第九章 数字视频转换

本章内容

一、去隔行

1. 隔行扫描
2. 去隔行方法

二、帧率变换

三、空间分辨率转换

1. 隔行扫描 --- 基本概念

• 隔行扫描方式是将一帧电视图像分成2场进行扫描，第一场完成光栅的奇数行，第二场完成光栅的偶数行。

• 如PAL制每秒传送25帧图像，那么每秒扫描50场，即场频为50Hz。

• 隔行扫描方式可以有效降低带宽的占用，因此为三大模拟彩色电视系统PAL、NTSC和SECAM采用，并且在高清电视广播标准中也仍然保留了隔行信号格式（1080i）。

1. 隔行扫描 --- 存在的问题

1. 行间闪烁（flicker）

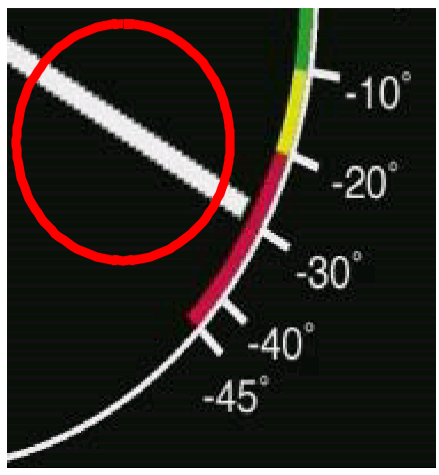
行间闪烁主要和细小的水平边缘有关，即图像中一条沿着水平方向的边，如门窗的上下边。因为隔行信号的特性，一个精细的水平边缘只能在相邻的两场中的一场显示，而在另一场中就消失了，这样这个精细的水平边缘就相当于以帧频（30Hz或者25Hz）在进行显示。观察者可以明显的觉察到这种异常的显示而如果这种边缘有垂直运动的话，这种闪烁也会随着垂直运动而移动，严重影响观看效果。

1. 隔行扫描 --- 存在的问题

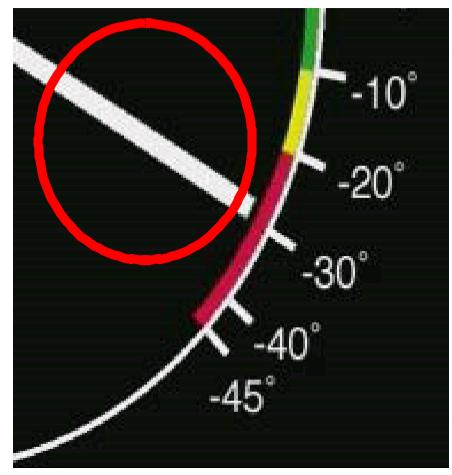
2. 锯齿（jaggies）现象

锯齿现象是隔行信号显示的一个普遍的现象，主要出现在静止或运动物体的对角线斜边处。对于静止或运动较少的图像，较低的垂直分辨率会引起台阶效应。当图像包含运动时，情况就变得更糟糕了。隔行图像的边缘每场只有一半的行数进行更新，这样就导致损失了边缘的细节。进一步，当边缘运动起来之后，下一场的边缘就会出现相对的位移，锯齿装的边缘就这样从一场运动到下一场，产生连续的锯齿现象。

1. 隔行扫描 --- 存在的问题



（1）有锯齿



（2）无锯齿

1. 隔行扫描 --- 存在的问题

3. 爬行（crawling）

爬行是一种隔行信号产生的虚像，它主要出现在没有运动和较少细节的图像中。当观察者离显示设备比较近的时候，这种现象更为明显。因为每条隔行扫描线每帧只更新一次，人眼有时会盯住一组扫描线。在下一场，新的扫描线会出现在旧的扫描线之间，看起来好像原来的那组扫描线向上或向下移动到了一个新的位置上。再下一场，这一组扫描线又出现在原来的位置。最终的观看效果就是一组扫描线快速的向上或向下移动，从而引起爬行现象。

1. 隔行扫描 --- 存在的问题

4. 羽化（feather）现象

羽化现象通常会出现在运动物体的轮廓处，主要是由于运动检测失误导致的。当运动的物体被误判断为静止，为了复原静止图像就会采用场间插值处理，那么在运动物体的轮廓边缘处，两个存在相对位移图像直接拼合，将导致图像出现虚影。

