

附录 E  
(规范性附录)  
补充增强信息(SEI)

## E.1 补充增强信息(SEI)语法

## E.1.1 补充增强信息(SEI)负载语法

SEI 负载语法见表 E.1。

表 E.1 SEI 负载语法表

描述符
sci_payload(PayloadType, PayloadSize) {
if(PayloadType == 0)
buffering_period(PayloadSize)
else if(PayloadType == 1)
pic_timing(PayloadSize)
else if(PayloadType == 2)
spherical_pano_video_parameter_set(PayloadSize)
else
reserved_sei_message(PayloadSize)
if(! byte_aligned()) {
bit_equal_to_one /* 应等于 1 */
while(! byte_aligned())
bit_equal_to_zero /* 应等于 0 */
}
}

## E.1.2 缓存周期 SEI 消息语法

缓存周期 SEI 消息语法见表 E.2。

表 E.2 缓存周期 SEI 消息语法表

描述符
buffering_period(PayloadSize) {
if(HrdBpPresentFlag) {
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++)
initial_cpb_removal_delay[SchedSelIdx]
}
}

### E.1.3 图像定时 SEI 消息语法

图像定时 SEI 消息语法见表 E.3。

表 E.3 图像定时 SEI 消息语法表

pic_timing(PayloadSize) {	描述符
if(CpbDpbDelaysPresentFlag) {	
cpb_removal_delay	u(v)
dpb_output_delay	u(v)
}	
}	

### E.1.4 全景视频参数 SEI 消息语法

全景视频参数 SEI 消息语法见表 E.4。

表 E.4 全景视频参数 SEI 消息语法表

spherical_pano_video_parameter_set(PayloadSize){	描述符
spherical_pano_flag	u(1)
if(spherical_pano_flag) {	
projection_type	u(2)
if(projection_type == 0){	
arrangement_raster_mode	u(10)
arrangement_matrix_mode	u(2)
}	
fullpano_width_inpixel	u(16)
fullpano_height_inpixel	u(16)
croppedarea_left_inpixel	u(16)
croppedarea_top_inpixel	u(16)
stereo_mode	u(2)
horizontal_angle_range	se(v)
pitch_angle_up_limit	se(v)
pitch_angle_down_limit	se(v)
initial_viewazimuth_angle	se(v)
initial_view_pitch_angle	se(v)
initial_view_roll_angle	se(v)
display_width_inpixels	u(8)
display_height_inpixels	u(8)

表 E.4 (续)

spherical_pano_video_parameter_set(PayloadSize){	描述符
<b>initial_horizontal_view_angle</b>	u(8)
<b>horizontal_view_angle_max</b>	u(8)
<b>horizontal_view_angle_min</b>	u(8)
<b>name_len</b>	u(8)
for(i=0;i<name_len;i++)	
<b>stitching_software_name[i]</b>	u(8)
<b>camera_count</b>	ue(v)
}	
}	

## E.2 补充增强信息(SEI)负载语义

### E.2.1 缓存周期 SEI 消息语义

如果应用需要,缓存周期 SEI 消息应与 IDR 图像一起出现。该消息提供 HRD 初始信息。

**initial\_cpb\_removal\_delay**[SchedSelIdx] 定义了在 HRD 初始化以后第一个缓存周期的第一个 CPB 的初始延迟,即从图像的第一个比特到达 CPB 的时间,到该图像的数据开始从 CPB 移除的时间。其码字长度为 initial\_cpb\_removal\_delay\_length\_minus1 + 1,本标准规定以 90 kHz 为单位。

该值应大于 0,并小于  $90\ 000 \times \text{CpbSize}[\text{SchedSelIdx}] \div \text{BitRate}[\text{SchedSelIdx}]$ 。

### E.2.2 图像定时 SEI 消息定义

**cpb\_removal\_delay** 定义了该 SEI 消息关联的图像从 CPB 中移除前等待的时间。该值还可以用来计算图像数据到达 CPB 的最早的可能时间。该码字长度为 cpb\_removal\_delay\_length\_minus1 + 1。

对于码流中的第一个图像,cpb\_removal\_delay 应为 0。

**dpb\_output\_delay** 定义了图像数据从 CPB 中移除后到解码图像从 DPB 输出的等待时间,用来计算图像的 DPB 输出时间。其码字长度为 dpb\_output\_delay\_length\_minus1 + 1。

### E.2.3 全景视频参数 SEI 消息定义

**spherical\_pano\_flag** 球面全景视频标识位,0—其他,1—球面全景视频。

**projection\_type** 投影方式标识位,0—立方体,1—圆柱,2—棱锥。

**arrangement\_raster\_mode** 当 projection\_type = 0,即投影方式为立方体时,6 个正方形图像(前、后、左、右、上和下,索引值分别为:1、2、3、4、5 和 6)按照光栅排列的方式。取值范围:0~719,共 720 种排列方式。

**arrangement\_matrix\_mode** 当 projection\_type = 0,即投影方式为立方体时,6 个正方形图像的排列方式。取值范围:0~3,共 4 种数组类型:0—1x6,1—6x1,2—2x3,3—3x2。

