

一种基于运动对象的形状编码新算法

石旭利, 张兆扬

(上海大学通信与信息工程学院, 上海 200072)

摘要: 本文提出一种基于运动对象的形状编码新方法,称之为基于曲率尺度空间 CSS(Curvature Scale Space)的自适应算术形状编码算法(CSSAS). 本算法主要分为两种编码模式: 帧内模式和帧间模式. 在帧内模式中,我们在对CSS算法进行改进的基础上,对任意形状对象的形状信息进行特征点的分层提取;并用自适应算术编码算法对提取的特征点进行编码. 在帧间模式中,提出了一种基于曲率尺度空间图(CSSI)的任意形状对象的运动估计算法. 上述运动估计/补偿后所获得的形状曲线的匹配部分使用基于弧长索引的压缩编码算法,而对于不匹配部分,采用与帧内CSSAS算法相同的方法进行压缩编码. 实验结果表明,本文提出的CSSAS算法与MPEG-4校验模型中基于上下文的算术形状编码算法(CAE)相比,在帧内模式时,CSSAS比CAE压缩比提高约25%,在帧间模式 D_n 较大时,CSSAS比CAE压缩比明显提高,而在重构形状的主观质量上,CSSAS这两种模式均优于CAE.

关键词: MPEG-4; 形状编码; CSS; CSSAS; CAE

中图分类号: TN911.22; TM919.81 **文献标识码:** A **文章编号:** 0372-2112(2004)01-0042-04

A New Motion Object-Based Shape Coding Algorithm

SHI Xu li, ZHANG Zhao yang

(School of Communication and Information Engineering, Shanghai, Univ., Shanghai 200072, China)

Abstract: In the paper, a new motion object based shape coding called CSSAS (curvature scale space adaptive arithmetic shape coding) is proposed. There are two coding modes, the intra and inter modes. In the intra mode, the modified CSS theory is used to extract feature points of the shape curve of arbitrary shape video object (VO) at different scales. All of the feature points are encoded by adaptive arithmetic encoding. In the inter mode, a CSSF based (Curvature scales space image based) motion compensation/estimation algorithm of arbitrary shape VO is proposed. The matched segments of shape are coded by using arc length index algorithm and the mismatched segments are coded by using intra CSSAS algorithm. Simulation results show that the proposed method is much better than CAE of MPEG-4 VM in coding efficiency (in inter mode, $D_n \geq \text{threshold}$) and subjective quality for both intra and inter modes.

Key words: MPEG-4; shape coding; CSS; CSSAS; CAE

1 引言

MPEG-4所支持的基于内容或基于视频对象(VO)的任意形状编码是其区别于传统编码标准的许多新特点中最核心的内容^[1]. 目前,国外在形状编码上的研究主要有:MPEG-4校验模型中N. Brady提交的基于上下文的算术形状编码(CAE)^[2,3];MPEG-4核心实验中K. J. O'connell为代表的基于顶点的形状编码(VBSE)^[4],以及色度键、基线等其他的形状编码方法. 其中,CAE已被MPEG-4采用. 这种基于块的技术虽然可以和框架中的基于块的纹理编码相兼容,但同时却无法避免帧内帧间编码中存在的方块效应.

为此,本文提出一种新的形状编码算法CSSAS. 其贡献在于:在帧内模式中,将因缺乏位置信息只能用于形状检索而不能用于形状编码的CSS算法^[5],通过用跟踪法改进,获得位置信息后,使之能用于形状编码,从而显著地提高压缩效率;在帧间模式中,在提出了一种基于曲率尺度空间图(CSSI)的曲

线运动估计算法的基础上,又用弧长索引法对匹配部分进行编码而对不匹配部分用帧内模式中改进的CSS方法编码,从而提高了编码的效率.

2 CSSAS形状编码算法总流程

传统的视频压缩编码是基于帧的图像序列的,而基于内容的视频编码则是基于VO的VOP序列的. 如果VOP处在纹理编码的I帧中,或VOP第一次出现在当前帧中,那么没有关于VOP的形状信息作为参考形状,这种情况的VOP称为第1类VOP,对其形状信息必须用帧内编码模式进行编码. 尺寸很小的VOP为第2类VOP,对其形状信息也采用帧内编码模式. 而对其余的VOP形状信息,则采用下式确定其为帧内还是帧间编码模式.

$$diff = \frac{\text{参考 Alph 平面像素} - \text{当前 Alph 平面像素}}{\text{像素总数}}$$

如果 $diff$ 大于一个阈值 Δ ,则对当前形状曲线进行帧内编码,

否则进行帧间编码.图 1 为编码总流程.

