

# 基于随机匹配的非局部相似块搜索算法<sup>①</sup>

余文森, 吴 薇

(武夷学院 数学与计算机学院, 武夷山 354300)

**摘 要:** 针对非局部相似块搜索问题, 提出一个基于随机匹配的  $k$  近邻块匹配算法. 在基于 Jump Flooding 传播的块匹配算法基础上, 改进其候选参考块的产生方式, 增加从查询块的局部邻域中随机产生候选参考块这一方式. 这一改进提高了候选参考块匹配的可能性, 进而提高了算法的匹配精确度. 实验结果表明改进算法在时间效率和并行性上, 与原算法相差不大, 但在匹配精确度上, 要优于原算法.

**关键词:** 随机匹配; 非局部相似块搜索;  $k$  近邻块匹配; 非局部自相似性; 局部自相似性

## Non-Local Similar Patch Search Algorithm Based on Random Match

YU Wen-Sen, WU Wei

(College of Mathematics and Computer Science, Wuyi University, Wuyishan 354300, China)

**Abstract:** A  $k$ -nearest neighbor patch match algorithm based on random match was proposed to solve the non-local similar patch search problem, which is an improvement of the patch match algorithm based on jump flooding. On the basis of the origin algorithm, the improved algorithm proposed an additional way to randomly generate candidate reference patch from the local neighborhood of each query patch, which raises the possibility of matching candidate patch to query patch and improves the matching accuracy. Experimental results show that the improved algorithm is comparable with the origin algorithm in time efficiency and parallelism, and outperforms the origin algorithm in matching accuracy.

**Key words:** random match; non-local similar patch search;  $k$ -nearest neighbor patch match; non-local self-similarity; local self-similarity

非局部自相似性是指一个图像的局部结构模式可能在整个图像中重复自己, 即一个局部图像块, 在整个图像范围内可能存在多个相似块与之对应. 近年来, 图像的这一属性被许多图像处理任务所利用, 如图像去噪<sup>[1]</sup>, 图像修复<sup>[2]</sup>, 图像超分辨率重建<sup>[3]</sup>等. 利用这一属性一般要包含一个非局部相似块的搜索过程, 即在整个图像范围内搜索每个局部块的相似块. 非局部相似块搜索本质上是一个  $k$  近邻搜索问题. 在这一搜索问题中, 参考块集合和查询块集合是一样的, 都是由同一幅图像中所有可能的局部块组成. 相似块搜索就是从参考块集合中搜索距离查询集合中各查询块最

近的  $k$  个块. 然而这一搜索过程极度耗时, 难以应用于实际的图像处理任务. 在实际应用中, 许多学者提出了一些近似精确匹配结果的算法, 这些算法通过局部搜索<sup>[4,5]</sup>、降低维度<sup>[6]</sup>等方法减少距离计算的次数或复杂度, 从而达到提高计算效率的目的.

最近, Barnes 等人<sup>[7]</sup>提出一个基于随机匹配的近似最近邻域块匹配算法. 该算法基于“相邻块的最近邻域块很可能也是相邻的”这一假设, 利用前面已确定的相邻块的匹配结果减少下一个块的计算量. 与已提出的近似最近邻域块匹配算法相比, Barnes 等人的算法获得了极大的加速. 随后, 他们<sup>[8]</sup>进一步将该算法

<sup>①</sup> 基金项目: 国家自然科学基金(60802013); 福建省自然科学基金(2015J01669); 福建省省属高校专项(JK2014054); 福建省中青年教师教育科研项目(JB14103)

收稿时间: 2015-07-24; 收到修改稿时间: 2015-10-08

推广到近邻块匹配问题,即搜索  $k$  个最相似的匹配块.但 Barnes 等人的算法不适合并行实现.为此, Yu 等人<sup>[9]</sup>将 Jump Flooding 传播模式<sup>[10]</sup>引入到 Barnes 算法的传播过程,提出一个并行友好的  $k$  近邻块匹配算法.在没有大幅度的匹配精确度损失的情况下, Yu 等人的算法提供了良好的并行性.算法的并行实现,可以重大加速  $k$  近邻块匹配过程.然而,这些算法主要考虑两幅图像之间的  $k$  近邻块匹配问题.而非局部相似块搜索问题一般是在同一幅图像内部的  $k$  近邻块匹配问题.根据非局部相似块搜索问题的这一具体特点,本文对 Yu 等人的算法进行改进,提出一个基于随机匹配的非局部相似块搜索算法.该算法在保持原算法的时间效率和并行性的情况下,提高了匹配精确度.