**fullpano\_width\_inpixel** 指示一帧全部图像(单目二维视频:包含一幅图像;双目三维视频:包含两幅图像)的宽度,单位为样点个数。

**fullpano\_height\_inpixel** 指示一帧全部图像(单目二维视频:包含一幅图像;双目三维视频:包含两

幅图像)的高度,单位为样点个数。

**croppedarea\_left\_inpixel** 指示一幅全景图像左侧被裁掉的列数,单位为样点个数。

**croppedarea\_top\_inpixel** 指示一幅全景图像顶部被裁掉的行数,单位为样点个数。

**stereo\_mode** 立体影像模式及画面布局标识位,0—单目二维视频,1—双目三维左右排列视频,2—双目三维上下排列视频。

**croppedarea\_width\_inpixel** 指示一幅全景图像的宽度,单位为样点个数。

**croppedarea\_height\_inpixel** 指示一幅全景图像的高度,单位为样点个数。

**croppedarea\_width\_inpixel** 和 **croppedarea\_height\_inpixel** 的取值根据 **stereo\_mode** 的值按照表 E.5 进行设置。

表 E.5 全景图像宽度与高度在不同 **stereo\_mode** 取值时的计算方法

语法元素	单目二维视频 ( <b>stereo_mode</b> = 0)	双目三维左右排列视频 ( <b>stereo_mode</b> = 1)	双目三维上下排列视频 ( <b>stereo_mode</b> = 2)
<b>croppedarea_width_inpixel</b>	$\text{fullpano\_width\_inpixel} - \text{croppedarea\_left\_inpixel}$	$\text{fullpano\_width\_inpixel}/2 - \text{croppedarea\_left\_inpixel}$	$\text{fullpano\_width\_inpixel} - \text{croppedarea\_left\_inpixel}$
<b>croppedarea_height_inpixel</b>	$\text{fullpano\_height\_inpixel} - \text{croppedarea\_top\_inpixel}$	$\text{fullpano\_height\_inpixel} - \text{croppedarea\_top\_inpixel}$	$\text{fullpano\_height\_inpixel}/2 - \text{croppedarea\_top\_inpixel}$

**horizontal\_angle\_range** 全景图像水平方向的环绕度数,以度为单位,取值范围是 0~360。

**pitch\_angle\_up\_limit** 全景图像俯仰角的上限,其值必须大于等于 **pitch\_angle\_down\_limit**,以度为单位,取值范围是 -90~90。

**pitch\_angle\_down\_limit** 全景图像俯仰角的下限,其值必须小于等于 **pitch\_angle\_up\_limit**,以度为单位,取值范围是 -90~90。

**initial\_viewazimuth\_angle** 初始显示画面中心的水平方位角度,取值范围是  $-\text{horizontal\_angle\_range}/2 \sim \text{horizontal\_angle\_range}/2$ 。当取值为 0 时,如果 **projection\_type** = 0,则初始显示画面中心的水平方位角度设置为裁剪之后图像的中心列所对应的水平方位角度;如果 **projection\_type** = 1,则初始显示画面中心的水平方位角度设置为裁剪之后向前图的中心列所对应的水平方位角度。

**initial\_view\_pitch\_angle** 初始显示画面中心的俯仰角,以度为单位,取值范围是 **pitch\_angle\_down\_limit** ~ **pitch\_angle\_up\_limit**。

**initial\_view\_roll\_angle** 绕显示画面中心旋转的角度,以度为单位,取值范围是 -180~180。

**display\_width\_inpixels** 指示显示画面的图像宽度,单位为样点个数。

**display\_height\_inpixels** 指示显示画面的图像高度,单位为样点个数。

**initial\_horizontal\_view\_angle** 指示画面显示的初始水平视角,以度为单位。

**horizontal\_view\_angle\_max** 指示画面显示的最大水平视角,以度为单位,其值应小于或等于 180。

**horizontal\_view\_angle\_min** 指示画面显示的最小水平视角,以度为单位,其值应大于 0。

**name\_len** 指示拼接软件名称的长度。

**stitching\_software\_name** 表示拼接软件的名称。

**camera\_count** 指示拍摄全景图像的摄像机个数。



附录 F  
(规范性附录)  
智能分析数据描述

## F.1 智能分析数据语法

## F.1.1 图像分析规则

图像分析规则语法见表 F.1。

表 F.1 图像分析规则语法表

analysis_rule() {	描述符
object_min_width_minus1	u(16)
object_min_height_minus1	u(16)
min_dura_time	u(16)
max_dura_time	u(32)
line_num	u(8)
trigger_direction	u(2)
invade_action_type	u(4)
face_similarity	u(8)
density_unit	u(2)
}	

## F.1.2 运动目标检测

运动目标检测语法见表 F.2。

表 F.2 运动目标检测语法表

Moving_object_detection()	描述符
object_num	u(8)
analysis_level	u(1)
for(i=0;i< object_num;i++){	
object_id[i]	u(16)
object_width_minus1[i]	u(16)
object_height_minus1[i]	u(16)
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
object_color[i]	u(8)

表 F.2 (续)

