一种结合空间预测和 CDS 的快速块匹配算法

禹 晶 [1] 苏开娜 [1]

摘 要 运动估计是根据视频序列中时间上相关的信息估计场景或目标的二维运动向量场的过程. 因为块运动估计的简单性和有效性, 它已经成为目前使用最广泛的运动估计方法. 本文设计了一种结合空间预测和 CDS 的快速块匹配算法. 若当前块和相邻块的运动相似, 则选择相邻块的运动向量中使当前块的匹配误差最小的一个作为当前块运动向量的预测估计, 再以该预测值为中心, 比较 SDSP 上搜索点的块匹配误差. 若当前块和相邻块的运动不相关, 则采用 CDS 算法从原点开始搜索运动向量. 实验结果表明, 本文设计的算法兼顾了搜索速率和精度, 相比 N3SS、DS、HEXBS、CDS、CDHS 算法, 更好地适用于超分辨率图像复原.

关键词 运动估计, 块匹配算法, 空间预测, CDS 算法
中图分类号 TP391

A Hybrid Method for Combining Spatial Prediction with the CDS Algorithm

YU Jing[1] SU Kai-Na[1]

Abstract Motion estimation refers to estimating 2-D motion vector field of the scene or object according to temporal information redundancy in a clipped video. Because of its simplicity and efficiency, block-based motion estimation has recently been widely used. This paper proposes a hybrid method for combining spatial prediction with the CDS algorithm. If the motion of the current block is similar to that of its neighbor blocks, we choose the best candidate block from the neighbor blocks and use its motion vector to form an initial estimate for the current block. The neighbor block whose motion vector yields the minimum block distortion is called the best candidate block. The true motion vector is then obtained by comparing the search points of SDSP centered at the initial estimate. If the current block is not correlated with its spatial neighbors, we search for the motion vector from the origin of the search window using the CDS algorithm. Experimental results show that the proposed algorithm achieves a better tradeoff between search speed and accuracy for super resolution restoration than N3SS, DS, HEXBS, CDS, and CDHS.

Key words Motion estimation, block-matching algorithm, spatial prediction, CDS algorithm

1 引言

超分辨率图像复原技术是将多幅模糊、有噪、频谱混叠的低分辨率降质图像 (或视频序列) 融合估计出一幅高质量高分辨率的图像. 运动估计就是根据视频序列中时间上相关的信息估计场景或目标的二维运动向量场的过程. 运动估计在超分辨率图像复原中的作用是把所有低分辨率观测帧的像素映射到参考帧的相应位置, 实现图像细节的恢复. 精确的运动估计是超分辨率图像复原的关键. 因为块运动估计的简单性和有效性, 它已经成为目前使用最广泛的运动估计方法.

块匹配算法 (Block-matching algorithm, BMA) 是将当前帧划分为图像块, 对于当前帧中的每一个图像块, 在参考帧的搜索窗口内搜索最佳匹配块, 参考帧中的最佳匹配块相对于当前帧图像块的位移称之为当前帧图像块的运动向量 (Motion vector, MV).

到目前为止, 国内外提出了很多快速块匹配算法, 根据不同的需求, 每种算法力求达到算法复杂度、搜索速率和图像质量的最佳折衷. 和全搜索相比, 它们在搜索速率上有很大的提高, 然而在搜索的准确度上仅有微小的下降. 现有的快速算法大致可分为如下四类:

1.1 采用固定搜索模式集的快速块匹配算法 (A 类)

这类快速搜索算法是基于全局误差场单调分布的假设, 误差曲面的特性直接影响块运动估计算法的性能. 对于图 1(a) 所示的搜索窗口内有且仅有一个全局极小点的单峰误差曲面, 这类算法经过搜索少量的像素就可以到达全局极小点. 但对于图 1(b) 所示的多个局部极小点的误差曲面, 这类算法不一定收敛到全局极小点, 很容易陷入某个局部极小点.

