 8q_{dk} - 3(M_C * P - p(i, j)) \quad (5)$$

图 7 以模板 M_1 的分解为例说明了式 (5) 的原理.

5	5	-3	= 8 ×	1	1	0	- 3 ×	(1	1	1	-	0	0	0)	
5	0	-3		1	0	0			1	1	1		0	1	0		0
-3	-3	-3		0	0	0			0	0	0		0	0	0		0
(Original template)				(Difference part)				(Common part)									

图 7 Kirsch 模板分解示例

Fig. 7 An example of templates decomposition of Kirsch

对每一个模板来说, 式 (5) 等号右边只有 q_{dk} 不同, 其中 $p(i, j)$ 为图像当前点的灰度值, 而 $M_C * P$ 为图像当前点的 3×3 邻域灰度值总和, 可以根据式 (6) 用前述积分图像快速获得.

$$M_C * P = ii(i + 1, j + 1) + ii(i - 2, j - 2) - (ii(i + 1, j - 2) + ii(i - 2, j + 1)) \quad (6)$$

由于计算图像中任意一点在 8 个方向的 q_k 并不是 Kirsch 算子边缘检测的目标, 真正的目标是寻找并计算 $\max(q_k)$. 根据式 (5), 只需找出 $\max(q_{dk})$, 然后再计算其相应的 q_k , 即得 $\max(q_k)$. 为减少寻找 $\max(q_{dk})$ 的运算量, 可以先比较任意两个相邻方

向 q_{dk} 的差异, 然后仅计算其中较大者. 例如, 因 $q_{d0} = M_{d0} * P$, 而 $q_{d1} = M_{d1} * P$, 两者比较时, 仅需根据 M_{d0} 和 M_{d1} 的微小差别进行如下比较:

若 $p(i + 1, j - 1) > p(i - 1, j)$, 求 q_{d0} ; 否则求 q_{d1} . 类似地,

若 $p(i - 1, j - 1) > p(i, j + 1)$, 求 q_{d2} ; 否则求 q_{d3} ;

若 $p(i - 1, j + 1) > p(i + 1, j)$, 求 q_{d4} ; 否则求 q_{d5} ;

若 $p(i + 1, j + 1) > p(i, j - 1)$, 求 q_{d6} ; 否则求 q_{d7} .

这样, 只需求 8 个 q_{dk} 中的 4 个, 然后从这 4 个已求出的 q_{dk} 中找出 $\max(q_{dk})$ 即可. 最后再根据式 (5) 计算其对应的 q_k , 即为 $\max(q_k)$. 进行比较时, 只要是两个相邻方向的 q_{dk} 就可以比较, 例如取 q_{d1} 和 q_{d2} 进行比较.

TDIKA 算法的详细步骤为:

1) 计算原图像的积分图像 (如果图像灰度信息处理过程中已算出积分图像, 则省略此步);

2) 扫描原图像中的每一点 (i, j) , 利用积分图像及式 (6) 计算 $M_C * P$;

3) 比较任意两个相邻方向 q_{dk} 的差异, 得到 8 个 q_{dk} 中较大的 4 个, 计算出这 4 个 q_{dk} , 并计算 $\max(q_{dk})$;

4) 根据式 (5) 计算 $\max(q_{dk})$ 对应的 q_k , 即 $\max(q_k)$;

5) 原图像扫描未结束则转 2), 否则程序终止.

3.2 TDIKA 算法的运算量分析

根据 TDIKA 算法的原理, 对任意一幅 $N \times N$ 的图像, 首先利用式 (3) 和 (4) 迭代计算积分图像, 运算量为 $2N^2$ 次加法; 然后对图像中每一点计算其边缘强度的最大值和方向, 每一点的运算量为:

1) 利用积分图像和式 (6) 计算 $M_C * P$, 需 3 次加法;

2) 通过 4 次数值比较, 找出 4 个较大的 q_{dk} 并计算. 计算每个 $q_{dk} = M_{dk} * P$ 为 3 个像素的灰度值求和, 仅需 2 次加法, 因此共需 8 次加法;

3) 再进行 3 次数值比较, 找出 $\max(q_{dk})$, 然后根据式 (5) 计算 $\max(q_{dk})$ 对应的 q_k , 得到 $\max(q_k)$, 需 2 次加法和 2 次乘法.