Moving_object_detection() {	描述符
<b>object_sort[i]</b>	u(3)
if(object_sort == 0x03){	
<b>vehicle_sort[i]</b>	u(3)
<b>vehicle_info_id[i]</b>	u(8)
}	
if(analysis_level == 0x01){	
<b>object_speed_val[i]</b>	u(16)
<b>object_speed_rad[i]</b>	u(9)
<b>object_traipt_x[i]</b>	u(16)
<b>object_traipt_y[i]</b>	u(16)
}	
}	
while(byte_aligned() == FALSE){	
<b>reserved_bit</b>	u(1)
}	
}	

### F.1.3 人员属性分析

人员属性分析语法见表 F.3。

表 F.3 人员属性分析语法表

Human_property_analysis() {	描述符
<b>human_num</b>	u(8)
for(i=0;i< human_num;i++) {	
<b>human_id[i]</b>	u(16)
<b>human_property_num[i]</b>	u(8)
for(j=0;j< property_num;i++)	
<b>human_property[i,j]</b>	u(8)
}	
}	

### F.1.4 机动车特征分析

机动车特征分析语法见表 F.4。

表 F.4 机动车特征分析语法表

Vehicle_property_analysis()	描述符
<b>vehicle_num</b>	u(8)
for(i=0;i< vehicle_num;i++){	
<b>vehicle_id[i]</b>	u(16)
<b>vehicle_property_num[i]</b>	u(8)
for(j=0;j< property_num;j++)	
<b>vehicle_property[i,j]</b>	u(8)
}	
}	

### F.1.5 人脸比对

人脸比对语法见表 F.5。

表 F.5 人脸比对语法表

face_match()	描述符
<b>face_num</b>	u(8)
for(i=0;i< face_num;i++){	
<b>face_id[i]</b>	u(16)
<b>face_similarity[i]</b>	u(8)
}	
}	

### F.1.6 车牌识别

车牌识别语法见表 F.6。

表 F.6 车牌识别语法表

vehicle_licence_recognition()	描述符
<b>vehicleLicence_num</b>	u(8)
<b>analysis_level</b>	u(1)
for(i=0;i< vehicleLicence_num;i++){	
<b>vehicle_licence_type[i]</b>	u(4)
<b>vehicle_licence_color[i]</b>	u(2)
<b>vehicle_licence_no[i]</b>	u(64)
if(analysis_level == 0x01){	
<b>vehicle_licence_modify_flag[i]</b>	u(1)

表 F.6 (续)

vehicle_licence_recognition() {	描述符
<b>vehicle_licence_cover_flag[i]</b>	u(1)
}	
}	
while(byte_aligned() == FALSE)	
<b>reserved_bit</b>	u(1)
}	

#### F.1.7 绊线检测

绊线检测语法见表 F.7。

表 F.7 绊线检测语法表

pass_extension() {	描述符
<b>pass_num</b>	u(8)
for(i=0; i<pass_num; i++) {	
<b>object_category</b>	u(2)
<b>object_size</b>	u(2)
<b>moving_direction</b>	u(4)
<b>position_top_left_x[i]</b>	u(16)
<b>position_top_left_y[i]</b>	u(16)
<b>position_width_minus1[i]</b>	u(16)
<b>position_height_minus1[i]</b>	u(16)
}	
}	

#### F.1.8 入侵检测

入侵检测语法见表 F.8。

表 F.8 入侵检测语法表

invade_extension() {	描述符
<b>invade_num</b>	u(8)
for(i=0; i<invade_num; i++) {	
<b>object_category</b>	u(2)
<b>object_size</b>	u(2)
<b>moving_direction</b>	u(4)

表 F.8 (续)

invade_extension() {	描述符
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
position_width_minus1[i]	u(16)
position_height_minus1[i]	u(16)
}	
}	

### F.1.9 逆行检测

逆行检测语法见表 F.9。

表 F.9 逆行检测语法表

retrograde_extension() {	描述符
retrograde_num	u(8)
for(i=0; i< retrograde_num; i++) {	
object_category	u(2)
object_size	u(2)
moving_direction	u(4)
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
position_width_minus1[i]	u(16)
position_height_minus1[i]	u(16)
}	
}	

### F.1.10 徘徊检测

徘徊检测语法见表 F.10。

表 F.10 徘徊检测语法表

hover_extension() {	描述符
hover_num	u(8)
for(i=0; i< hover_num; i++) {	
object_category	u(2)
object_size	u(2)
moving_direction	u(4)

表 F.10 (续)

hover_extension() {	描述符
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
position_width_minus1[i]	u(16)
position_height_minus1[i]	u(16)
}	
}	

## F.1.11 遗留物检测



遗留物检测语法见表 F.11。

表 F.11 遗留物检测语法表

remnant_extension() {	描述符
<b>remnant_num</b>	u(8)
for(i=0; i< remnant_num; i++) {	
<b>object_category</b>	u(2)
<b>object_size</b>	u(2)
<b>object_color</b>	u(4)
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
position_width_minus1[i]	u(16)
position_height_minus1[i]	u(16)
}	
}	

## F.1.12 目标移除检测

目标移除物检测语法见表 F.12。

表 F.12 目标移除物检测语法表

moveout_extension() {	描述符
<b>moveout_num</b>	u(8)
for(i=0; i< moveout_num; i++) {	
<b>object_category</b>	u(2)
<b>object_size</b>	u(2)
<b>object_color</b>	u(4)