这类算法使用固定搜索模式集搜索运动向量, 且每个块的运动向量的搜索彼此独立. 众所周知的搜索算法有三步搜索 (Three step search, 3SS)[1]、

收稿日期 2005-11-25 收修改稿日期 2006-4-7
Received November 25, 2005; in revised form April 7, 2006
北京市自然科学基金项目 (4031004), 北京市教委科技发展计划项目 (KM200310005006) 资助
Supported by Beijing Natural Science Foundation (4031004), Scientific Research Common Program of Beijing Municipal Commission of Education(KM200310005006)
1. 北京工业大学计算机学院 北京 100022
1. College of Computer Science and Technology, Beijing University of Technology, Beijing 100022
DOI: 10.1360/aas-007-0355

新三步搜索 (New three step search, N3SS)[2]、四步搜索 (Four step search, 4SS)[3]、菱形搜索 (Diamond search, DS)[4,5]、六边形搜索 (Hexagon-based search, HEXBS)[6]、十字形菱形搜索 (Cross-diamond search, CDS)[7]、十字形菱形六边形搜索 (Cross-diamond-hexagonal search, CDHS)[8] 等. 这类算法具有简单性和齐整性的优点, 容易实现.

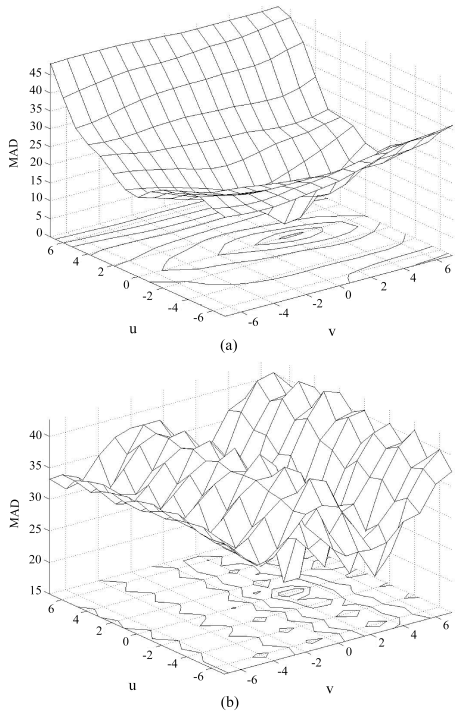

图 1 两个块的 MAD(Mean absolute difference) 误差曲面 (a) 误差曲面仅有一个全局极小点 (b) 误差曲面有多个局部极小点

Fig. 1 MAD (Mean absolute difference) error surface for two different blocks. (a)Unimodal error surface with a global minimum error points (b)Non-unimodal error surface with multiple local minimum error points

1.2 基于块间相关的快速块匹配算法 (B 类)

这类方法利用与当前块在空间上和 (或) 时间上相关的邻近块的运动向量预测当前块的运动向量. 通过计算邻近块运动向量的统计特征 (如均值、中间值, 加权平均值等)[9,10] 或者根据某准则选择其中一个邻近块的运动向量 (如选择使当前块匹配误差最小的运动向量)[11] 作为当前块运动向量的预测值. 获得预测值后, 对搜索窗口和搜索中心重新定义, 在新的搜索窗口内执行缩小的全搜索算法[11]. 还有一些算法, 以预测向量为中心执行缩小的全搜索算法, 不重新定义搜索窗口[9,10].

1.3 分级快速块匹配算法 (C 类)

这类方法[12,13] 利用不同分辨率的图像, 块的大小不变. 通过对图像下采样, 降低图像的分辨率, 级越低, 图像的分辨率越低. 这类方法用低一级所得运动向量作为高一级运动估计的初值, 通过由粗到细的过程, 得到原始图像的每个块的运动向量. 这类方法有相对较好的运动估计精确度, 但在搜索运动向量的细化过程中, 需比较缩小的全搜索窗口内所有位置的块匹配误差, 因此仍然耗费很大的计算量.

