

# 基于频域的 MPEG2 码率转换技术及其码率控制策略

范晓东, 潘 健, 余松煜

(上海交通大学图象通信与信息处理研究所, 上海 200030)

**摘 要:** 本文针对 MPEG2 三种不同的预测方式, 讨论了由参考帧中各块的 DCT 系数直接求得匹配块的 DCT 系数的方法, 提出了频域内运动补偿的概念和基于频域的无偏视频码率转换技术. 其码率控制策略使用了率失真优化求得各编码帧最佳的量化参数, 并在 GBFOS 算法的基础上对率失真优化的快速算法进行了进一步简化, 在保证解的精度的前提下大大降低了计算复杂度.

**关键词:** MPEG2; 码率转换; 率失真优化

**中图分类号:** TN911.21

**文献标识码:** A

**文章编号:** 0372-2112 (2001) 01-0057-04

## A Frequency Domain MPEG-2 Drift-Free Transcoder and Its Rate Control Strategy

FAN Xiao-dong, PAN Jian, YU Song-yu

(Institute of Image Communication & Information Processing, Shanghai Jiaotong University, Shanghai 200030, China)

**Abstract:** The methods of calculating the DCT coefficients of the matching blocks from the derived DCT coefficients of the blocks in the reference frames under three different prediction modes of MPEG2 is discussed, based on which a drift-free MPEG2 transcoder that achieves motion compensation in the frequency domain is proposed. The rate-distortion optimization is applied to the rate control strategy to get the optimal quantization parameter of each frame, with its fast algorithm further simplified on the basis of GBFOS method. Our approach can lead to the solutions of satisfactory accuracy but only with a fraction of the complexity of the existing algorithms.

**Key words:** MPEG2; transcoding; rate-distortion optimization

### 1 引言

最简单的码率转换技术可以通过加大量化阶来实现视频码率的降低, 但是, 对参考帧进行的粗量化往往会对转换后的视频码流引入新的误差 (称为偏移误差), 而这种误差在一个 GOP 内是逐步积累的<sup>[1]</sup>. 为此, 文献[1]引入了运动补偿来实现无偏码率转换技术. 由于现有的运动补偿过程大多在空间域中实现, 所以该技术还使用了 IDCT/DCT 过程, 增加了系统的复杂度. 本文提出了一种在频域内实现运动补偿的方法, 该方法考虑了帧、场结构和半像素的运动补偿算法, 在保证正确进行运动补偿的前提下简化了系统.

在码率转换中的码率控制策略可以使用 TM5 算法<sup>[2]</sup>和率失真优化<sup>[3~7]</sup>方法, 后者可以在保证视频编码码率小于或等于规定值的前提下求得使量化误差最小的量化阶值. 为降低复杂度, 一般在帧一级上实现率失真优化, 而每帧图像采用固定量化阶编码<sup>[7]</sup>, 而本文则考虑了各局部图像的活动性差异对主观编码质量的影响. 率失真问题的求解可以使用 DP 及 Lagrangian 算法, 其中 DP 算法得到全局最优解, 而 Lagrangian 算法在 R-D 曲线的凸包络线上求得局部最优解, 但该方法

应用了迭代过程来求得最佳的 Lagrangian 乘数值<sup>[4]</sup>, 计算过程仍然较为复杂. 为此, P. A. Chou 等提出了 GBFOS 算法<sup>[9]</sup>, 通过树的裁剪过程来求得凸包络线. 而本文在 GBFOS 算法的基础上进行了进一步简化, 提出了一种基于梯度的搜索方法, 在保证解的精度同时大大降低了计算复杂度.

### 2 码率转换技术及频域内运动补偿算法

码率转换技术的详细讨论可以见文[1]及文[3], 本文不再赘述, 只在此列出算法框图. 图中, 虚线框中的部分为空间域中的运动补偿过程, 而 DCT 与 IDCT 模块则实现空间域与

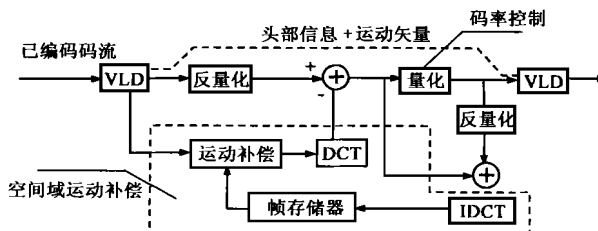


图1 码率转换算法框图

频域间的转换. 为简化这一转换过程, 本文提出一种于频域内的运动补偿过程.