## 1 相关工作

非局部相似块搜索是在整个图像范围内搜索  $k$  个最相似的匹配块(简称  $k$  近邻块).解决这一问题的最基本方法是穷举搜索法.穷举搜索法在搜索每个查询块的  $k$  近邻块时,需要计算查询块与所有参考块之间的距离,然后取距离最小的  $k$  个参考块作为查询块的  $k$  近邻块.这一方法能够得到精确匹配的结果,但搜索过程极度耗时,难以被实际图像处理任务所用.许多学者为减少计算时间,提出了一些近似精确匹配结果的算法.其中, Barnes 等人提出的近似算法及其扩展在匹配精确度、时间效率以及内存消耗等性能指标上都优于当前主要的一些近似算法.本文的工作是建立在这些算法基础上的,下面简要回顾一下相关算法.

### 1.1 Barnes 近似最近邻域块匹配算法

Barnes 近似最近邻域块匹配算法<sup>[7]</sup>是为计算两幅图像间图像块的对应性而提出的.该算法首先定义一个最近邻域场 NNF(nearest-neighbor field).假设查询图像为  $A$ ,参考图像为  $B$ ,每个图像块的块坐标用其中心点的坐标表示. NNF 是定义在  $A$  中所有可能块坐标上的一个坐标偏移量的函数  $f: A \mapsto R^2$ .给定某一特定的距离计算函数  $D$ ,如果查询图像  $A$  中坐标为  $x_A$  的查询块的最近邻域块为参考图像  $B$  中坐标为  $x_B$  的块,则有:

$$f(x_A) = x_B - x_A \quad (1)$$

Barnes 算法包括初始化和迭代修改两个阶段.初始化阶段一般利用随机采样的偏移值或其它的一些先验知识,对最近邻域场 NNF 进行初始化.比如,采用跨整个参考图像的独立均匀的随机采样偏移值,作为

NNF 的初始值.

迭代修改阶段通过迭代交叉执行传播和随机搜索两个步骤,产生候选参考块,并利用候选参考块对 NNF 进行改善.传播步骤用前面已确定的相邻块的匹配块作为候选参考块,改善当前查询块的匹配块信息.假设当前查询块的块坐标为  $x$ ,相邻块的块坐标为  $x_n$ ,那么实际使用的相邻块的块坐标为:在奇数次迭代中,  $x_n = x - \Delta p$  ( $\Delta p$  取值为 (0,1) 或 (1,0)),在偶数次迭代中,  $x_n = x + \Delta p$ .如果用  $D(f(x))$  表示查询图像中坐标为  $x$  的块与参考图像中坐标为  $x + f(x)$  的块的距离.则传播过程,在奇数次迭代中可以用公式(2)表示,在偶数次迭代中可以用公式(3)表示.

$$f(x) = \begin{cases} f(x_n) + \Delta p & \text{for } D(f(x_n)) < D(f(x)) \\ f(x) & \text{for } D(f(x_n)) \geq D(f(x)) \end{cases} \quad (2)$$

$$f(x) = \begin{cases} f(x_n) - \Delta p & \text{for } D(f(x_n)) < D(f(x)) \\ f(x) & \text{for } D(f(x_n)) \geq D(f(x)) \end{cases} \quad (3)$$

随机搜索步骤则以查询块的当前最近邻域值为中心,随机产生的一系列候选参考块,并利用这些候选参考块改善当前查询块的匹配块信息.具体来说,首先由以下公式产生一系列的候选偏移值.

$$u_i = v_0 + w\alpha^i R_i \quad (4)$$

其中,  $v_0 = f(x)$ ,  $R_i$  为  $[-1,1] \times [-1,1]$  中的一个均匀分布的随机点,  $w$  为搜索半径,取值为图像的最大维度(即图像宽度和高度的最大值),  $\alpha$  为搜索半径衰减率,  $i=1,2,\dots$  直到  $w\alpha^i$  小于一个像素.然后,利用这些候选偏移值所表示的参考块来改善查询块的匹配块信息.这一过程可以用如下公式表示:

$$f(x) = \begin{cases} u_i & \text{for } D(u_i) < D(f(x)) \\ f(x) & \text{for } D(u_i) \geq D(f(x)) \end{cases} \quad (5)$$