1.4 减少参与块匹配误差计算的像素个数的快速块匹配算法 (D 类)

上述的三类块运动估计算法都是通过限制搜索位置的个数, 降低运动估计的计算量. 这类方法通过减少参与块匹配误差计算的像素个数加速运动估计. 一种方法是在每一个块中分别沿水平和垂直方向按 2 : 1 的比率下采样, 这样, 计算量将降低为原来的 1/4.

2 算法构想

如前所述, A 类算法的搜索很容易陷入某个局部极小点. 但是, 局域内误差场单调分布的假设是成立的. 如果运动向量的初始估计在全局极小点的邻域内, 再用 A 类算法搜索, 那么经过搜索少量的像素就可以到达全局极小点. 对于一个图像序列, 运动目标往往覆盖许多图像块, 空间上相邻块的运动是高度相关的. 因此, 通过空间预测估计初始的运动向量, 期望运动向量的初值在误差曲面全局极小点的邻域内.

根据以上分析, 本文设计了一种结合空间预测和 CDS 的快速块匹配算法. 该算法的过程是: 从左到右, 从上到下依次扫描当前帧中的每个图像块. 左上角的块是扫描到的第一个块, 没有可作预测的相邻块, 故用全搜索算法比较搜索窗口内所有位置的块匹配误差, 找出最小块匹配误差 (Minimum block distortion, MBD) 点, 即为求解的运动向量. 对于第一行、第一列和最后一列的块, 没有足够的相邻块用来预测当前块的运动向量, 且这些块的运动向量的准确性又直接影响其他块搜索的准确性, 所以用修改的快速全搜索算法 SEA(见 3.1 节). 对于其他块, 首先通过空间预测估计初始的运动向量, 预测集选择当前块左边、上边、右上方相邻块的运动向量和 $(0,0)^{\mathrm{T}}$, 并分别计算它们指向的块和当前块之间的块匹配误差. 忽略左上方相邻块的原因是它和左边、上边的相邻块高度相关, 它能被这两个块或其中的一个块所代替. 若运动向量为 $(0,0)^{\mathrm{T}}$ 对应的块匹配误差最小, 表明当前块和相邻块的运动不相似 (例如场景中存在独立的运动物体), 则采用 CDS 算

法从原点开始搜索运动向量；否则, 从相邻块的运动向量中选择使当前块匹配误差最小的一个作为预测向量, 再以该预测值为中心, 比较 SDSP(见图 2) 上的 5 个点, 不重新定义搜索窗口, 若周围的 4 个点之一是 MBD 点, 则以该 MBD 点为中心形成新的 SDSP, 如此继续, 直到小菱形的中心是 MBD 点或搜索点到达预定的搜索窗口边界, 每次仅需再计算 3 个或 2 个点的块匹配误差. 实验表明, 用小菱形代替 3×3 的方形 (见图 3) 搜索, 搜索点数明显减少, 而搜索精度的降低却是可以忽略的.

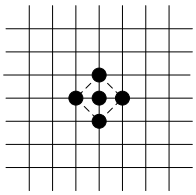

图 2　小菱形搜索模式 (SDSP)
Fig. 2　Small diamond search pattern (SDSP)

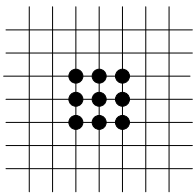

图 3　3×3 方形搜索模式
Fig. 3　3×3 square search pattern

3　算法实现

3.1　修改的 SEA

SEA[14] 是一种快速全搜索算法, 它的运动估计结果和全搜索算法相同, 能够搜索到全局极小点, 但它的计算量有较大程度地减小.