这表明, 对一幅 $N \times N$ 的图像, TDIKA 算法进行数值比较的运算量为 $4N^2 + 3N^2 = 7N^2$ 次; 与经典算法和 FKC 算法相同, 总算术运算量为 $2N^2 + (3 + 8 + 2)N^2 = 15N^2$ 次加法和 $2N^2$ 次乘法.

表 1 (见下页) 对经典算法、FKC 算法和 TDIKA 算法进行了综合比较, 可以看出 TDIKA 算法的加法次数仅为 FKC 算法的 71% (已知积分图像时仅 62%), 乘法次数仅为 FKC 算法的 67%.

表 1 各算法运算量和特点比较

Table 1 Comparison of arithmetic operations and characteristics among algorithms

算法	加法次数 (N^2)	乘法次数 (N^2)	数值比较次数 (N^2)	是否完全整数运算	减少运算量的原理	其它特点
经典	56	16	7	是	系数相同各项组合	-
FKC	21	3	7	否	移行推算	-
TDIKA	15	2	7	是	模板分解 + 积分图像	积分图像可再利用
TDIKA	13*	2	7	是	模板分解 + 积分图像	积分图像可再利用
TDKA1	16	2	7	是	模板分解	-
TDKA2	13	2	8~15	是	模板分解	-

注: 带 * 的数字为已知积分图像时的加法次数.

3.3 积分图像在 TDIKA 算法中的作用讨论

为了分析积分图像在 TDIKA 算法中的贡献大小, 提出另外两种只依赖模板分解, 无需积分图像的算法 (Templates decomposition based Kirsch algorithm, TDKA) 作为比较, 分别称之为 TDKA1 算法和 TDKA2 算法.

3.3.1 TDKA1 算法

TDKA1 算法改设公共模板为 M'_C (见图 6), 将式 (1) 表示为

$$q_k = 8q_{dk} - 3(M'_C * P) \quad (7)$$

仍以模板 M_1 为例, 此时图 7 的模板分解形式变为

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} = 8 \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - 3 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

TDKA1 算法与 TDIKA 算法的不同之处在于: TDKA1 算法用式 (7) 代替了式 (5). 式 (7) 的 $M'_C * P$ 项为直接卷积运算, 无需积分图像. TDKA1 算法的总运算量见表 1, 分析略.

3.3.2 TDKA2 算法

将图 6 中的差值模板 $M_{dk}(k = 0, 1, 2, \dots, 7)$ 除中央点保持为 0 外, 其余位置的 1 均改为 0, 而 0 均改为 1, 将可得到 8 个对应的模板, 记为 $M'_{dk}(k = 0, 1, 2, \dots, 7)$. 则式 (1) 还可表示为

$$q_k = 5q_{dk} - 3(M'_{dk} * P) \quad (8)$$

仍以模板 M_1 为例, 此时图 7 的模板分解形式变为

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} = 5 \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - 3 \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

TDKA2 算法与 TDIKA 算法不同之处仅在于: TDKA2 算法根据式 (8) 来计算 $\max(q_{dk})$ 对应的 q_k , 而不是根据式 (5). 值得说明的是, TDKA2 算法在进行模板差异比较时仍使用式 (5) 的模板分解形式. 式 (8) 的 $M'_{dk} * P$ 项为直接卷积运算, 无

需积分图像, 但其中的 $M'_{dk}(k = 0, 1, 2, \dots, 7)$ 需要根据对应的边缘方向 k 来选择, 即需作如下数值比较:

若 $k = 0$, 选择 M'_{d0} ; 否则, 若 $k = 1$, 选择 M'_{d1} ; ...

否则, 若 $k = 6$, 选择 M'_{d6} ; 否则, 选择 M'_{d7} .

这里的数值比较次数为 $(1 \sim 8)N^2$, 取决于图像每一点最大边缘强度的方向 k . 对图像中任意一点, $k = 7$ 时的比较次数最多, 因此, 对于不同的图像, 此数值比较运算量会有所差异. TDKA2 算法的总运算量见表 1, 分析略.