### 1.2 Barnes 近似 $k$ 近邻域块匹配算法

Barnes 等人<sup>[8]</sup>进一步将上述算法推广到  $k$  近邻(kNN)块匹配问题,即搜索  $k$  个最相似的匹配块.他们首先将最近邻域场 NNF  $f$  的定义推广到  $k$  近邻域场 kNNF  $f$ ,这时  $f$  是一个多值映射,有  $k$  个值.每个值代表一个  $k$  近邻域块的块坐标,而不是原来的坐标偏移量.然后,修改最近邻域块匹配算法,使之能够处理  $k$  近邻(kNN)块匹配问题.

可能的修改方案很多,这里仅介绍最直接的一种方案.初始化阶段,给每个查询块赋  $k$  个随机的初始值.传播步骤利用相邻块的 kNN 所产生的候选参考块,来改善当前查询块的 kNN 信息.假设  $x = (x_1, x_2)$

表示当前查询块的块坐标,在奇数次迭代中, $\mathbf{x} - \Delta\mathbf{p}$  ( $\Delta\mathbf{p}$  取值为 (0,1) 或 (1,0)) 表示相邻块的块坐标, $f_i(\mathbf{x} - \Delta\mathbf{p})$  表示相邻块的第  $i$  个  $k$  近邻域块的块坐标.那么,相邻块的 kNN 所产生的候选参考块坐标可用如下公式表示:

$$f_i(\mathbf{x} - \Delta\mathbf{p}) + \Delta\mathbf{p} \quad (6)$$

在偶数次迭代中,相邻块的块坐标为  $\mathbf{x} + \Delta\mathbf{p}$  ( $\Delta\mathbf{p}$  取值为 (0,1) 或 (1,0)), 其 kNN 所产生的候选参考块坐标为:

$$f_i(\mathbf{x} + \Delta\mathbf{p}) - \Delta\mathbf{p} \quad (7)$$

随机搜索步骤则以当前查询块的每个  $k$  近邻域块为中心,随机产生的一系列候选参考块,并利用这些候选参考块来改善当前查询块的 kNN 信息.具体来说,首先候选参考块的坐标由以下公式产生:

$$\mathbf{u}_i^j = \mathbf{v}_j + w\alpha^i \mathbf{R}_i \quad (8)$$

其中,  $\mathbf{v}_j = f_j(\mathbf{x})$  ( $j=1, \dots, k$ ) 表示当前查询块的第  $j$  个  $k$  近邻域块的块坐标,  $\mathbf{R}_i$  为  $[-1,1] \times [-1,1]$  中的一个均匀分布的随机点,  $w$  为搜索半径,取值为图像的最大维度,  $\alpha$  为搜索半径衰减率,  $i=1,2, \dots$  直到  $w\alpha^i$  小于一个像素.然后,利用这些候选参考块信息来改善当前查询块的 kNN 信息.

利用候选参考块改善 kNN 时,采用了最大堆.对于每个查询块,首先,根据查询块与其  $k$  个  $k$  近邻域块的距离建立一个最大堆.其次,计算查询块与候选参考块的距离.最后,将候选参考块的距离与堆中的最大值进行比较,如果候选参考块的距离小于堆中的最大值,则用候选参考块替换最大值对应的  $k$  近邻域块,同时调整最大堆.

### 1.3 并行友好的近似 $k$ 近邻块匹配算法

因传播步骤存在严重的串行依赖性,即在传播步骤,每个查询块的 kNN 信息需要利用其相邻块的 kNN 所产生的候选参考块进行改善. Barnes 等人的算法不适合并行实现.为解决串行依赖性问题,必需改变传播模式. Yu 等人<sup>[9]</sup>将 Jump Flooding 传播模式引入到 Barnes 算法,提出一个并行友好的  $k$  近邻块匹配算法.

算法由两个基本部分组成:初始化过程和传播过程.初始化过程与 Barnes 算法一样.传播过程由多个传播回合组成,其传播步长  $L$  按  $n, n/2, n/4, \dots, 1$  递减,其中  $n$  为查询图像宽度和高度的最大值.在一个传播回合中,对于每个查询块,首先通过步长为  $L$  的 8 邻域块的 kNN 信息产生候选参考块.然后,计算每个候选参考块与查询块的距离,并与查询块当前 kNN 的最大

距离进行比较,如果候选参考块的距离更小,则用候选参考块替换最大距离对应的 kNN 块.