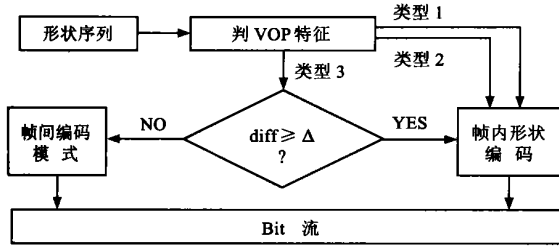


图 1 CSSAS 编码总流程

在帧内模式和帧间模式中,我们的算法对输入的二值 Alpha 平面序列中的曲线序列进行压缩编码,而 CAE 是对矩形界定的平面进行分块压缩编码,存在方块效应^[1].

3 CSSAS 的帧内编码模式

3.1 用改进的 CSS 算法进行特征点的提取

3.1.1 CSS 算法 CSS 算法首先由 F. Mokhtarian 等提出^[5].主要用于图像的远程检索中.其原理是先将形状曲线处理为以弧长步长 1 递增的离散点的集合:

$$\{f(s) = (x(s), y(s)) | 1 \leq s \leq N\} \quad (1)$$

其中 N 为总弧长即形状曲线的总像素数.再用高斯函数对式(1)的曲线坐标进行卷积:

$$\begin{aligned} x(s, t) &= x_0(s) * g(s, t) \\ y(s, t) &= y_0(s) * g(s, t) \end{aligned} \quad (2)$$

式中, t 代表尺度参数.它与高斯函数的参数 δ 满足关系式 $\delta = \sqrt{2t}$,文献^[5]中的证明表示,式(2)表示的曲线族具有下列性质:

- (1) 包含次序不变.若形状 A 在形状 B 的内部,则曲率形变以后,这种包含关系不变.
- (2) 在曲率形变过程中不会引起曲线的交叉,即所有的曲线随着曲率形变操作逐渐变为凸曲线并最终成为一个圆点.
- (3) 局部曲率绝对值最大点(角点)和局部曲率零点(拐点)的数量在演化过程中单调减少.
- (4) 此过程可并行实现.
- (5) 曲线总长逐渐缩短.

上述性质(1)、(2)保证了用高斯函数对曲线进行卷积后所得到的一系列曲线仍保持各自的独立性,因此对于任意形状的物体来说,仍可利用上述理论进行特征点的提取.性质(3)表明利用高斯函数对曲线进行卷积后,可形成曲线的多尺度空间表示.原始曲线表示为最精细尺度的空间曲线.随着尺度的增加,曲线的某些细节信息被去除而保留基本的信息.将这种“剥离”的操作进行下去,直到在要求的粗糙尺度上获得需要的曲线的最基本的信息.称这样形成的一系列曲线所构成的空间为原始曲线的多尺度空间.然后用曲率函数计算每一点的曲率,称局部曲率负最小和局部曲率正最大点为特征点“角”.以弧长 s 为参数的尺度为 t 的曲线的曲率方程为:

$$k(s, t) = \frac{x_{ss}(s, t)y_{ss}(s, t) - x_{ss}(s, t)y_s(s, t)}{((x_s(s, t))^2 + (y_s(s, t))^2)^{3/2}} \quad (3)$$

式中, $x_s(s, t) = \frac{\partial}{\partial s}(x(s) * g(s, t))$

$$x_{ss}(s, t) = \frac{\partial^2}{\partial s^2}(x(s) * g(s, t))$$

利用式(2)在高斯函数的某一 t 下,对多尺度空间中的某一条曲线进行特征点“角”的提取,在形状剧变部分获得密集的点而在平坦部分获得稀疏的点集.从而既可以保留重要的信息,又能用较少的比特表示图像.