3.1.1　SEA 的原理

SEA 通过一个有上下界约束的不等式限制了搜索空间. 设块的大小是 $N \times N$, 允许搜索的最大步长是 $window$, $I_k(i,j)$ 和 $I_{k-\triangle k}(i,j)$ 分别表示当前帧和参考帧中像素 (i,j) 处的灰度值, SAD (Sum of absolute differences) 准则用于计算块匹配误差, 得到不等式 (1) 如下：

$$R - \mathrm{SAD}(u_0, v_0) \le M(u,v) \le R + \mathrm{SAD}(u_0, v_0) \tag{1}$$

其中, u , v 表示运动向量的两个分量, $-window \le u, v \le window$. $R = \sum_{i,j} |I_k(i,j)|$ 表示当前块的所有像素灰度值之和, $M(u,v) = \sum_{i,j} |I_{k-\triangle k}(u+i, v+j)|$ 表示运动向量 (u,v) 对应的参考帧中匹配块的所有像素灰度值之和. $\mathrm{SAD}(u_0, v_0)$ 为初始值, 表示初始预测估计的运动向量 (u_0, v_0) 对应的 SAD.

SEA 仅需计算参考帧中搜索窗口内满足不等式 (1) 的搜索位置的块匹配误差. 显然, 满足不等式 (1) 的搜索点数比搜索窗口内总的搜索点数少, 因此, 算法降低了计算量. SEA 的效率取决于参考帧中匹配块的像素灰度值和的快速计算和初始的运动估计.

从不等式 (1) 中可以看到, 若界的范围大, 或搜索窗口内搜索点对应的块内像素灰度值之和彼此相近, 都会增加搜索位置的个数. 场景改变大或初始估计不准确会导致不等式 (1) 界的范围较大, 而图像平滑部分的搜索点对应的块内像素灰度值之和彼此相近.

3.1.2　参考帧中匹配块的像素灰度值和的快速计算

根据不等式 (1), 参考帧中每个搜索点对应的块内像素灰度值之和必须已知. 如果当判断搜索窗口内的搜索点是否满足不等式 (1) 时, 再累计块内像素灰度值之和, 那么计算量相当大. 本节讨论参考帧中匹配块的像素灰度值和的快速计算. 设图像的大小是 $H \times W$, 在像素 (i,j) 处的灰度值用 $f(i,j)$ 表示. 如图 4 所示, 将参考帧划分为 $H - N + 1$ 条窄带, 每一条窄带包含 N 行像素.

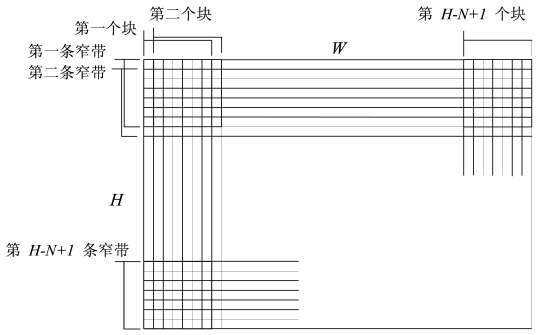

图 4　块内像素灰度值和的快速计算示意图
Fig. 4　Fast calculation of the sum of all the pixels in a matching candidate block

第 1 步. 计算第一条窄带每一列像素灰度值之和, 记为 $C_{11}, C_{12}, \cdots, C_{1,W}$. 第二条窄带每一列像素灰度值之和记为 $C_{21}, C_{22}, \cdots, C_{2,W}$, 显然 $C_{21} = C_{11} - f(1,1) + f(N+1,1)$, $C_{22} = C_{12} - f(1,2) + f(N+1,2), \cdots, C_{2,W} = C_{1,W} - f(1,W) + f(N+1,W)$. 同理计算得到其余窄带每一列的像素灰度值之和. C 矩阵的大小为 $(H - N + 1) \times W$.