更多的传播回合,可以改善输出的 kNN. Yu 等人提出两种改进的方法: JF+2 算法和 JF+logn+1 算法. JF+2 算法是在上述算法(简称为 JF 算法)结束后,再进行两次步长分别为 2,1 的传播回合. JF+logn+1 算法则是在上述算法结束后,再进行 logn+1 次步长分别为  $n, n/2, n/4, \dots, 1$  的传播回合. Yu 等人的算法,在保持与 Barnes 算法相接近的匹配误差性能的情况下,提供了良好的并行性.

## 2 基于随机匹配的非局部相似块搜索算法

上述算法适用于两个不同图像之间的  $k$  近邻块匹配问题.而本文关注的非局部相似块搜索属于一幅图像内部的  $k$  近邻块匹配问题.对于这个问题,我们可以进一步利用图像的另一个特性,即局部自相似性.所谓的局部自相似性是指图像的局部结构模式可能在局部邻域内重复自己,也就是说一个图像块在它的局部邻域内可能存在多个相似块与之对应.前面两个算法没能很好地利用这一特性. Barnes 算法中,  $k$  近邻域场 kNNF  $f$  的初始值是在整个图像范围内随机采样产生的.虽然在传播过程利用相邻块的 kNN 信息来修改当前查询块的 kNN 信息.但是它只是利用相邻块的 kNN 块,而不是相邻块本身. Yu 等人的算法与 Barnes 算法相似,只不过在传播过程改为利用不同步长的 8 邻域块的 kNN 信息.这些算法,通过多次传播,虽然也可能利用到局部邻域块,但这需要较多的传播回合,从而延长了算法的收敛时间.本文建议的搜索算法,是对 Yu 等人算法的改进.主要改进原算法候选参考块的产生方式,增加从查询块的局部邻域中随机产生候选参考块这一方式.这一改进充分利用图像的局部自相似性,提高候选参考块匹配的可能性,从而缩短算法的收敛时间.在同样数量的传播回合下,改进算法可以得到更好的匹配精确度.

具体来说,在每个传播回合,增加一个随机搜索步.这一步骤通过随机采样,在查询块的局部邻域内,产生  $k$  个邻域块来改善查询块的 kNN 信息.假设查询图像  $A$  的尺寸为  $m \times n$ ,参考图像为  $A$  本身,  $k$  近邻域场 kNNF  $f$  的函数值代表  $k$  近邻域块的块坐标.本文建议算法的详细流程可以分为以下两个阶段:

1)初始化阶段:完成  $k$  近邻域场 kNNF 的初始化.

具体来说, 对于每个查询块, 首先, 通过独立均匀分布的随机采样, 在参考图像  $A$  的所有可能块坐标范围内, 产生  $k$  个随机的块坐标, 作为该查询块的初始 kNN 块. 然后, 计算该查询块与其 kNN 块的距离, 得到  $k$  个距离. 最后, 把  $k$  个初始 kNN 块和对应的距离作为该查询块的初始 kNN 信息保存.

2)传播阶段: 通过多个传播回合, 不断改善  $k$  近邻域场 kNNF. 具体来说, 在每个传播回合中, 假设其传播步长为  $l$  (第一个传播回合  $l = \max(m, n)$ ), 以后每个传播回合, 其传播步长  $l$  为前一回合的一半. 直到  $l < 1$ , 则停止传播回合), 对于查询图像  $A$  的每一查询块, 通过以下两个步骤, 产生候选参考块, 并利用候选参考块改善该查询块的 kNN 信息.

a)随机搜索步. 通过独立均匀分布的随机采样, 在查询块的局部邻域内产生  $k$  个随机块坐标, 作为候选参考块.

b)Jump Flooding 传播步. 首先, 选择以查询块为中心, 步长为  $l$  的 8 邻域块的 kNN 块, 作为候选参考块.

上述两步骤中, 利用候选参考块改善查询块 kNN 信息的实施流程是一样的. 首先, 根据查询块当前 kNN 信息中的距离建立一个最大堆  $H$ . 其次, 计算每一候选参考块与查询块的距离. 最后, 通过最大堆  $H$ , 把每一候选参考块对应的距离与查询块当前 kNN 信息中的最大距离(也就是  $H$  中的最大值)进行比较. 如果候选参考块的距离更小, 则用候选参考块及对应的距离替换查询块当前 kNN 信息中距离最大的 kNN 块及对应距离. 同时, 删除  $H$  中的最大值, 并把该候选参考块对应的距离插入  $H$ , 调整  $H$  成最大堆.