## 2.1 帧预测方式的运动补偿

MPEG-2 中的匹配块  $M$  可由附近的四个块  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$  的组合来表示:

$$M = \sum_{i=1}^4 U_i \times A_i \times V_i \quad (1)$$

其中,

$$U_1 = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}, V_1 = \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix}, U_2 = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}, V_2 = \begin{bmatrix} 0 & I_{8-w} \\ 0 & 0 \end{bmatrix}$$

$$U_3 = \begin{bmatrix} 0 & 0 \\ I_{8-h} & 0 \end{bmatrix}, V_3 = \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix}, U_4 = \begin{bmatrix} 0 & 0 \\ I_{8-h} & 0 \end{bmatrix}, V_4 = \begin{bmatrix} 0 & I_{8-w} \\ 0 & 0 \end{bmatrix} \quad (2)$$

$h$  和  $w$  的意义见图 2:

$$\text{易得, } DCT(M) = \sum_{i=1}^4 DCT(U_i) \times DCT(A_i) \times DCT(V_i) \quad (3)$$

而  $DCT(A_i)$  可以直接从输入码流中获得,  $DCT(U_i)$  和  $DCT(V_i)$  可以事先算得, 所以完全能够在 DCT 域内计算匹配块的 DCT 系数, 从而完成运动补偿过程.

## 2.2 场预测方式的运动补偿

在进行场方式的运动估计时, 将预测帧中宏块  $M$  分为  $16 \times 8$  的顶场  $M_1$  和底场  $M_2$ , 同时将参考帧  $R$  也分为顶场  $R_1$  和底场  $R_2$ , 分别使用  $R_1$  和  $R_2$  对顶场  $M_1$  和底场  $M_2$  进行运动估计, 因此场预测宏块有两个运动矢量. 如上节, 令匹配块  $M$  附近的四个块  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$  的隔行图像方式用块  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$  表示, 其中,  $A_1$ 、 $A_2$  构成顶场的  $8 \times 16$  像素, 而  $A_3$ 、 $A_4$  构成底场的  $8 \times 16$  像素. 两者关系可用下式表示:

$$A_1 = P_1 \times A_1 + P_2 \times A_3, A_2 = P_1 \times A_2 + P_2 \times A_4 \quad (4)$$

$$A_3 = P_3 \times A_1 + P_4 \times A_3, A_4 = P_3 \times A_2 + P_4 \times A_4$$

其中, 矩阵  $P_1 \sim P_4$  的定义见附录 1. 对上式两边求 DCT, 便可以得到将逐行图像表示成隔行图像后 DCT 系数间的关系, 然后, 通过式 (3) 得到顶场  $M_1$  和底场  $M_2$  的匹配块  $M_1$  和  $M_2$  的 DCT 系数. 下面, 要从实际编码使用的匹配块  $M$  的顶场  $M_1$  和底场  $M_2$  的 DCT 系数恢复出  $M$  的 DCT 系数, 即求式 (4) 所描述的变换的逆, 其结果如下:

$$A_1 = Q_1 \times A_1 + Q_2 \times A_3, A_2 = Q_1 \times A_2 + Q_2 \times A_4 \quad (5)$$

$$A_3 = Q_3 \times A_1 + Q_4 \times A_3, A_4 = Q_3 \times A_2 + Q_4 \times A_4$$

其中, 矩阵  $Q_1 \sim Q_4$  的定义见附录 2.

## 2.3 半像素预测方式的运动补偿

在 MPEG-2 中, 半像素预测方式所得的匹配块由原来各块通过线性内插而得, 如图 3 中, 点  $P_3$  的灰度值可以用点  $a_{1,1}$  和  $a_{1,2}$  的灰度值来表示:  $P_3 = (a_{1,1} + a_{1,2} + 1)/2$ . 因此, 若运动矢量指向  $P_3$ , 则此时对应的匹配块相当于先将原来在参考帧中相邻的四个块中的所有像素同时水平右移半个像素, 然后再利用式 (1) 求得. 下面, 仍然利用矩阵运算来表示右移过程:

$$A_{1,h} = A_1 \times Q_{h1} + A_2 \times Q_{h2}, A_{2,h} = A_2 \times (Q_{h1} + Q_{h3}) \quad (6)$$

$$A_{3,h} = A_3 \times Q_{h1} + A_4 \times Q_{h2}, A_{4,h} = A_4 \times (Q_{h1} + Q_{h3})$$