（1）有羽化

（2）无羽化

1. 隔行扫描

• 对于目前的逐行扫描显示设备（如计算机）来说，需要将输入的隔行扫描视频进行扫描格式转换。

• 去隔行算法的优劣直接影响到最终显示图像的质量。

• 不同格式的视频进行转换，通常都要先进行去隔行处理。如标清转高清。

2. 去隔行算法

• 去隔行插值技术的主要思想是利用时域和空域中的临近像素点对待处理像素点进行估算。

• 分为三大类：

（1） 线性方法

（2） 非线性方法

（3） 基于运动补偿的去隔行算法

2. 去隔行算法 --- 线性方法

• 线性方法是去隔行处理中比较直观的方法，也是较早被采用的方法。

•为了方便说明，不妨假定偶数时刻的场为顶场，奇数时刻的场为底场。

$$P_{\mathrm{p}}(x,y,t)=\begin{cases}P_{\mathrm{i}}(x,y,t), & \text{if } \ y \bmod 2 = t \bmod 2,\\ \sum P_{\mathrm{i}}(x,y+m\Delta y,t+n\Delta t)h(m,n) & \text{otherwise.}\end{cases}$$

其中，x和y分别表示二维空间坐标（x表示水平方向，y表示垂直方向），t表示当前场所对应的时间点，下标i表示隔行扫描和p表示逐行扫描，h (m, n)为滤波器系数。

2. 去隔行算法 --- 线性方法

• 线性方法可以分为空间插值处理、时间插值处理以及时空混合滤波处理。

• 空间插值处理也称为场内（intra-field）插值处理，通常是采用场内垂直滤波器进行垂直插值计算。

•理想的垂直滤波器应该是半带宽的低通滤波器，但这种滤波器需要无限的空间长度，实际中需要限定一个长度范围，简单的方法是采用“行平均法”，即根据上下行采样点数据的平均计算该未知采样点。

2. 去隔行算法 - 线性方法 – 空间插值

• 线性方法可以分为空间插值处理、时间插值处理以及时空混合滤波处理。

• 空间插值处理也称为场内（intra-field）插值处理，通常是采用场内垂直滤波器进行垂直插值计算。

•理想的垂直滤波器应该是半带宽的低通滤波器，但这种滤波器需要无限的空间长度，实际中需要限定一个长度范围，简单的方法是采用“行平均法”，即根据上下行采样点数据的平均计算该未知采样点。

2. 去隔行算法 - 线性方法 – 空间插值

• 行平均法等效的滤波器：

$$h(m,n)=\begin{cases}1/2, & \text{if } (m,n)=(1,0),(-1,0),\\ 0, & \text{otherwise.}\end{cases}$$

• 该滤波器的频谱图：

在垂直方向上，有一定的低通滤波效果，但与理想的半带宽低通滤波器有较大的偏差。

2. 去隔行算法 - 线性方法 – 空间插值

• 为了提高垂直滤波器的滤波性能，可以适当增加滤波器的长度，以使频率响应更加接近理想曲线。可以将垂直插值的参考行扩展到4行：

$$h(m,n)=\begin{cases}7/16, & \text{if } (m,n)=(1,0),(-1,0),\\ 1/16 & \text{if } (m,n)=(3,0),(-3,0),\\ 0, & \text{otherwise.}\end{cases}$$

• 该滤波器的频谱图如右图

2. 去隔行算法 - 线性方法 – 时间插值

• 时间域插值也称为场间（inter-field）插值。

• 因为在前后相邻两场中，对应待插值行的位置都存在原始行信息。可以采用对应行复制的方法，通常称之为“场合并法”。

• 假设当前第t场与第t-1场合并处理。其等效的滤波器表示为：

$$h(m,n)=\begin{cases}1, & \text{if } (m,n)=(0,-1)\\ 0, & \text{otherwise.}\end{cases}$$