第 2 步. 第一条窄带第一个块的像素灰度值之和记为 SN_{11}, 则 $SN_{11} = C_{11} + C_{12} + \cdots + C_{1,N}$. 沿水平方向向右平移 1 个像素, 得到第二个块的像素灰度值之和, 记为 SN_{12}, 则 $SN_{12} = SN_{11} - C_{11} + C_{1,N+1}$, 依此类推, 可以得

到 $SN_{13}, SN_{14}, \cdots, SN_{1,W-N+1}$. 同理计算得到其余窄带的每一个块的像素灰度值之和. SN 矩阵的大小为 $(H-N+1)\times(W-N+1)$.

本文设计的算法仅需对当前帧中第一行、第一列和最后一列各块对应的搜索窗口内的每一个匹配块的像素灰度值求和, 因此, 只需计算如图 5 所示的阴影部分的 $C_{i,j}(i=1,2,\cdots,H-N+1; j=1,2,\cdots,W)$ 和 $SN_{m,n}(m=1,2,\cdots,H-N+1; n=1,2,\cdots,W-N+1)$ 的值, 而不用计算整个 C 和 SN 矩阵的值.

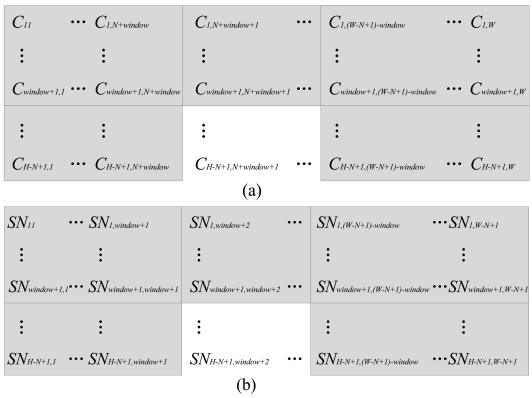

图 5 C 和 SN 矩阵中的阴影区域 (a) C 矩阵 (b) SN 矩阵

Fig. 5 Elements shown shaded of matrices C and SN (a) Matrix C (b) Matrix SN

3.1.3 初始 MV 的估计

初值 $\mathrm{SAD}(u_0, v_0)$ 越小, 搜索空间越小. 当前块的运动向量可以用空间上相邻块的运动向量近似估计. 当前帧第一行各块的预测集的元素选择左边相邻块的运动向量和 $(0,0)^{\mathrm{T}}$, 第一列各块的预测集的元素选择上边相邻块的运动向量和 $(0,0)^{\mathrm{T}}$, 最后一列各块的预测集的元素选择左边、上边相邻块的运动向量和 $(0,0)^{\mathrm{T}}$. 预测集中使当前块匹配误差最小的运动向量即为初始的运动估计 (u_0, v_0), 从而得到要求的初值 $\mathrm{SAD}(u_0, v_0)$.

初始的 $\mathrm{SAD}(u_0, v_0)$ 在搜索过程中并不是恒不变的, 如果搜索位置 (u_p, v_p) 对应的 $\mathrm{SAD}(u_p, v_p)$ 小于 $\mathrm{SAD}(u_0, v_0)$, 那么不等式 (1) 中 $\mathrm{SAD}(u_0, v_0)$ 的值用 $\mathrm{SAD}(u_p, v_p)$ 代替, 这样进一步减少了搜索位置.

3.2 结合空间预测和 CDS 的快速块匹配算法

3.2.1 CDS 算法

本文将多种快速块匹配算法进行了总结比较, 六边形的搜索速率高于菱形, 但搜索精度过低. 对于静止或近似静止块较多的视频序列, 提前终止搜索的技术不但可以提高搜索速率, 而且可以提高搜索的准确率. 超分辨率图像复原不仅要求速率, 而且要求精度. 所以, 为了兼顾搜索速率和精度, 当相邻块的运动向量不能预测当前块的运动向量时, 采用 CDS 算法从搜索窗口的原点开始搜索运动向量.