表 F.12 (续)

moveout_extension()	描述符
position_top_left_x[i]	u(16)
position_top_left_y[i]	u(16)
position_width_minus1[i]	u(16)
position_height_minus1[i]	u(16)
}	
}	

## F.1.13 目标数量统计

目标数量统计语法见表 F.13。

表 F.13 目标数量统计语法表

object_statistics()	描述符
begin_time	u(32)
end_time	u(32)
object_num	u(16)
person_num	u(16)
face_num	u(16)
vehicle_num	u(16)
thing_num	u(16)
object_density_abs	u(2)
person_density_abs	u(2)
face_density_abs	u(2)
vehicle_density_abs	u(2)
thing_density_abs	u(2)
object_density_rlt	u(8)
person_density_rlt	u(8)
face_density_rlt	u(8)
vehicle_density_rlt	u(8)
thing_density_rlt	u(8)
total_object_flowrate	u(16)
person_object_flowrate	u(16)
vehicle_object_flowrate	u(16)
reserved_bits	u(6)
}	

## F.2 智能分析信息扩展语义

### F.2.1 图像分析规则

**object\_min\_width\_minus1** 加 1 和 **object\_min\_height\_minus1** 加 1 等于目标的最小宽度和最小高度。

**min\_dura\_time** 表示最小持续时间,单位为秒。

**max\_dura\_time** 表示最大持续时间,单位为秒。

**line\_num** 表示包含绊线的条数。

**trigger\_direction** 表示触发方向,0 为从左到右,1 为从右到左,2 位任意方向。

**invade\_action\_type** 表示入侵行为的类型,如表 F.14 所示。

表 F.14 **invade\_action\_type** 的取值说明

<b>invade_action_type</b> 取值	含义
0	进入区域
1	离开区域
2	区域内出现
3	区域内消失
4	在区域内
5~15	自定义

**face\_similarity** 为人脸相似度,取值为百分比,取值不带百分号,大于该值认为是人脸。**density\_unit** 为密度检测数值单位,0 为密度等级,1 为密度百分比,2 为个数,其余数值保留。

### F.2.2 运动目标检测

**object\_num** 为 8 位无符号整数,表示识别出的目标数量。

**analysis\_level** 为 1 位无符号整数,表示分析级别。**analysis\_level** 等于 0 表示基本级别,**analysis\_level** 等于 1 表示高级。

**object\_id[i]** 为 16 位无符号整数,表示第 i 个目标的编号。

**object\_width\_minus1[i]** 加 1 和 **object\_height\_minus1[i]** 加 1 分别等于第 i 个目标的宽度和高度。以样点为单位计算的第 i 个目标的宽度、高度为:

以样点为单位计算的第 i 个目标的宽度、高度为:

**objectWidthInSample[i] = object\_width\_minus1[i] + 1**

**objectHeightInSample[i] = object\_height\_minus1[i] + 1**

**position\_top\_left\_x[i]** 和 **position\_top\_left\_y[i]** 分别表示第 i 个目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个目标的左上角位置的横坐标值、纵坐标值为:

**objectTopLeftSamplePositionX[i] = position\_top\_left\_x[i]**

**objectTopLeftSamplePositionY[i] = position\_top\_left\_y[i]**

**object\_color[i]** 为 8 位无符号整数,表示第 i 个目标的主体颜色编号,见表 F.15。

表 F.15 object\_color 的取值说明

object_color 取值	含义
01	黑
02	白
03	灰
04	红
05	蓝
06	黄
07	橙
08	棕
09	绿
10	紫
11	青
12	粉
13	透明
...	...
99	其他