由表 1 可知, TDIKA 算法在已知积分图像的情况下, 比 TDKA1 算法少 $3N^2$ 次加法; 比 TDKA2 算法的数值比较次数要少. 因此, 若实际应用系统在图像灰度信息处理过程中已算出积分图像, 选用 TDIKA 算法的效率较高.

4 综合应用实例

以文献 [12] 中结合灰度图像、梯度图像和二值图像分别运用积分图像进行快速人脸检测的方案为例, 图 8 是原理框图. 灰度图的积分图像原来是用于后续的灰度信息分块统计的, 应用 TDIKA 算法可实现灰度图的积分图像二次利用. 如图 8 中虚线所示, 灰度图的积分图像可被再次用于梯度图像的提取. 或者说, 如果先应用 TDIKA 算法提取梯度图像, 则提取梯度图像过程中已算出的灰度图的积分图像, 可在灰度图像的分块统计中再次利用.

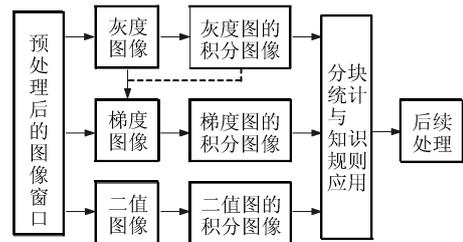


图 8 综合应用实例原理框图

Fig. 8 Frame map of an example's principle in integrative application

5 运用模板分解和积分图像减少卷积运算量思路的通用性

上述 Kirsch 边缘检测算法运用模板分解和比较模板之间差异的方法, 显著减少了多模板与图像进行卷积的运算量; 积分图像在算法中也有一定的辅助作用, 但并不十分明显. 不过, 这并不能否认积分图像具有减少卷积运算量的性能. 事实上, 在模板尺度较大的情况下 (如 5×5 , 7×7 模板), 结合积分图像能显著减少一些边缘检测算子或空域滤波模板与图像的卷积运算量. 运用模板分解和积分图像减少卷积运算量思路的通用性, 可以用下面的几个例子来说明.

5.1 提高 8 方向 Prewitt 算子的边缘检测效率

与 Kirsch 算子的模板分解形式类似, 可设 8 方向 Prewitt 算子^[15,16] 的 8 个差值模板为 $T_{dk}(k = 0, 1, 2, \dots, 7)$, 公共模板为 T_C (或 T'_C), 如图 9 所示.

T_{d0}	T_{d1}	T_{d2}	T_{d3}	T_{d4}
1 0 0 1 1 0 1 0 0	0 0 0 1 1 0 1 1 0	0 0 0 0 1 0 1 1 1	0 0 0 0 1 1 0 1 1	0 0 1 0 1 1 0 0 1
T_{d5}	T_{d6}	T_{d7}	T_C	T'_C
0 1 1 0 1 1 0 0 0	1 1 1 0 1 0 0 0 0	1 1 0 1 1 0 0 0 0	1 1 1 1 1 1 1 1 1	1 1 1 1 0 1 1 1 1

图 9 Prewitt 的差值模板和公共模板

Fig. 9 The difference and common templates of Prewitt

令 $q_{dk} = T_{dk} * P$, 则对 Prewitt 边缘检测来说, 式 (5) 可表示为 $q_k = -2q_{dk} + (T_C * P - p(i, j))$, 式 (7) 可表示为 $q_k = -2q_{dk} + T'_C * P$.

由于任意两相邻的 T_{dk} 只有一个 1 的位置不相同, 即具有与前述 Kirsch 差值模板完全相同的特点, 因此也可采用模板差异比较的方法来减少多模板与图像的卷积运算量. 实现快速 8 方向 Prewitt 边缘检测类似前述 TDIKA 算法和 TDKA1 算法的原理, 甚至运用 16 方向 Prewitt 算子^[16] 进行边缘检测仍可依此模板分解和积分图像的思路来提高计算效率, 具体算法在此不再赘述.