本节回顾 CDS 算法. CDS 算法的搜索过程如下:

第 1 步 (开始). 将 CSP(十字形搜索模式) 的中心放在搜索窗口的原点, 比较 CSP 上的 9 个点, 若十字形中心是 MBD 点, 则算法终止, 称为一步终止, 否则转向第 2 步.

第 2 步 (半菱形搜索). 加入距 CSP 上的 MBD 点最近的, 且属于以搜索窗口原点为中心的 LDSP(大菱形搜索模式) 上的 2 个点 (即坐标为 $(\pm1, \pm1)$ 的 4 个点中的 2 个点), 参与比较. 若上一步的 MBD 点为 $(\pm1, 0)$ 或 $(0, \pm1)$, 且此步的 MBD 点和上一步 CSP 上的 MBD 点相同, 则算法终止, 称为两步终止, 否则转向第 3 步.

第 3 步 (搜索). 以上一步中的 MBD 点为中心形成新的 LDSP. 若 LDSP 中心是 MBD 点, 则转向第 4 步, 否则重复执行第 3 步.

第 4 步 (结束). 将 LDSP 切换为 SDSP(小菱形搜索模式). SDSP 上 5 个点中的 MBD 点的坐标为最终求解的运动向量, 指向最佳匹配块.

对于一步终止和两步终止的运动向量的搜索, CDS 算法分别需比较 9 个和 11 个搜索点. 对于没有运动或运动极微小的块的运动向量, 例如属于背景区域的块的运动向量, 通过一步搜索就可以得到; 对于运动稍大的块的运动向量, 通过两步搜索也可以得到.

3.2.2 结合空间预测和 CDS 的快速块匹配算法的实现

本文设计了一种结合空间预测和 CDS 的快速块匹配算法, 该算法的实现过程如下:

```
for m=1:BH /*BH 为垂直方向的块数 */
  for n=1:BW /*BW 为水平方向的块数 */
    if 当前块在左上角
      全搜索算法
    elseif 当前块在第一行、第一列或最后一行
      修改的 SEA
    else /* 其他块 */
      通过空间预测估计初始的运动向量
      if (0,0)^T 对应的块匹配误差最小
        CDS 算法
      else
        以预测向量为中心, 比较 SDSP 上搜索
        点的块匹配误差
      end (if)
    end (if)
  end (for)
end (for)
```

表 1　在不同视频序列上各种块匹配算法的性能比较

Table 1　Performance comparisons of various block-matching algorithms using different video sequences

	SEA	N3SS	DS	HEXBS	CDS	CDHS	本文算法
			Mobile				
MSE	29.9902	31.0605	30.6395	48.8654	30.2350	30.2529	30.1967
Search Points	109.5938	19.8333	15.5444	12.4728	13.6864	13.1858	11.3105
Avg. Prob	1	0.9296	0.9290	0.6735	0.9426	0.9389	0.9407
Avg. Dist	0	0.2731	0.3017	0.6266	0.2670	0.2769	0.2572
			Tennis				
MSE	133.6361	141.2911	143.1319	148.0842	146.2507	157.5826	140.5608
Search Points	188.8163	17.3119	14.3052	11.4926	11.2607	7.2363	16.2807
Avg. Prob	1	0.8889	0.9133	0.8985	0.9133	0.8941	0.9281
Avg. Dist	0	0.3370	0.3467	0.3885	0.3505	0.4336	0.2845
			Garden				
MSE	187.0346	298.3756	216.2669	219.8203	219.1742	230.1548	193.2317
Search Points	108.2667	20.8432	21.6750	15.8848	23.3992	19.5508	14.4803
Avg. Prob	1	0.3508	0.7992	0.7220	0.7977	0.7705	0.8758
Avg. Dist	0	1.8622	1.0271	1.4672	1.1202	1.2422	0.5365
			Surfside				
MSE	15.4450	18.8202	16.3269	16.9755	16.3932	17.0565	16.0171
Search Points	134.1201	21.8936	23.8711	16.5879	25.4565	21.2881	15.9082
Avg. Prob	1	0.3779	0.7739	0.5386	0.7632	0.6123	0.7593
Avg. Dist	0	2.0813	1.4289	1.8543	1.4808	1.8962	1.3645