object\_sort[i]为3位无符号整数,表示第i个目标的类别,见表F.16。

表 F.16 object\_sort 的取值说明

object_sort 取值	含义
0x01	人员
0x02	人脸
0x03	机动车
0x04	非机动车
0x05	物品
0x06	场景

vehicle\_sort[i]为3位无符号整数,表示第i个目标如果为车辆时车辆分类信息,见表F.17。

表 F.17 vehicle\_sort 的取值说明

vehicle_sort 取值	含义
0x01	客车
0x02	货车
0x03	其他车辆

vehicle\_info\_id[i]为11位无符号整数,表示第i个目标如果为车辆时车辆详细信息编号。

object\_speed\_val[i]为16位无符号整数,表示第i个目标的运动速度,单位为样点数/秒。

**object\_speed\_rad[i]** 为 9 位无符号整数, 表示第 i 个目标运动方向, 以角度为单位, 取值范围 [0, 359], 水平向右为 0, 逆时针转动时角度增加。

**object\_traipt\_x[i]** 为 16 位无符号整数, 表示第 i 个目标的运动轨迹点 X 坐标, 单位为样点值。

**object\_traipt\_y[i]** 为 16 位无符号整数, 表示第 i 个目标的轨迹点 Y 坐标, 单位为样点值。

**object\_traipt\_x[i]** 和 **object\_traipt\_y[i]** 分别表示第 i 个目标的运动轨迹点的横坐标值、纵坐标值。以样点为单位计算的第 i 个目标的运动轨迹点位置的横坐标值、纵坐标值为:

**objectTraipntSamplePositionX[i] = object\_traipt\_x[i]**

**objectTraipntSamplePositionY[i] = object\_traipt\_y[i]**

**reserved\_bit** 应等于 0。

#### F.2.3 人员属性分析

**human\_num** 为 8 位无符号整数, 表示识别出的人员数量。

**human\_id[i]** 为 16 位无符号整数, 表示识别出的第 i 个人员编号。

**human\_property\_num[i]** 为 8 位无符号整数, 表示识别出的第 i 个人员的属性数量。

**human\_Property[i,j]** 为 8 位无符号整数, 表示识别出的第 i 个人员的第 j 个属性。

#### F.2.4 机动车特征分析

**vehicle\_num** 为 8 位无符号整数, 表示识别出的车辆数量。

**vehicle\_id[i]** 为 16 位无符号整数, 表示识别出的车辆编号。

**vehicle\_property\_num[i]** 为 8 位无符号整数, 表示识别出的第 i 个车辆的属性数量。

**vehicle\_property[i,j]** 为 8 位无符号整数, 表示识别出的第 i 个车辆的第 j 个属性。

#### F.2.5 人脸比对

**face\_num** 为 8 位无符号整数, 表示识别出的符合比对条件的人脸数量。

**face\_id[i]** 为 16 位无符号整数, 表示识别出的第 i 个人脸编号。

**face\_similarity[i]** 为 8 位无符号整数, 表示人脸相似度。

#### F.2.6 车牌识别

**vehicle\_licence\_num** 为 8 位无符号整数, 表示识别出的车牌数量。

**vehicle\_licence\_type[i]** 为 4 位无符号整数, 表示识别出的第 i 个车牌种类, 见表 F.18。

表 F.18 **vehicle\_licence\_type** 的取值说明

vehicle_licence_type 取值	含义
0x00	大型汽车号牌
0x01	小型汽车号牌
0x02	使、领馆汽车号牌
0x03	港澳入出境车号牌
0x04	教练汽车号牌
0x05	警用汽车号牌
0x06	武警汽车号牌
0x07	军用汽车号牌
reserved	其他

**vehicle\_licence\_color[i]** 为 2 位无符号整数, 表示识别出的第 i 个车牌颜色, 见表 F.19。

表 F.19 **vehicle\_licence\_color** 的取值说明

vehicle_licence_color 取值	含义
0x00	蓝色
0x01	白色
0x02	黄色
0x03	黑色

**vehicle\_licence\_no[i]** 为 8 位字符串, 表示识别出的第 i 个车牌编号。

**vehicle\_licence\_modify\_flag[i]** 为 1 位无符号整数, 等于 1 表示车牌有涂改; 等于 0 表示车牌无涂改。

**vehicle\_licence\_cover\_flag[i]** 为 1 位无符号整数, 等于 1 表示车牌有遮挡; 等于 0 表示车牌无遮挡。

**reserved\_bit** 应等于 0。

### F.2.7 绊线检测

**pass\_num** 指示通过警戒线目标的个数。

**object\_category** 指示通过警戒线目标的类别, **object\_category** 等于 0 表示人, **object\_category** 等于 1 表示车, **object\_category** 等于 2 表示其他物体。

**object\_size** 指示通过警戒线目标的尺寸, **object\_size** 等于 0 表示小尺寸, **object\_size** 等于 1 表示中等尺寸, **object\_size** 等于 2 表示大尺寸, **object\_size** 等于 3 表示巨大尺寸。

**moving\_direction** 指示通过警戒线目标的运动方向, **moving\_direction** 等于 0 表示北, **moving\_direction** 等于 1 表示东北, **moving\_direction** 等于 2 表示东, **moving\_direction** 等于 3 表示东南, **moving\_direction** 等于 4 表示南, **moving\_direction** 等于 5 表示西南, **moving\_direction** 等于 6 表示西, **moving\_direction** 等于 7 表示西北, **moving\_direction** 等于 8 表示上, **moving\_direction** 等于 9 表示右上, **moving\_direction** 等于 10 表示右下, **moving\_direction** 等于 11 表示下, **moving\_direction** 等于 12 表示左下, **moving\_direction** 等于 13 表示左上, **moving\_direction** 等于 14 表示左, **moving\_direction** 等于 15 表示右。

**position\_top\_left\_x[i]** 和 **position\_top\_left\_y[i]** 分别表示第 i 个通过警戒线目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个通过警戒线目标的左上角位置的横坐标值、纵坐标值为:

```
passTopLeftSamplePositionX[i] = position_top_left_x[i]
```

```
passTopLeftSamplePositionY[i] = position_top_left_y[i]
```

**position\_width\_minus1[i]** 加 1 和 **position\_height\_minus1[i]** 加 1 分别等于第 i 个通过警戒线目标的宽度和高度。以样点为单位计算的第 i 个通过警戒线目标的宽度、高度为:

```
passWidthInSample[i] = position_width_minus1[i] + 1
```

```
passHeightInSample[i] = position_height_minus1[i] + 1
```