5.2 减少空域滤波模板与图像的卷积运算量

常见的一些空域滤波模板也可结合模板分解和积分图像降低卷积运算量. 以尺度较大的 5×5 高通滤波模板^[15] 为例, 按如下方式分解

$$\frac{1}{25} \times \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} - \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

其中, 分解得到的全为 1 的子模板与图像的卷积即 5×5 邻域内各像素的灰度值之和, 可用积分图像快速算出, 从而使卷积运算量大为减少, 积分图像此时的贡献十分明显. 若是 7×7 的滤波模板^[15], 这种结合模板分解和积分图像的算法效率将更高.

5.3 减少 LOG 算子与图像的卷积运算量

LOG 算子进行边缘检测的模板^[7] 相对较难分解, 若按下面方式分解

$$\begin{bmatrix} -2 & -4 & -4 & -4 & -2 \\ -4 & 0 & 8 & 0 & -4 \\ -4 & 8 & 24 & 8 & -4 \\ -4 & 0 & 8 & 0 & -4 \\ -2 & -4 & -4 & -4 & -2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 2 \\ 0 & 4 & 12 & 4 & 0 \\ 0 & 12 & 28 & 12 & 0 \\ 0 & 4 & 12 & 4 & 0 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix} - 4 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

则其卷积运算量仍然可以得到一定程度的减少. 其中全为 1 的子模板与图像的卷积利用积分图像来计算.

6 实验结果与分析

在 CPU 为 PIII700、内存 256M 的微机完成实验, 用 delphi 7.0 编程, 实验标准图像来自 <http://sipi.usc.edu/database/database.cgi?volume=misc>. 表 2 (见下页) 是对多幅标准图像进行检测取平均的结果, 经典算法与 FKCA 算法检测时间的对比参照文献 [9]. 表 2 的数据说明: 1) 从检测速度看, 本文的 TDIKA 算法、TDKA1 算法和 TDKA2 算法均比 FKCA 算法要快; 2) 实际检测时间并不完全取决于理论上各算法的算术运算量, 因为 FKCA 算法包含非整数运算, 而本文的 TDIKA 算法则在计算积分图像时需专门扫描原图像一次, 这些都会影响实际检测时间; 3) 若已知积分图像 (即在图像灰度信息处理过程中已算出积分图像), 选用 TDIKA 算法效率最高, 仅为 FKCA 算法的 50%~70% 左右.

7 结论

鉴于利用 8 方向 Kirsch 算子进行边缘检测的运算量很大, 其快速算法较少, 降低其运算量对其推广应用具有实际意义. 本文提出了一种基于模板分解和积分图像的 Kirsch 边缘检测快速算法 (TDIKA 算法). 与当前主流快速算法 (FKCA 算法) 相比, TDIKA 算法的加法次数仅为 71% (已知积分图像时仅 62%), 乘法次数仅为 67%. 仿真实验也表明, 已知积分图像时, TDIKA 算法的检测时间仅为

FKC 算法的 50%~70% 左右。另外, 本文提出的运用模板分解和积分图像减少卷积运算的思路具有一定通用性, 可在一些其它边缘检测和空域滤波算法中应用。

表 2 各算法的检测时间比较

Table 2 Comparison of detection time among algorithms

算法	图像大小 (pixels)	检测时间 1	检测时间 2
FKC	256 × 256	0.02	0.02
TDIIKA	256 × 256	0.02	0.01
TDKA1	256 × 256	0.02	0.02
TDKA2	256 × 256	0.02	0.02
FKC	512 × 512	0.09	0.09
TDIIKA	512 × 512	0.08	0.06
TDKA1	512 × 512	0.08	0.08
TDKA2	512 × 512	0.08	0.08
FKC	1024 × 1024	0.37	0.37
TDIIKA	1024 × 1024	0.32	0.26
TDKA1	1024 × 1024	0.33	0.33
TDKA2	1024 × 1024	0.32	0.32