3.1.2 特征点的定位跟踪算法 但是 CSS 算法的最大的缺陷在于性质(e),长度的缩减使得曲线上点的位置偏离其在原始曲线上的位置,会使得特征点的位置信息无效,这就不能在 VOP 形状编码中.为此,我们用特征点的定位跟踪算法给予解决:在较粗的尺度 t 上将特征点角提取出来,得到角点集合 P_t ;减小尺度的值,在较细的尺度 $t-1$ 上将特征点角提取出来,得到角点集合 P_{t-1} .根据性质(c),准确定位后的两集合 CP_t 与 CP_{t-1} 之间有 $CP_t \subset CP_{t-1}$.设 $A(X_A, Y_A) \in P_t$,在 P_{t-1} 中寻找与 A 距离最近的点.若 $B(X_B, Y_B) \in P_{t-1}$,满足 $D(A, B) = \min\{D(A, B_i) | B_i \in P_{t-1}\}$,则将 A 点的位置纠正为 $A(X_B, Y_B)$,其中 $D(A, B_i)$ 为 A, B_i 两点之间的欧几里德距离.按上述方法减小尺度 $t-1$ 的值并按由粗糙到细节的方法进行跟踪可以得到 A 点在原始曲线上的位置 $A(X_0, Y_0)$.对 P_t 中的所有点进行上述跟踪算法则可以得到矫正的点集 CP_t .同样对集合 P_{t-1} 进行尺度从 $t-1$ 到 0 的跟踪,可以得到尺度上的点集 CP_{t-1} .因此利用尺度值的改变,我们可以得到一系列尺度为 $t, t-1, \dots$ 等的以特征点角为顶点的多边形,这些多边形在尺度值较大时,与原始曲线的差别较大,但压缩比高,适合于低比特率的形状编码;在尺度较小时,与原始形状曲线的误差小,直至完全表示原始的形状.适用于无损或失真较小的形状编码.这种特性与 MPEG-4 中分级编码的思想特别吻合^[1].需要说明的是,尺度 t 并不一定按照整数递减.

上述算法当原始曲线的某些部分属于圆弧组成时由于算法的准则是提取局部曲率绝对值最大点,所以会使形状曲线中该部分所提取的特征点数为零.为此,我们用棋盘距离对此改进,提取出逼近点^[9].这些逼近点定位跟踪后和上述 CSS 算法所求得特征点“角”按照在原始形状曲线上以弧长为参考的先后顺序排列后形成了改进 CSS 算法特征点的集合.

3.2 特征点编码算法

由于特征点的编码算法是对特征点的位置信息进行编码传输,而文^[4]中的算法是对多边形顶点的位置进行编码,所以可用文^[4]的算法为基本算法,修改后为本文的特征点编码算法.

基本算法简介如下:设编码坐标之差的绝对值的标志符号为 $FLGX, FLGY$.对 $FLGX$ 和 $FLGY$ 进行固定长度 3 比特的编码.编码之前先对坐标之差 (X_i, Y_i) 进行预处理,确定其八分圆系数及每一个坐标之差 (X_i, Y_i) 中的主分量和次分量,主分量为坐标之差 (X_i, Y_i) 的 x 和 y 分量中的较大分量而次分量为坐标之差 (X_i, Y_i) 的 x 和 y 分量中较小分量.然后按照下列步骤对坐标之差 (X_i, Y_i) 进行编码:

- (1) 按照八分圆的定义计算每一个特征点所属的八分圆

系数.

(2) 对八分圆系数按照概率分布计算码字并进行固定长度为 $3b$ 的编码.

(3) 对主分量进行编码, 若八分圆系数为 0, 3, 4, 7, 则对 X_i 相对所属八分圆的起点的幅值进行码长为 $FLGX$ 的编码; 若八分圆系数为 1, 2, 5, 6, 则对 Y_i 相对所属八分圆的起点的幅值进行码长为 $FLGY$ 的编码;

(4) 对次分量进行编码: 若八分圆系数为 0, 3, 4, 7, 则对 y 进行码长为 $\min\{FLGY+1, \log_2(|X_i|+1)\}$ 的编码; 若八分圆系数为 1, 2, 5, 6, 则对 x 相对八分圆的起点的幅值进行码长为 $\min\{FLGX+1, \log_2(|Y_i|+1)\}$ 的编码;

对上述基本算法改进如下:

基本算法中的步骤(2)、(3)、(4)用自适应算术编码方法进行改进. 把八分圆系数 0, 1, 2, 3, 4, 5, 6, 7 出现的过程看作马尔可夫过程, 则可以用其出现的概率驱动算术编码器; 同样可用主分量和次分量值出现的概率驱动算术编码器. 具体步骤参见文[6].