### F.2.8 入侵检测

**invade\_num** 指示进入禁入区域目标的个数。

**object\_category** 指示进入禁入区域目标的类别, **object\_category** 等于 0 表示人, **object\_category** 等于 1 表示车, **object\_category** 等于 2 表示其他物体。

**object\_size** 指示进入禁入区域目标的尺寸, **object\_size** 等于 0 表示小尺寸, **object\_size** 等于 1 表示

中等尺寸,object\_size 等于 2 表示大尺寸,object\_size 等于 3 表示巨大尺寸。

**moving\_direction** 指示进入禁入区域目标的运动方向,moving\_direction 等于 0 表示北,moving\_direction 等于 1 表示东北,moving\_direction 等于 2 表示东,moving\_direction 等于 3 表示东南,moving\_direction 等于 4 表示南,moving\_direction 等于 5 表示西南,moving\_direction 等于 6 表示西,moving\_direction 等于 7 表示西北,moving\_direction 等于 8 表示上,moving\_direction 等于 9 表示右上,moving\_direction 等于 10 表示右下,moving\_direction 等于 11 表示下,moving\_direction 等于 12 表示左下,moving\_direction 等于 13 表示左上,moving\_direction 等于 14 表示左,moving\_direction 等于 15 表示右。

**position\_top\_left\_x[i]**和**position\_top\_left\_y[i]** 分别表示第 i 个进入禁入区域目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个进入禁入区域目标的左上角位置的横坐标值、纵坐标值为:

```
passTopLeftSamplePositionX[i] = position_top_left_x[i]
passTopLeftSamplePositionY[i] = position_top_left_y[i]
```

**position\_width\_minus1[i]**加 1 和 **position\_height\_minus1[i]**加 1 分别等于第 i 个进入禁入区域目标的宽度和高度。以样点为单位计算的第 i 个进入禁入区域目标的宽度、高度为:

```
passWidthInSample[i] = position_width_minus1[i] + 1
passHeightInSample[i] = position_height_minus1[i] + 1
```

#### F.2.9 逆行检测

**retrograde\_num** 指示向反方向运动目标的个数。

**object\_category** 指示向反方向运动目标的类别,object\_category 等于 0 表示人,object\_category 等于 1 表示车,object\_category 等于 2 表示其他物体。

**object\_size** 指示向反方向运动目标的尺寸,object\_size 等于 0 表示小尺寸,object\_size 等于 1 表示中等尺寸,object\_size 等于 2 表示大尺寸,object\_size 等于 3 表示巨大尺寸。

**moving\_direction** 指示向反方向运动目标的运动方向,moving\_direction 等于 0 表示北,moving\_direction 等于 1 表示东北,moving\_direction 等于 2 表示东,moving\_direction 等于 3 表示东南,moving\_direction 等于 4 表示南,moving\_direction 等于 5 表示西南,moving\_direction 等于 6 表示西,moving\_direction 等于 7 表示西北,moving\_direction 等于 8 表示上,moving\_direction 等于 9 表示右上,moving\_direction 等于 10 表示右下,moving\_direction 等于 11 表示下,moving\_direction 等于 12 表示左下,moving\_direction 等于 13 表示左上,moving\_direction 等于 14 表示左,moving\_direction 等于 15 表示右。

**position\_top\_left\_x[i]**和**position\_top\_left\_y[i]** 分别表示第 i 个向反方向运动目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个向反方向运动目标的左上角位置的横坐标值、纵坐标值为:

```
passTopLeftSamplePositionX[i] = position_top_left_x[i]
passTopLeftSamplePositionY[i] = position_top_left_y[i]
```

**position\_width\_minus1[i]**加 1 和 **position\_height\_minus1[i]**加 1 分别等于第 i 个向反方向运动目标的宽度和高度。以样点为单位计算的第 i 个向反方向运动目标的宽度、高度为:

```
passWidthInSample[i] = position_width_minus1[i] + 1
passHeightInSample[i] = position_height_minus1[i] + 1
```

#### F.2.10 徘徊检测

**hover\_num** 指示在警戒区域徘徊运动目标的个数。

**object\_category** 指示在警戒区域徘徊运动目标的类别, object\_category 等于 0 表示人, object\_category 等于 1 表示车, object\_category 等于 2 表示其他物体。

**object\_size** 指示在警戒区域徘徊运动目标的尺寸, object\_size 等于 0 表示小尺寸, object\_size 等于 1 表示中等尺寸, object\_size 等于 2 表示大尺寸, object\_size 等于 3 表示巨大尺寸。

**moving\_direction** 指示在警戒区域徘徊运动目标的运动方向, moving\_direction 等于 0 表示北, moving\_direction 等于 1 表示东北, moving\_direction 等于 2 表示东, moving\_direction 等于 3 表示东南, moving\_direction 等于 4 表示南, moving\_direction 等于 5 表示西南, moving\_direction 等于 6 表示西, moving\_direction 等于 7 表示西北, moving\_direction 等于 8 表示上, moving\_direction 等于 9 表示右上, moving\_direction 等于 10 表示右下, moving\_direction 等于 11 表示下, moving\_direction 等于 12 表示左下, moving\_direction 等于 13 表示左上, moving\_direction 等于 14 表示左, moving\_direction 等于 15 表示右。