2. 去隔行算法 - 线性方法 – 时间插值

• 场合并法应用到运动较为剧烈的视频序列时，会产生很明显的羽化现象。

•为了提升处理效果，也可以采用对称结构的滤波器。这种方法利用前后两场对应行的平均值来计算当前场的数据，通常称为“场平均法”。

• 场平均法等效滤波器为：

$$h(m,n)=\begin{cases}1/2, & \text{if } (m,n)=(0,1),(0,-1),\\ 0, & \text{otherwise.}\end{cases}$$

2. 去隔行算法 - 线性方法 – 时间插值

• 场合并法应用到运动较为剧烈的视频序列时，会产生很明显的羽化现象。

•为了提升处理效果，也可以采用对称结构的滤波器。这种方法利用前后两场对应行的平均值来计算当前场的数据，通常称为“场平均法”。

• 场平均法等效滤波器为：

$$h(m,n)=\begin{cases}1/2, & \text{if } (m,n)=(0,1),(0,-1),\\ 0, & \text{otherwise.}\end{cases}$$

2. 去隔行算法 - 线性方法 – 时空混合插值

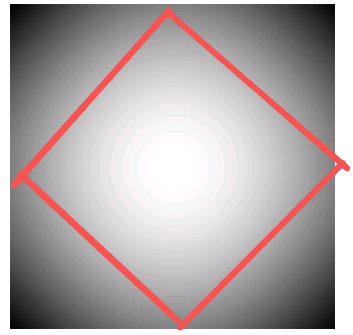
• 为了结合时间域和空间域（垂直）插值两者的优点，更为有效的方法是同时使用垂直滤波器和时间滤波器。

• 举例来说，可以采用上、下行和前、后场数据的平均值进行计算，通常称之为“十字滤波法”，其等效的滤波器表达式为：

$$h(m,n)=\begin{cases}1/4, & \text{if } (m,n)=(1,0),(-1,0),(0,1),(0,-1),\\ 0, & \text{otherwise.}\end{cases}$$

2. 去隔行算法 - 线性方法 – 时空混合插值

• 十字滤波法频谱如下图，可以看出这种滤波器稍类似于理想的菱形频谱。



2. 去隔行算法 – 非线性方法

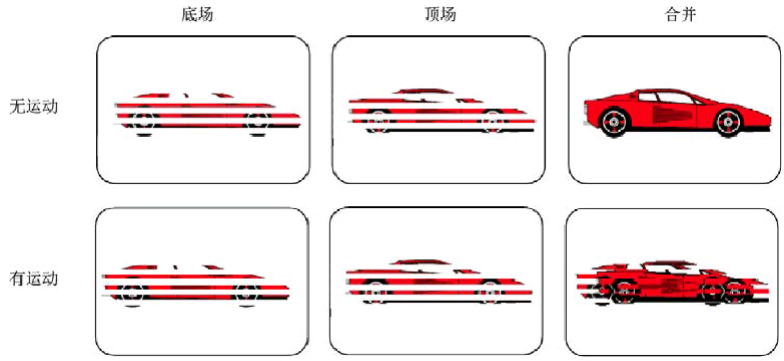
• 运动自适应方法的基本思想是将图像去隔行处理分为针对运动图像和针对静止图像的处理。

•对于静止的图像采用时间域插值可以较好的还原逐行图像。对于运动的图像，前、后场的信息就不足以用于估计当前场的数据，应该只采用当前场数据本身的数据在空间域内进行计算，也称为场内插值处理，以避免出现严重的羽化现象。

2. 去隔行算法 – 非线性方法

• 运动自适应方法的关键问题：

判断图像在视频中处于运动状态或是静止状态，以及确定运动的强度



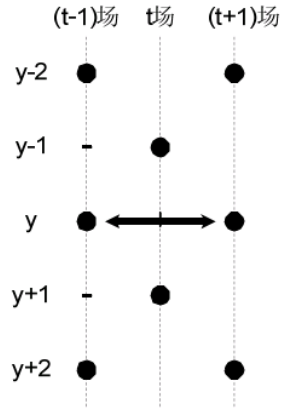
2. 去隔行算法 – 非线性方法

•基本的运动检测方法是直接计算相邻帧的对应位置像素的差值。如果帧间差值小于预先设定的阈值，该采样点被判断为静止，否则被认为是运动。

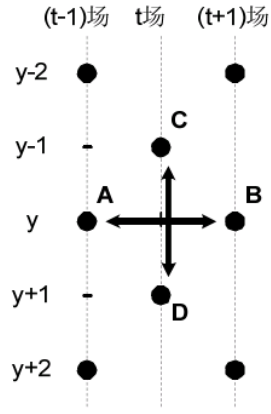
•一种改进的运动检测方法，同时考虑了时间域和空间域的信息来降低运动检测器的误判。该方法如下：