4 CSSAS 的帧间编码模式

4.1 基于曲率尺度空间图(CSSI)的运动估计算法

CSSI 是由下述方法得到的. 使 δ 从 0 以步长 0.1 (可选) 逐渐增大, 用(3)式在每个给定的 δ 值下对曲线进行平滑处理, 则得到 δ 值递增的光滑曲线簇, 对每条光滑曲线用(4)式计算并定位其曲率为零的点, 即拐点. 显然, 当 δ 增到足够大时, 光滑的曲线上不再有拐点存在. 以归一化弧长 μ 为横坐标, 以 δ 为纵坐标, 将光滑曲线簇上的每个拐点的位置 (μ, δ) 在平面上标出, 则得到对应于此光滑曲线簇的 CSSI.

大量实验结果表明, CSSI 有如下的特性:

(a) CSSI 反映了原始形状曲线的局部特征. CSSI 的每条曲线与原始形状曲线上的凸部或凹部一一对应. 原始形状曲线上的局部形变特征将引起 CSSI 上其相应曲线的形变.

(b) 选择原始曲线上不同的点为弧长零点, 将引起 CSSI 上曲线的平移.

(c) 原始曲线的平移不会引起 CSSI 图的变化.

在曲线的运动估计中, 曲线的变化仅由平移运动和曲线本身的部分形变组成. 为了找出参考曲线和当前曲线之间的运动矢量和形状不匹配部分, 我们按照 CSSI 上 $\delta = 0$ 所定位的曲线上的拐点, 并利用每对拐点之间的曲线对对应的原始曲线上的凸部或凹部进行分段匹配判断.

对当前帧进行扫描, 以得到的曲线上的第一个点为弧长零点 C_0 , 以逆时针方向为弧长增加的方向计算并绘出当前曲线 CSSI, 使当前曲线的 CSSI 与参考曲线的 CSSI 进行匹配运算. 若当前曲线上所选的弧长零点 C_0 与参考曲线上的弧长零点 r_0 不对应, 则必然引起 CSSI 上曲线的平移, 由匹配运算可以得出平移弧长 Δs , 反映在原始曲线上则表示当前曲线上与参考曲线上弧长零点 r_0 最对应的点为从 C_0 位置沿逆时针方向前进 Δs 的点 c'_0 . r_0 映射到当前帧中的点为 r'_0 , 则运动矢量为: $MV = r_0 c'_0$.

由 CSSI 的性质(a)可知, 与参考曲线的 CSSI 的曲线不匹

配的当前曲线的 CSSI 的曲线, 对应于当前曲线上与参考曲线不匹配的曲线部分. 只要得出 CSSI 上不匹配的曲线, 则可以得出由拐点所定位的不匹配曲线部分.

因此, 计算运动矢量与寻找形变部分的关键在于当前曲线的 CSSI 与参考曲线的 CSSI 的匹配运算. 在实验中, 我们发现, CSSI 曲线顶点的匹配将引起 CSSI 的匹配. 因此, 匹配算法进一步简化为 CSSI 曲线最大值的匹配问题, 具体算法如下:

(a) 提取 CSSI 曲线的最大值

(b) 最大值的匹配搜索算法

若相邻四条曲线满足: 匹配、不匹配、不匹配、匹配的条件, 则其对应的曲线的 8 个拐点 1-8 之中, 必有 4、5 两拐点的凸部与参考曲线的对应部分不匹配; 若相邻三条曲线满足匹配、不匹配、匹配的条件, 则其对应的曲线上的 6 个拐点 1-6 中, 必有 3、4 两拐点之间的凹部与参考曲线的对应部分不匹配.

4.2 基于弧长索引的编码算法

上述运动估计算法已经找到了当前曲线相对于参考曲线的运动矢量 MV 以及以拐点为分隔点的当前曲线与参考曲线的匹配和不匹配部分. 设经过运动估计以后,

$$C_c = \tilde{V}_r^M \ominus \tilde{V}_c^m = Y_{j \in I^M}(\tilde{V}_j^M + mv) \ominus Y_{j \in I^m}(\tilde{V}_j^m) \\ = Y_{j \in I^M} \left(Y_{j \in I^s} (C_r(i) + mv) \right) \ominus Y_{j \in I^m} \left(Y_{j \in I^s} C_r(i) \right) \quad (4)$$