**position\_top\_left\_x[i]** 和 **position\_top\_left\_y[i]** 分别表示第 i 个在警戒区域徘徊运动目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个在警戒区域徘徊运动目标的左上角位置的横坐标值、纵坐标值为:

```
passTopLeftSamplePositionX[i] = position_top_left_x[i]
```

```
passTopLeftSamplePositionY[i] = position_top_left_y[i]
```

**position\_width\_minus1[i]** 加 1 和 **position\_height\_minus1[i]** 加 1 分别等于第 i 个在警戒区域徘徊运动目标的宽度和高度。以样点为单位计算的第 i 个在警戒区域徘徊运动目标的宽度、高度为:

```
passWidthInSample[i] = position_width_minus1[i] + 1
```

```
passHeightInSample[i] = position_height_minus1[i] + 1
```

### F.2.11 遗留物检测

**remnant\_num** 指示警戒区域内遗留物的个数。

**object\_category** 指示警戒区域内遗留物的类别, object\_category 等于 0 表示人, object\_category 等于 1 表示车, object\_category 等于 2 表示其他物体。

**object\_size** 指示警戒区域内遗留物的尺寸, object\_size 等于 0 表示小尺寸, object\_size 等于 1 表示中等尺寸, object\_size 等于 2 表示大尺寸, object\_size 等于 3 表示巨大尺寸。

**object\_color** 指示警戒区域内遗留物的颜色, 见表 F.20。

表 F.20 **object\_color** 的取值说明

object_color 取值	含义
01	黑
02	白
03	灰
04	红
05	蓝
06	黄
07	橙
08	棕
09	绿
10	紫

表 F.20 (续)

object_color 取值	含义
11	青
12	粉
13	透明
...	...
15	其他

**position\_top\_left\_x[i]** 和 **position\_top\_left\_y[i]** 分别表示第 i 个警戒区域内遗留物的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个警戒区域内遗留物的左上角位置的横坐标值、纵坐标值为：

passTopLeftSamplePositionX[i] = position\_top\_left\_x[i]

passTopLeftSamplePositionY[i] = position\_top\_left\_y[i]

**position\_width\_minus1[i]** 加 1 和 **position\_height\_minus1[i]** 加 1 分别等于第 i 个警戒区域内遗留物的宽度和高度。以样点为单位计算的第 i 个警戒区域内遗留物的宽度、高度为：

passWidthInSample[i] = position\_width\_minus1[i] + 1

passHeightInSample[i] = position\_height\_minus1[i] + 1

#### F.2.12 目标移除检测

**moveout\_num** 指示警戒区域内移除目标的个数。

**object\_category** 指示警戒区域内移除目标的类别, object\_category 等于 0 表示人, object\_category 等于 1 表示车, object\_category 等于 2 表示其他物体。

**object\_size** 指示警戒区域内移除目标的尺寸, object\_size 等于 0 表示小尺寸, object\_size 等于 1 表示中等尺寸, object\_size 等于 2 表示大尺寸, object\_size 等于 3 表示巨大尺寸。

**object\_color** 指示警戒区域内移除目标的颜色, 见表 F.21。

表 F.21 object\_color 的取值说明

object_color 取值	含义
01	黑
02	白
03	灰
04	红
05	蓝
06	黄
07	橙
08	棕
09	绿
10	紫
11	青

表 F.21 (续)

object_color 取值	含义
12	粉
13	透明
...	...
15	其他

**position\_top\_left\_x[i]** 和 **position\_top\_left\_y[i]** 分别表示第 i 个警戒区域内移除目标的左上角的横坐标值、纵坐标值。以样点为单位计算的第 i 个警戒区域内移除目标的左上角位置的横坐标值、纵坐标值为：

$$\text{passTopLeftSamplePositionX}[i] = \text{position_top_left_x}[i]$$

$$\text{passTopLeftSamplePositionY}[i] = \text{position_top_left_y}[i]$$

**position\_width\_minus1[i]** 加 1 和 **position\_height\_minus1[i]** 加 1 分别等于第 i 个警戒区域内移除目标的宽度和高度。以样点为单位计算的第 i 个警戒区域内移除目标的宽度、高度为：