1. 如果 $|A-B|>T_1$，表示有运动；
2. 如果 $|A-B|<T_1$，而且 $\left|\frac{A+B}{2}-\frac{C+D}{2}\right|>T_2$，也表示有运动；
3. 如果 $|A-B|<T_1$，而且 $\left|\frac{A+B}{2}-\frac{C+D}{2}\right|<T_2$，表示静止。

2. 去隔行算法 – 非线性方法



（1）基本运动检测



（2）改进的运动检测方法

2. 去隔行算法 – 非线性方法

•由于在运动检测过程中会存在一定的干扰因素影响运动状态的确定，因此运动检测的输出结果通常表示运动状态的可能性。

•在最终运算时，通常将运动检测结果表示为一个权重值M，对场间和场内两种插值处理方法的结果进行加权，得到最终的插值结果

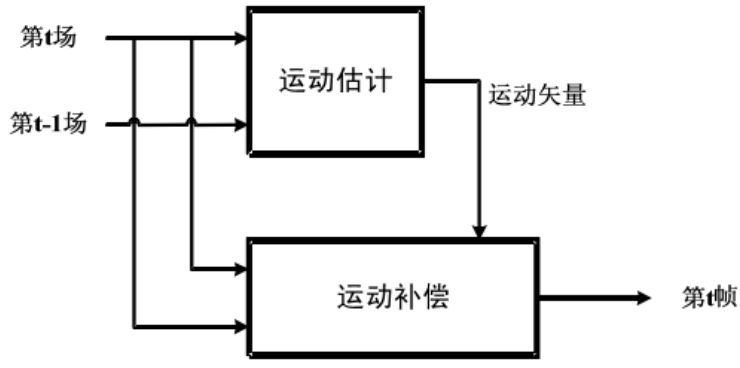
$$P_{MA}(x,y,t)=\begin{cases}P_i(x,y,t), & \text{if } y \bmod 2 = t \bmod 2,\\ M\times P_{intra}(x,y,t)+(1-M)\times P_{inter}(x,y,t) & \text{otherwise.}\end{cases}$$

2. 去隔行算法 – 基于运动补偿的去隔行算法

•基于运动检测的运动自适应算法可以得到较好的图像质量，但是其中的运动信息仍是一个标量，即只能得到运动状态的强度数值，而实际中的运动是一个矢量信息，除了大小强度之外，还存在方向信息。

•基于运动补偿技术的去隔行算法其主要思想是通过运动估计找出待处理像素点在相邻场中最匹配的数据信息，从而确定目标的运动矢量，以此在对应运动轨迹上进行插值，最终得到更加精确的计算结果。