这里, \tilde{V}_r^M 和 \tilde{V}_c^m 分别表示参考曲线上与当前曲线完全匹配的部分和不匹配部分, \tilde{V}_j^M 和 \tilde{V}_j^m 表示匹配和不匹配部分中的第 j 段. 对于上述表示式的第一项, 用弧长索引进行编码, 第二项则用帧间模式中的无损 CSSAS 方法进行编码, 则总的编码长度为:

$$l(C_c) = l(\tilde{V}_r^M) \ominus l(\tilde{V}_c^m) = \sum_{j \in I^M} (l(\inf I_j^s) + l(\sup I_j^s)) \\ \ominus \sum_{j \in I^m} (l(\tilde{V}_j^m) + l(mv)) \quad (5)$$

其中 $\inf I_j^s$, $\sup I_j^s$ 为序列索引为 j 的匹配段对应的弧长索引集合的下限和上限(对匹配曲线段来讲, 为其起点和终点). 在解码端, 匹配曲线的起点和终点对应的弧长通过解码得到了以后, 就可以将参考曲线上以此起点和终点为起点和终点的曲线段“拷贝”下来, 加上运动矢量 mv , 即可以恢复当前曲线上的匹配部分.

5 实验结果

实验中所使用的序列为 MPEG-4 中的标准测试序列 QCIF 格式(176*144)的“Bream”序列. 图 2 为 Bream 序列第 8 帧进行 CAE 和 CSSAS 的帧内、帧间编码时, 恢复图像的主观质量的比较. 这里纹理编码采用 MPEG-4 VM 中的纹理编码方法, 即对 VOP 以外的象素采用填充技术进行处理后, 再进行基于块的编码. 可以看出, CSSAS 与 CAE 在相同 D_n 条件下, CSSAS 主观质量优于 CAE. 图 3 和图 4 为 Bream 序列第 8 帧帧内/间编码时, 在相同 D_n 条件下, CAE 与 CSSAS 的率失真曲线. 显然, CSSAS 比 CAE 在帧内编码时, 压缩比有很大提高, 而在帧间编码 D_n 较大时, 压缩比有明显提高.

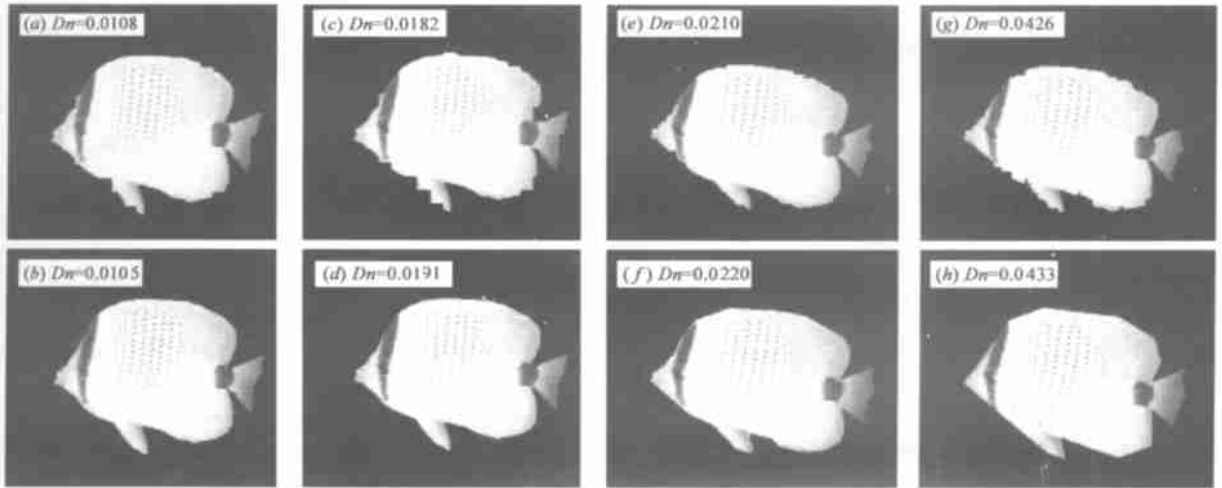


图2 Bream 序列重构图 (a) $D_n = 0.0108$ 时, CAE 帧内编解码重构图 (b) $D_n = 0.0105$ 时, CSSAS 帧内编解码重构图 (c) $D_n = 0.0182$ 时, CAE 帧内编解码重构图 (d) $D_n = 0.0191$ 时, CSSAS 帧内编解码重构图 (e) $D_n = 0.021$ 时, CAE 帧间编解码重构图 (f) $D_n = 0.022$ 时, CSSAS 帧间编解码重构图 (g) $D_n = 0.0426$ 时, CAE 帧间编解码重构图 (h) $D_n = 0.0433$ 时, CSSAS 帧间编解码重构图