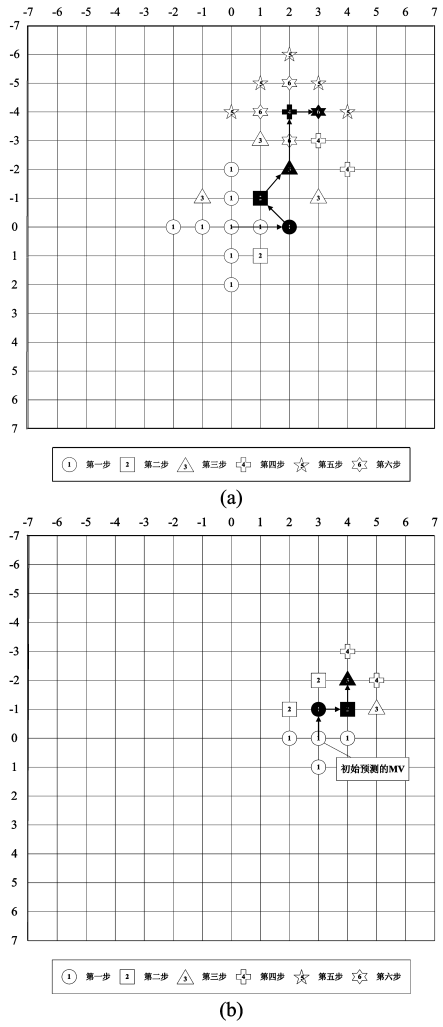

图 6　本文算法搜索 MV 的两个例子 (a) 第一种情况 (b) 第二种情况

Fig. 6　Examples of the proposed algorithm search procedure (a)The first condition (b)The second condition

图 6(a) 和 (b) 分别展示了在上述两种情况下搜索 MV 的例子. 第一种情况是当 $(0,0)^{\mathrm{T}}$ 对应的块匹配误差最小时, 采用 CDS 算法从原点开始搜索运动向量；第二种情况是当某相邻块对应的块匹配误差最小时, 以该相邻块的 MV 为中心继续搜索当前块的运动向量.

4　性能评价

评价快速块匹配算法通常有 4 个准则：1) 平均每个像素的误差平方和 (MSE)；2) 每个块的平均搜索点数 (Search points)；3) 采用快速算法搜索得到的运动向量和全搜索算法搜索得到的最优运动向量相等的块个数占总数的比例 (Avg. Prob)；4) 采用快速算法搜索得到的运动向量和全搜索算法搜索得到的最优运动向量之间的平均距离 (Avg. Dist).

根据上述的 4 个准则, 本文在 Mobile、Tennis、Garden 和 Surfside 等测试序列上将 SEA、N3SS、DS、HEXBS、CDS、CDHS 算法和本文设计的算法做比较. 块的大小选择 16×16, 用 MAD 准则计算块匹配误差. Mobile 序列中, 整个场景向右移动, 场景中存在多个目标, 墙上的日历向上平移, 玩具火车向左近似作平移运动, 火车推动前面的小球作旋转运动. Tennis 序列中, 球拍托球, 使球上下运动, 背景是静止的. Garden 序列中, 整个场景向右运动, 运动程度较大. Surfside 序列中, 背景是汹涌澎湃的大海, 前景中的三个人都向前缓慢移动, 运动相似. 表 1 比较了在上述的 Mobile、Tennis、Garden 和 Surfside 4 个序列上各种块匹配算法的 4 个评价指标.