其中, 矩阵  $Q_{h1} \sim Q_{h3}$  的定义见附录 3. 同样, 若运动矢量指向  $P_2$ , 相当于下移半个像素, 用式 (7) 表示:

$$A_{1,v} = Q_{v1} \times A_1 + Q_{v2} \times A_2, A_{2,v} = Q_{v1} \times A_2 + Q_{v2} \times A_4 \quad (7)$$

$$A_{3,v} = (Q_{v1} + Q_{v2}) \times A_3, A_{4,v} = (Q_{v1} + Q_{v2}) \times A_4$$

其中, 矩阵  $Q_{v1} \sim Q_{v3}$  的定义见附录 4. 而若运动矢量指向  $P_1$ , 则可以看作右移和下移的组合. 通过式 (6) 和式 (7), 便可以由原有各整像素块的 DCT 系数求得右移或下移各半像素块的 DCT 系数, 进而通过式 (3) 得到半像素预测方式的匹配块的 DCT 系数.

$$\begin{matrix} a_{0,0} & a_{0,1} & a_{0,2} \\ *P_1 & & \\ a_{1,0} & a_{1,1} & a_{1,2} \\ *P_2 & *P_3 & \\ a_{2,0} & a_{2,1} & a_{2,2} \end{matrix}$$

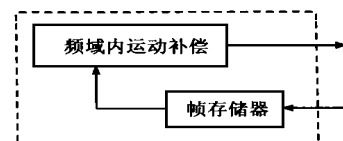


图 3 半像素运动矢量块像素的重建

图 4 频域内运动补偿过程

通过 2.1 ~ 2.3 的讨论, 可以由参考帧中各块的 DCT 系数求得相应匹配块的 DCT 系数, 因此, 图 1 中的线性框可以简化为图 4 中的情形, 而省略 IDCT/DCT 过程.

## 3 基于率失真优化的码率控制算法

在图 1 所示的码率转换器框图中, 二次量化过程的量化阶的选择即由码率控制算法实现. 一般而言, 随着量化阶的减小, 量化失真将会相应减小, 但同时带来了编码码率的提升. 因此, 率失真优化的实质便是在保证编码码率小于或等于规定值的前提下, 通过适当地选取量化阶, 实现量化失真最小. 用数学表示, 即为:

$$\text{求: } \min D_i(q_i), \text{ 且满足 } R_i(q_i) \leq R_T, R_T \text{ 为规定码率} \quad (8)$$

其中,  $R_i(q_i)$  为该编码帧的编码比特数,  $D_i(q_i)$  为量化后重建图像相对于原图的 MSE 值, 即量化失真. 考虑到计算复杂度, 一般可以在帧一级上实现率失真优化, 即对一帧内各宏块使用同一个量化阶进行编码. 然而, 图像编码的主观质量不但与量化阶有关, 还和图像本身的分布特别是活动性 (即图像的信源方差) 直接相关<sup>[8]</sup>. 因此本文采用类似于 TM5 的方法, 量化阶由两部分相乘而得, 一个乘数为量化参数, 它在一帧图像内变化很小, 另一个乘数为各宏块的归一化活动性系数 (计算方法见文献 [2]), 它反映了局部图像间的差异. 率失真优化通过对量化参数的选择来调整各帧图像的量化水准, 最终各宏块的量化阶仍由量化参数与归一化活动性系数相乘而得.

### 3.1 率失真优化问题的描述

本文提出的码率控制算法针对恒定码率 (CBR) 编码模式, 以一个 GOP 为单位来进行率失真优化, 对  $P$ 、 $B$  帧的量化阶选择同时考虑到其参考帧的量化阶的影响. 在此基础上, 还作了进一步的简化, 即使用同一类型的已编码或当前编码帧的率失真数据来代替该 GOP 内未编码帧的率失真数据, 则率失真优化过程可由下式描述:

$$\text{求: } \min D, \text{ 且满足: } R \leq R_{\text{GOP}}$$

其中,

$$D = [N_I \times D(q_I) + N_P \times D(q_I, q_P) + N_B \times D(q_I, q_P, q_B)] \quad (9)$$

$$R = [N_I \times R(q_I) + N_P \times R(q_I, q_P) + N_B \times R(q_I, q_P, q_B)] \quad (10)$$

在上式中,  $R_{\text{COP}}$  为该 COP 内剩余的比特数, 其初始值为  $(R_T / \text{帧率}) \times \text{COP 长度}$ , 每帧编码完毕后,  $R_{\text{COP}} = R_{\text{COP}} - \text{Rate}(\text{Rate 为该帧编码使用的比特数})$ .  $D$  和  $R$  为使用当前量化参数值所对应的量化失真度和编码比特数,  $N_I, N_P, N_B$  为当前 COP 内剩余 I、P、B 帧的个数.