### 3 实验及结果分析

这一节, 首先通过实验, 比较了本文算法、Yu 等人的 JF+2 算法(以下简称 JF2 算法)、Barnes 等人的近似  $k$  近邻域块匹配算法(以下简称 Barnes 算法)等三个算法的实施性能. 其次通过时间复杂度的分析, 比较了三个算法的时间效率. 我们从 Caltech-256<sup>[11]</sup> 图片库的 12 类图片中, 随机选取如图 1 所示的 12 副图像做为测试图像. 图像大小的范围从 0.01MP(MP 表示百万像素单位)到 0.2MP. 为了比较各算法的匹配精确度, 采用文献<sup>[9]</sup>中定义的平均匹配误差作为评价标准. 平均匹配误差定义为查询图像  $A$  中所有可能的查询块与其  $k$  个 kNN 块的 MSE(均方误差)值的平均值, 用公式

表示为:

$$E = \sum_{x=1}^N \sum_{i=1}^k MSE(A(x), A'(x_i)) / (kN) \quad (9)$$

其中,  $N$  为查询图像  $A$  中查询块的数量,  $A(x)$  为第  $x$  个查询块,  $A'(x_i)$  为  $A(x)$  的第  $i$  个 kNN 块.  $MSE(A(x), A'(x_i))$  为  $A(x)$  与  $A'(x_i)$  的均方误差.



图 1 测试图像

在实验中, 统一设置块的大小为  $7 \times 7$ ,  $k$  为 7. 本文算法中独有的参数: 随机搜索步的局部邻域, 设置为一个以查询块坐标为中心, 大小为  $31 \times 31$  的块. Barnes 算法中的迭代次数设置为 4. 三个比较算法的实验结果如表 1 所示. 从实验结果上来看, 本文算法的平均匹配误差比 JF2 算法要小, 与 Barnes 算法很接近. 因此, 本文算法的匹配精确度要比 JF2 算法好, 与 Barnes 算法接近. 主要原因就在于本文算法充分利用局部自相似性这一特性, 加速算法的收敛速度, 在同样数量的传播回合下, 可以得到更好的匹配精确度.

表 1 三个比较算法的平均匹配误差

测试图片	JF2	Barnes	本文算法
(a)	267.09	265.92	265.73
(b)	272.23	268.16	268.69
(c)	98.62	96.08	96.43
(d)	38.52	36.90	37.42
(e)	224.20	216.95	217.80
(f)	116.01	112.67	112.21
(g)	217.92	214.03	214.54
(h)	165.40	159.12	159.92
(i)	190.83	190.06	189.35
(j)	208.04	204.17	204.57
(k)	374.95	370.12	370.47
(l)	323.37	317.87	318.99

从时间复杂度上分析,三个算法在初始化阶段是一样的,区别在于 kNN 信息改善阶段.这一阶段的操作主要是利用候选参考块改善 kNN 信息,也就是,计算候选参考块与查询块的距离,并与当前 kNN 块的最大距离进行比较.如果候选参考块的距离更小,则用候选参考块替换当前距离最大的 kNN 块.如果把上述操作看成是基本操作,就可以以产生的候选参考块数量来计算基本操作的频度,从而粗略地比较三个算法时间复杂度.下面我们以计算一个查询块的 kNN 信息为例,比较三个算法的时间复杂度.

Barnes 算法,在 kNN 信息改善阶段包括传播和随机搜索两个步骤.传播步,利用两个相邻块的 kNN 信息产生  $2k$  个候选参考块.随机搜索步通过公式(8)产生候选参考块,候选块的数量由  $k$  和  $i$  的最大取值决定. $i$  取满足条件  $w\alpha^i \geq 1$  的最大值.公式中的参数  $w$  一般设置为查询图像的最大维度  $n$ ,  $\alpha$  一般设置为 0.5,在这种情况下,产生的候选参考块数量为  $k \log_2 n$ . Barnes 算法,在这两个阶段产生的候选块总数为  $2k + k \log_2 n$ .此外,该算法的 kNN 信息改善阶段一般要迭代 4 次,因此,产生的候选块总数为  $8k + 4k \log_2 n$ ,即基本操作的频度为  $8k + 4k \log_2 n$ .