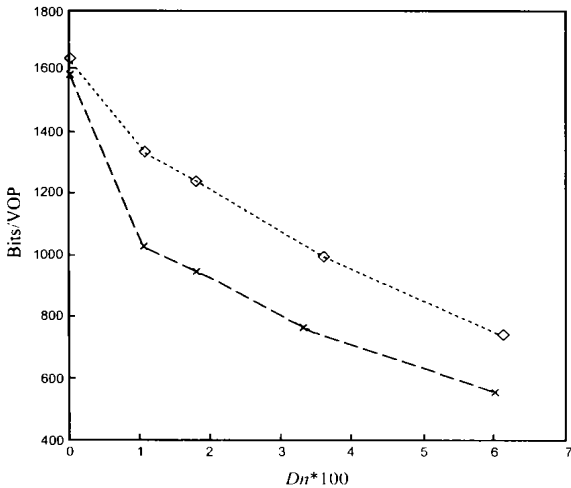


图3 帧内模式相同 D_n 条件下, CAE 与 CSSAS 对 Bream 序列编码对比特数的比较

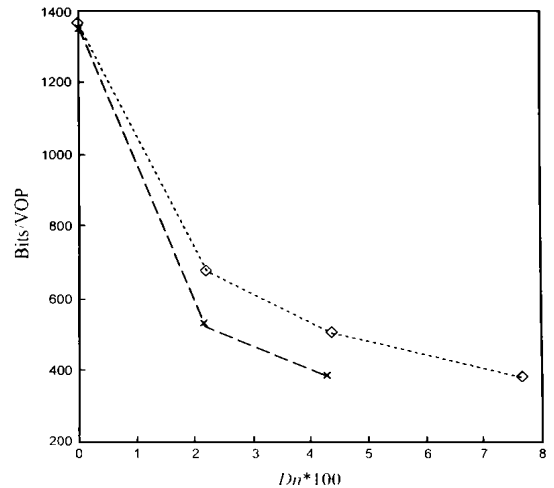


图4 帧间模式相同 D_n 条件下, CAE 与 CSSAS 对 Bream 序列编码对比特数的比较

6 结论

从以上理论分析和实验结果可以看出,本文提出的 CSSAS 算法无论从压缩比还是从重构形状的主观质量上均优于 CAE. 虽然 CAE 在压缩比特别是重构形状的主观质量上不尽如人意,但与现有的基于块的纹理编码算法的兼容性是其最大的优点. CAE 仍是 MPEG-4 中对 VO 形状信息进行编码的首选. 同时,从算法复杂度来说, CSSAS 在处理多对象视频序列时,其运算速度不如 CAE, 尚需改进.

参考文献:

[1] Coding of moving pictures and audio [S]. ISO/IEC JTC1/SC29/WG11, MPEG99/4960, Melbourne, October 1999.
 [2] N Brady, et al. Shape compression of moving objects using context based arithmetic encoding [J]. Signal Processing: Image Communication 2000, 15: 601-616.

[3] N Brady. Context based arithmetic encoding of 2D shape sequences [J]. Proc. Special Session in shape Coding, 1997, ICIP: 29-32.
 [4] K J O'Connell. Object adaptive vertex based shape coding method [J]. IEEE Trans. Circuits Syst. Video Technol., 1997, 7(1): 251-255.
 [5] F Mokhtarian, et al. Robust and efficient shape indexing through curvature scale space [A]. Proceedings of the Seventh British Machine Vision Conference [C]. 1996, Edinburgh, UK, BMVC-1: 35-42.
 [6] 石旭利, 张兆扬. 基于曲率尺度空间的 VOP 形状编码算法 [J]. 中国图像图形学报, 2002, 7(10): 1071-1076.

作者简介:

石旭利 女, 1970 年生于河南, 现为上海大学通信与信息工程学院讲师, 主要从事图像压缩编码和多媒体通信方面的研究.

张兆扬 男, 1938 年生于浙江海宁, 教授, 博士生导师, 上海大学通信与信息学科带头人, 研究方向为数字视频处理、多媒体通信和计算机视觉.