### 3.2 率失真优化问题的求解

首先介绍 GBFOS 算法. 该算法通过求得 R-D 曲线的凸包络线来获得位于该包络线上的率失真优化问题的局部最优解<sup>[9]</sup>. 整个算法过程对应于树的裁剪过程, 该算法由两部分组成, 即最小斜率值节点的搜索过程及各节点斜率值的修正过程. 其中最小斜率值节点的搜索过程用于保证裁剪过程得到的 R-D 值所对应的点严格位于 R-D 曲线上的凸包络线上, 每经过一次裁剪后, 由于当前的量化失真度值和编码比特数已经发生了变化, 还必须进行各节点斜率值的修正过程.

GBFOS 算法与 Lagrangian 算法相比运算速度有了较大的提高, 但是, 由于最小斜率值节点的搜索必须针对整棵树中的所有节点进行, 且每次裁剪过程结束后都必须进行各节点斜率值的修正过程, 所以计算量仍然较大. 为此, 本文在此基础上作了进一步的简化. 其算法过程如下:

- (1) 令 I、P、B 三种帧类型图像量化参数初始值为 1.
- (2) 分别计算 I、P、B 三种帧类型图像量化参数增加 1 所引起的量化失真度的增加值与编码比特数的减少值的比率, 并对量化参数进行离散化, 量化参数取 1~42 之间的整数.
- (3) 选取最小的比率值所对应的帧类型的量化参数加 1, 然后使用式 (10) 计算量化失真度和编码比特数. 若编码比特数小于或等于  $R_{\text{COP}}$ , 则结束求解算法, 否则转至步骤 (2).

从以上的算法过程不难看出, 本算法与 GBFOS 算法相比, 其最小斜率值节点的搜索及各节点斜率值的修正过程所对应的目标集合中的节点大大减少, 从当前树中的所有节点减少为三个节点, 相应计算量也大大降低. 应该指出的是, 本算法得到的 R-D 值并不严格位于 R-D 曲线的凸包络线上, 但实验证明, 本算法的优化结果在多数场合仍然较为令人满意.

使用上述算法来求得式 (9) 描述的条件最值问题的解, 即可得到对当前帧进行量化的量化参数, 同时将该量化参数与相应宏块的归一化活动性系数相乘, 就可以求得对各个宏块进行量化的量化阶.

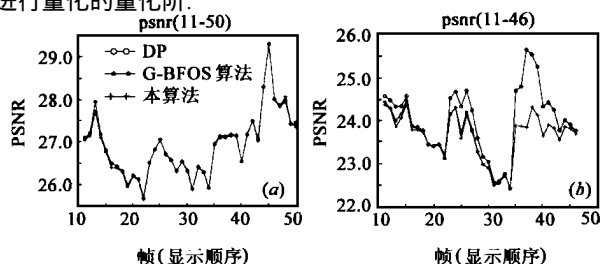


图5 使用三种搜索算法的重建图像的 PSNR 值  
(a) Football 序列; (b) Football & Garden 序列

## 4 实验结果

使用图 1 中所述的结构 (其中, 虚线框中对应的运动补偿过程已由图 4 中所示的频域内运动补偿代替), 利用软件实现了 MPEG-2 的码率转换器. 其中, 码率控制算法使用了本文提出的基于率失真优化的方法. 图 5 (a) 和图 5 (b) 比较了在对 Football 序列和 Flower&Garden 进行从 4Mbps/s 到 2Mbps/s 的码率转换中, 使用本文的率失真优化求解算法和 DP 算法及 GBFOS 算法所得的图像编码质量, 其中编码质量的衡量采用了峰值信噪比 PSNR 值. 表 1 进一步比较了使用这三种搜索方法进行码率控制及 TM5 算法得到的平均峰值信噪比值, 从中可以看出, 使用本算法在大大降低了算法复杂度的前提下仍能得到较为令人满意的局部最优解, 其性能普遍好于 TM5 算法过程. 由于本算法通过减少 GBFOS 算法的最小斜率值节点的搜索范围和斜率值修正过程的计算范围, 来达到降低算法复杂度的目的, 所以表 2 中列出了分别采用 GBFOS 算法和本算法计算过程中进行最小斜率值搜索和斜率值修正所处理的节点个数, 从中可以证实本算法的确大大降低了 GBFOS 算法的复杂度.