从表中可以看到, SEA 的性能最优, 但每个块的平均搜索点数过高. 菱形的平均搜索点数高于六

边形, 但搜索误差低于六边形. 若当前帧中图像块的运动向量之间有较高的相关性, 例如 Garden 和 Surfside 序列, 则本文的算法可以同时显著地加速搜索速率和提高搜索精度. 而在 Tennis 序列中, 静止的背景中存在多个目标, 一部分图像块运动向量之间的相关性较差, 为此, 本文的算法比较了更多的搜索点, 虽然每个块的平均搜索点数略有增加, 但是搜索误差相对较小. Mobile 序列中也存在多个目标, 但整个场景的运动是全局性的, 因此本文的算法既提高了搜索的速率, 又提高了搜索的准确率.

5 结论

运动估计是超分辨图像复原中的重要技术, 其中块匹配算法是常用的运动估计方法. 本文设计了一种结合空间预测和 CDS 的快速算法. 实验结果表明, 该算法兼顾了搜索速率和精度, 相比 N3SS、DS、HEXBS、CDS、CDHS 算法, 更好地适用于超分辨率图像复原.

超分辨率图像复原需要分数像素精度的运动估计. 本文算法的运动估计结果仅仅精确到整数像素精度, 再找出结果在参考帧二倍插值图像中对应的点, 比较它和它周围的 8 个点的块匹配误差, 找到 MBD 点, 从而达到半像素精度的运动估计.

References

1 Koga T, Iinuma K, Hirano A, Iijima Y, Ishiguro T. Motion compensated interframe coding for video conferencing. In: Proceedings of IEEE National Telecommunication Conference, IEEE, 1981. G5.3.1∼G5.3.5

2 Li R, Zeng B, Liou M L. A new three-step search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1994, **4**(4): 438∼442

3 Po L M, Ma W C. A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, **6**(3): 313∼317

4 Tham J Y, Ranganath S, Ranganath M, Kassim A A. A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, **8**(4): 369∼377

5 Zhu S, Ma K K. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 2000, **9**(2): 287∼290

6 Zhu C, Lin X, Chau L P. Hexagon-based search pattern for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, **12**(5): 349∼355

7 Cheung C H, Po L M. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, **12**(12): 1168∼1177

8 Cheung C H, Po L M. Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. *IEEE Transactions on Multimedia*, 2005, **7**(1): 16∼22

9 Luo L J, Zou C, Gao X Q. A new prediction search algorithm for block motion estimation in video coding. *IEEE Transactions on Consumer Electronics*, 1997, **43**(1): 56∼61

10 Xu J B, Po L M, Cheng C K. Adaptive motion tracking block-matching algorithms for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 1999, **9**(7): 1025∼1029

11 Hsieh C H, Lu P C, Shyn J S, Lu E H. Motion estimation algorithm using inter-block correlation. *Electronic Letters*, 1990, **26**(5): 276∼277

12 Uz K M, Vetterli M, LeGall D. Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Transactions on Circuits and Systems for Video Technology*, 1991, **1**(1): 86∼99

13 Zhang Y Q, Zafar S. Motion-compensated wavelet transform coding for color video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 1992, **2**(3): 285∼296

14 Li W, Salari E. Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing*, 1995, **4**(1): 105∼107

禹 晶 北京工业大学计算机学院硕士研究生. 主要研究方向为模式识别与图像处理.
E-mail: yujing@emails.bjut.edu.cn
(**YU Jing** Master student at College of Computer Science and Technology, Beijing University of Technology. Her research interest covers pattern recognition and image processing.)

苏开娜 北京工业大学计算机学院教授. 1970 年毕业于清华大学电子工程系, 主要研究方向为图像处理、计算机视觉及视频智能监测. 本文通信作者. E-mail: sukaina@bjut.edu.cn
(**SU Kai-Na** Professor at College of Computer Science and Technology, Beijing University of Technology. She graduated from Department of Electronic Engineering, Tsinghua University in 1970. Her research interest covers image processing, computer vision, and video surveillance. Corresponding author of this paper.)