JF2 算法,在 kNN 信息改善阶段包含  $2 + \log_2 n$  个传播回合.每个传播回合产生  $8k$  个候选参考块.因此,产生的候选参考块总数为  $16k + 8k \log_2 n$ ,即基本操作的频度为  $16k + 8k \log_2 n$ .

本文建议算法,在 kNN 信息改善阶段包含  $\log_2 n$  个传播回合.每个传播回合包括随机搜索和 Jump Flooding 传播两个步骤.随机搜索步产生  $k$  个候选参考块,Jump Flooding 传播步产生  $8k$  个候选参考块.因此,产生的候选参考块总数为  $9k \log_2 n$ ,即基本操作的频度为  $9k \log_2 n$ .

因此,从时间效率上看,这三个算法具有同一数量级的时间复杂度.但如文献[9]所分析,JF2 和本文算法能够提供更好的并行性.如果采用并行化加速算法,则 JF2 和本文算法能提供更高的时间效率.

#### 4 结论

本文在 Yu 等人算法基础上,提出一个基于随机匹配的非局部相似块搜索算法.这一算法是在 Yu 等人算法基础上,对其传播回合进行改进.在每一传播回合中,增加一个随机搜索步.这一步骤充分利用图像的局部自相似性,从查询块的局部邻域中,随机产生候

选参考块,并利用这些候选参考块改善查询块的 kNN 信息.图像局部自相似性表明一个图像块在它的局部邻域内可能重复自己,也就是局部邻域内的候选参考块与查询块匹配的可能性更大.所以,本文的改进提高了候选参考块匹配的可能性,进而加速了算法收敛速度.在同样数量的传播回合下,本文算法可以得到更好的匹配精确度.实验结果验证了这一论述,在同样的传播回合下,本文算法的平均匹配误差要比 JF2 小,与 Barnes 算法接近.从算法的时间效率上看,本文算法、JF2 算法和 Barnes 算法具有同一数量级的时间复杂度.但本文算法、JF2 算法有更好的并行性.总之,本文算法在保持原算法的时间效率和并行性的情况下,提高了匹配精确度.

#### 参考文献

- 1 Buades A, Coll B. A non-local algorithm for image denoising. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, CA. 2005. 60–65.
- 2 Wong A, Orchard J. A nonlocal-means approach to exemplar-based inpainting. IEEE International Conference on Image Processing. San Diego, California, USA. 2008. 2600–2603.
- 3 Glasner D, Bagon S, Irani M. Super-resolution from a single image. IEEE International Conference on Computer Vision. Kyoto, Japan. 2009. 349–356.
- 4 Zhang H, Yang J, Zhang Y, Huang T. Non-local kernel regression for image and video restoration. The 11th European Conference on Computer Vision (ECCV). Crete, Greece. 2010. 566–579.
- 5 Zhang H, Yang J, Zhang Y, Huang T. Image and video restorations via nonlocal kernel regression. IEEE Transactions on Cybernetics, 2013, 43(3): 1035–1046.
- 6 Kopf J, Fu C, Cohen-OR D, et al. Solid texture synthesis from 2d exemplars. ACM Siggraph, 2007, 26(3): 1–9.
- 7 Barnes C, Shechtman E, Finkelstein A. PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Trans. on Graphics (Proc. SIGGRAPH), 2009, 28(3): 1–11.
- 8 Barnes C, Shechtman E, Finkelstein A. The generalized PatchMatch correspondence algorithm. The 11th European Conference on Computer Vision (ECCV). Crete, Greece. 2010. 29–43.
- 9 Yu P, Yang X, Chen L. Parallel-friendly patch match based on jump flooding. The 9th International Forum of Digital TV & Wireless Multimedia Communication (IFTC 2012). Shanghai, China. 2012. 15–21.
- 10 Rong G, Tan TS. Jump flooding in GPU with applications to Voronoi diagram and distance transform. ACM Symposium on interactive 3D graphics and Games. Redwood City, CA. 2006. 109–116.
- 11 Caltech-256 Object Category Dataset. [http://www.vision.caltech.edu/Image\\_Datasets/Caltech256/](http://www.vision.caltech.edu/Image_Datasets/Caltech256/).