2. 去隔行算法 – 基于运动补偿的去隔行算法



本章内容

一、去隔行

二、帧率变换

三、空间分辨率转换

二、帧率变换 --- 概述

• 帧率变换包括帧率提升和帧率降低。

• 帧率提升：在原有的视频帧之间插值出新帧，以提高每秒钟播放的视频帧数。

• 应用场合：

1）不同制式的电视、电影之间帧率转换

2）慢镜头播放

3）视觉效果增强：例如，100Hz的电视通过帧频提升去除普通电视的闪烁效应

二、帧率变换 --- 帧率提升的方法

1. 帧复制法
2. 帧平均法
3. 运动补偿法

二、帧率变换 --- 帧重复法、帧平均法

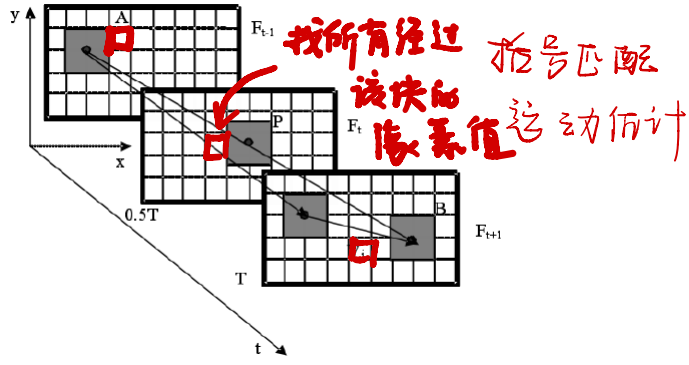
- 帧重复法：完全复制时间轴上最近的相邻帧作为新帧

瞬移

- 帧平均法：对时间轴上的相邻若干帧进行线性组合，得到新的帧。
- 帧平均法分2种：一种直接取相邻两帧的平均值，第二种则以和相邻两帧的距离作为权值进行线性运算。
- 帧重复法和帧平均法，没有考虑视频的运动特性，会造成模糊或运动边缘锯齿效果；对于视频中的运动对象，内插出的帧该运动对象没有处在正确的位置，会产生错位现象。

二、帧率变换 --- 运动补偿法

• 运动补偿法：对原始帧进行运动估计，得到运动矢量，运动补偿插值得到待插帧。



二、帧率变换 --- 运动补偿法

• 帧率上变换中的运动估计不仅涉及到前一帧和后一帧，还涉及到两帧中间的待插帧。如果将视频编码中的运动估计算法直接用在帧率上变换中，待插帧中的某些像素可能被多条运动轨迹或者没有运动轨迹通过，前者被称为“重叠”问题，后者被称为“空洞”问题。

• 双向运动估计：该算法固定待插帧当前块的位置，在前一帧和后一帧搜索范围内以某种策略寻找最佳匹配块，相应的运动矢量作为待插帧当前块的运动矢量，即为双向运动估计算法。

二、帧率变换 --- 运动补偿法

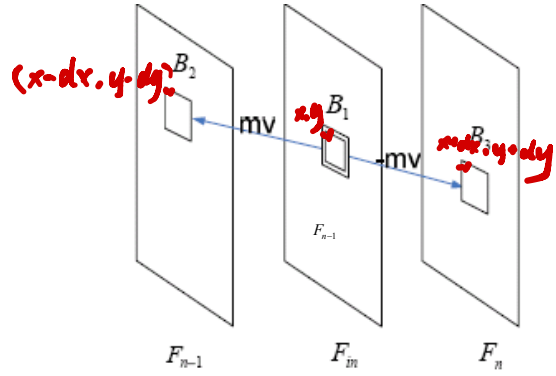


Figure 3. Overlapped block bi-directional motion estimation.

说明：上图中 F_{n-1}和 F_n 为原本有的帧，现在要在这两帧之间内插出一帧 F_{in} ，以实现2倍帧率上变换。

二、帧率变换 --- 运动补偿法

我们把 F_{in} 分成 8*8 的块，对每一个 8*8 的块都要进行双向运动估计；当找到前一帧 F_{n-1} 和后一帧 F_n 中最匹配的块后，进行运动补偿得到 F_{in} 这个 8*8 的块的像素的值。

假设 F_{in} 中当前这个需要找到的 8*8 的块的左上角坐标为 (x, y)，，则双向运动估计公式：

$$SAD = \sum_{i=0}^{7}\sum_{j=0}^{7}\left|F_n(x+d_x+i, y+d_y+j) - F_{n-1}(x-d_x+i, y-d_y+j)\right|$$

二、帧率变换 --- 运动补偿法

设运动估计搜索范围为 $(-16,16)$，则 d_x 和 d_y 的范围在 $(-16,16)$ 之间。在 $(-16,16)\times(-16,16)$ 范围内找到一个最小的 SAD，则这时的 d_x 和 d_y 就是运动矢量 mv_x 和 mv_y。

接下来完成当前块的运动补偿：

$$F_{in}(x+i,y+j)$$
$$=\frac{1}{2}\left(F_n(x+mv_x+i,y+mv_y+j)+F_{n-1}(x-mv_x+i,y-mv_y+j)\right)$$
$$0\le i\le 7,\quad 0\le j\le 7$$

给其中一个

本章内容

一、去隔行

二、帧率变换

三、空间分辨率转换

三、空间分辨率转换

- 最近邻插值
- 双线性插值
- 三次线性插值