注: 检测时间 1 含求积分图像, 检测时间 2 为已知积分图像。

References

- Lee S W. Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996, **18**(6): 648~652
- Li H, Chutatape O. Fundus image features extraction. In: Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2000. 3071~3073
- Banumathi A, Devi R K, Kumar V A. Performance analysis of matched filter techniques for automated detection of blood vessels in retinal images. In: Proceedings of the Technology Annual Conference on Convergent Technologies for Asia-Pacific Region. IEEE, 2003. 543~546
- Yu Ye, Lu Jian-Hua, Zheng Jun-Li. Color image edge detection algorithm. *Journal of Tsinghua University*, 2005, **45**(10): 1339~1343
(于焯, 陆建华, 郑君里. 一种新的彩色图像边缘检测算法. 清华大学学报, 2005, **45**(10): 1339~1343)
- Bai Zheng-Yao, He Pei-Kun, Liu Zhou-Feng. A method of SAR speckle suppression based on Kirsch direction template. *Systems Engineering and Electronics*, 2004, **26**(7): 881~882
(柏正尧, 何佩琨, 刘洲峰. 基于 Kirsch 方向模板的 SAR 相干斑噪声抑制方法. 系统工程与电子技术, 2004, **26**(7): 881~882)
- Kirsch R. Computer determination of the constituent structure of biological images. *Computer in Biomedical Research*, 1971, **4**(3): 315~328
- Lang Rui. *Science of Digital Image Processing: Realizing of Visual C++*. Beijing: Publishing House of Hope Industry, 2003. 256~257
(郎锐. 数字图像处理学 - Visual C++ 实现. 北京: 希望电子出版社, 2003. 256~257)
- Lu Li, Li Qing, Wang Neng-Chao. Parallel algorithm of computing Kirsch operator using PVM. *Signal Processing*, 1997, **13**(4): 363~368

(卢力, 李青, 王能超. 并行 Kirsch 算子计算在 PVM 环境中的实现. 信息处理, 1997, **13**(4): 363~368)

- Zheng Xiang, Huang Yi-Yun. Fast Kirsch edge detection arithmetic. *Journal of China Institute of Communications*, 1996, **17**(1): 131~134
(郑翔, 黄艺云. Kirsch 边缘检测算子的快速算法. 通信学报, 1996, **17**(1): 131~134)
- Crow F. Summed-area tables for texture mapping. *ACM Computer Graphics*, 1984, **18**(3): 207~212
- Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. IEEE, 2001. 511~518
- Zhang Min-Gui. A Study of Face Recognition Methods Based on Wavelet and SVM [Ph. D. dissertation], Northwestern Polytechnical University, 2003
(张敏贵. 基于小波和支持向量机的人脸识别方法研究 [博士学位论文]. 西北工业大学, 2003)
- Shao Ping, Yang Lu-Ming, Huang Hai-Bin, Zeng Yao-Rong. Rapid algorithm of template matching for face detection. *Journal of Computer Applications*, 2006, **26**(7): 1607~1610
(邵平, 杨路明, 黄海滨, 曾耀荣. 适于人脸检测的模板匹配快速算法. 计算机应用, 2006, **26**(7): 1607~1610)
- Shao Ping, Yang Lu-Ming, Huang Hai-Bin, Zeng Yao-Rong. Rapid algorithm of template matching based on integral image. *Computer Science*, 2006, **33**(12): 225~229
(邵平, 杨路明, 黄海滨, 曾耀荣. 基于积分图像的快速模板匹配. 计算机科学, 2006, **33**(12): 225~229)
- Schowengerdt R A. *Techniques for Image Processing and Classification in Remote Sensing*. Beijing: Science Press, 1991. 55~71
(肖温格 R A. 遥感中的图像处理 and 分类技术. 北京: 科学出版社, 1991. 55~71)
- Qin Xuan-Yun, Bu Ying-Yong. Construction of multidirection template and fast algorithm for template matching. *Computer Engineering and Design*, 2003, **24**(6): 91~96
(秦宣云, 卜英勇. 多方向模板的构造及匹配的快速算法. 计算机工程与设计, 2003, **24**(6): 91~96)



邵平 副教授, 中南大学信息科学与工程学院博士研究生. 主要研究方向为图像处理和模式识别. 本文通讯作者.

E-mail: shao551@126.com

(SHAO Ping Associate professor, Ph.D. candidate at Department of Information Science and Engineering, Central South University. His research

interest covers image processing and pattern recognition. Corresponding author of this paper.)



杨路明 中南大学信息科学与工程学院教授. 主要研究方向为数据库, 图像处理和模式识别.

E-mail: yangluming@csu.edu.cn

(YANG Lu-Ming Professor at Department of Information Science and Engineering, Central South University. His research interest covers database,

image processing, and pattern recognition.)