表 1 四种码率控制算法的编码平均 PSNR 值

码率控制算法	平均峰值信噪比值 (dB)	
	Football 序列	Flower&Garden 序列
DP 算法	26.942dB	24.011dB
GBFOS 算法	26.943dB	23.899dB
本算法	26.937dB	23.657dB
TM5 算法	26.908dB	23.271dB

表 2 两种率失真优化快速算法的计算复杂度比较

	Football 序列		Flower & Garden 序列	
	平均每帧最小斜率值搜索过程处理节点数	平均每帧斜率值修正过程处理节点数	平均每帧最小斜率值搜索过程处理节点数	平均每帧斜率值修正过程处理节点数
GBFOS 算法	10194065	27700584	11857910	30017210
本算法	643	2284	829	3028

## 5 结论

本文提出了基于频域内运动补偿的无偏 MPEG2 码率转换技术及率失真意义下最优的码率控制策略. 实验结果表明, 使用本文提出的算法来求解 R-D 优化问题, 不仅可以大大降低计算复杂度, 且在多数场合能够获得与 GBFOS 算法相当的局部最优解.

## 参考文献:

- [1] P. A. Assuncao, M. Ghanbari. Post-processing of MPEG-2 coded video for transmission at lower bit rates [J]. IEEE Int. Conf. ASSP, ICASSP '96, Atlanta, USA, May 1996, 4:1998 - 2001.
- [2] Test Model 5 [S]. ISO/IEC JTC1/SC29/WG11, MPEG03/457.
- [3] P. A. Assuncao, M. Ghanbari. Optimal bit rate conversion of MPEG-2 video bit streams [J]. Electronics Letters, April, 1997, 33 (8): 675 - 677.
- [4] H. Sun, W. Kwok, M. Chien, et al. MPEG coding performance improve-

ment by jointly optimizing coding mode decisions and rate control [J]. IEEE Trans. CSVT, June 1997, 7(3): 449 - 458.

- [5] J. Lee, B. W. Dickinson. Rate-distortion optimized frame type selection for MPEG encoding [J]. IEEE Trans. CSVT, June 1997, 7(3): 501 - 509.
- [6] K. Remchandran, M. Vetterli. Best wavelet packet bases in a rate-distortion sense [J]. IEEE Trans. image processing, April 1993, 2(2): 160 - 174.
- [7] L. J. Lin, A. Ortega. Bit rate control using piecewise approximated rate-distortion characteristics [J]. IEEE Trans. CSVT, August 1998, 8(4): 446 - 458.
- [8] Y.-S. Saw, P. M. Grant, J. M. Hannah, et al. Source model for transform video coder and its application-part I: fundamental theory [J]. IEEE Trans. CSVT, April 1997, 7(2): 287 - 297.
- [9] P. A. Chou, T. Lookabaugh, R. M. Gray. Optimal pruning with applications to tree-structured source coding and modeling [J]. IEEE Trans. information theory, March 1989, 35(2): 299 - 315.

#### 作者简介:



**范晓东** 1975 年生, 2000 年获上海交通大学硕士学位, 现在美国约翰·霍普金斯大学就读博士学位, 目前主要从事视频编码和图像处理方面的研究。



**潘 健** 1972 年生, 1999 年获上海交通大学博士学位, 现在上海广电股份有限公司工作, 目前主要从事数字电视方面的研究和开发工作。

#### 附录 1

矩阵  $P_1 - P_4$  的定义如下:

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, P_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$P_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, P_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

#### 附录 2

矩阵  $Q_1 - Q_4$  的定义如下:

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, Q_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, Q_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

#### 附录 3

矩阵  $Q_{h1} - Q_{h3}$  的定义如下:

$$Q_{h1} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}, Q_{h2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, Q_{h3} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$

#### 附录 4

矩阵  $Q_{v1} - Q_{v3}$  的定义如下:

$$Q_{v1} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}, Q_{v2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, Q_{v3} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$