$$\text{passWidthInSample}[i] = \text{position_width_minus1}[i] + 1$$

$$\text{passHeightInSample}[i] = \text{position_height_minus1}[i] + 1$$

### F.2.13 目标数量统计

**begin\_time** 表示统计开始时间, 对连续视频有效, 取值等于从公元 1970 年 1 月 1 日 0 时整至该值所表示的实际时间的秒数。

**end\_time** 表示统计结束开始时间, 对连续视频有效, 取值等于从公元 1970 年 1 月 1 日 0 时整至该值所表示的实际时间的秒数。

**object\_num** 表示目标总数。

**person\_num** 表示目标为人员的总数。

**face\_num** 表示目标为人脸的总数。

**vehicle\_num** 表示目标为车辆的总数。

**thing\_num** 表示目标为物体的总数。

**object\_density\_abs** 表示区域内目标密度等级, 取值如表 F.22 所示。



**person\_density\_abs** 表示区域内人员密度等级, 取值如表 F.22 所示。

**face\_density\_abs** 表示区域内人脸密度等级, 取值如表 F.22 所示。

**vehicle\_density\_abs** 表示区域内车辆密度等级, 取值如表 F.22 所示。

**thing\_density\_abs** 表示区域内物体密度等级, 取值如表 F.22 所示。

表 F.22 目标密度取值与含义对应关系表

取值	含义
0	很稀疏
1	稀疏
2	密集
3	很密集

**object\_density\_rlt** 表示区域内目标相对密度, 取值为百分比, 不含百分号。

**person\_density\_rlt** 表示区域内人员相对密度,取值为百分比,不含百分号。

**face\_density\_rlt** 表示区域内人脸相对密度,取值为百分比,不含百分号。

**vehicle\_density\_rlt** 表示区域内车辆相对密度,取值为百分比,不含百分号。

**thing\_density\_rlt** 表示区域内物体相对密度,取值为百分比,不含百分号。

**total\_object\_flowrate** 表示时间段内的目标总个数。

**person\_object\_flowrate** 表示时间段内的人员总个数。

**vehicle\_object\_flowrate** 表示时间段内的人员总个数。

**reserved\_bits** 应等于 0。

附录 G  
(规范性附录)  
音频档次和级别

### G.1 概述

本附录描述了不同档次和级别所对应的各种限制。档次与级别规定了对比特流的限制,因此也限制了比特流解码所需的能力。每个档次定义了一个算法特征的子集,并限定所有与该档次一致的解码器都应支持。每个级别定义了对本标准中的语法要素取值的限制集合。相同的级别定义集合用于所有的档次,但单独的应用对所支持的档次可能支持不同的级别。一般来说,对于特定的一个档次,不同的级别对应于对解码器负载和存储器容量的不同要求。

如果一个解码器能对某个档次和级别所规定的语法元素正确解码,则称此解码器在这个档次和级别上符合本标准。如果比特流中不存在某个档次和级别所不允许的语法元素,并且其所含有的语法元素的值不超过此档次和级别所允许的范围,则认为此比特流在这个档次和级别上符合本标准。

profile\_id 和 level\_id 定义了比特流的档次和级别。

注:解码器不宜因为 profile\_id 或 level\_id 的取值落在本标准所规定的值之间,就推定这个值所代表的能力处于规定好的档次与级别之间。

### G.2 音频档次

音频档次主要定义编码器所包括的主要编码工具,目前分 3 个档次:简单档次、主要档次和高级档次。profile\_id 采用 2 比特表示,0 表示禁止,1 表示简单档次、2 表示主要档次,3 表示高级档次。具体定义见表 G.1。

表 G.1 音频档次定义

编码工具	简单档次	主要档次	高级档次
ACELP	支持	支持	支持
BWE	支持	支持	支持
TVC	不支持	不支持	支持
识别特征参数的直接编码模式	不支持	支持	支持
识别特征参数的预测编码模式	不支持	不支持	支持

### G.3 音频级别

音频级别主要限制编码参数取值和编解码延迟,level\_id 采用 4 比特表示,共 16 个级别,见表 G.2 和表 G.3。

表 G.2 音频级别定义

level_id	级别
0	禁止
1	1.0
2	1.1
3	1.2
4~15	保留

表 G.3 音频级别 1.0~1.2 参数限制

参数	级别		
	1.0	1.1	1.2
内部采样频率/kHz	12.8 和 16	24 和 25.6	32 和 38.4
音频超帧样本点数	512	512	512
最大编解码延迟/ms	60	40	30
最大比特率/(bit/s)	23 050	34 000	48 610

注 1：编解码延迟包括：1 个超帧长度 +LPC 分析窗前瞻样本+其他延迟(如采样频率转换等)。

注 2：比特率指 RAW 格式码率，包括帧头和扩展帧头开销。

附录 H  
(规范性附录)  
异常声音事件类型定义

异常声音事件类型定义见表 H.1。

表 H.1 异常声音事件类型定义

事件类型	事件描述
0	正常声音事件(包括闹市噪声,汽车噪声,高斯噪声和正常人说话声等)
1	人的尖叫声、救命声
2	枪声
3	爆炸声
4	报警声
5	玻璃破碎声
6~255	保留



附录 I  
(资料性附录)  
VAD 检测

### I.1 概述

VAD 检测将输入的音频信号分为两类:语音和非语音(噪声或静音)。识别特征参数提取时需要检测每帧信号的类别,将来模式识别模块会根据信号的类别,将非语音帧信号的识别特征参数丢掉。识别模块在对识别特征参数进行后处理时,需要连续的帧参数计算识别特征参数的一阶导数和二阶导数,因此识别特征参数提取时需要保留非语音帧信号的识别特征参数。

### I.2 VAD 检测介绍

VAD 检测包含两个阶段:第一阶段是基于帧的检测阶段,内部包含三种检测方式;第二阶段为决策阶段。第一阶段的每种检测结果都存储在循环缓冲中,用来分析并得出语音的似然值。第二阶段的最终决策结果需要参考缓冲中的最初几帧,所以该阶段提供了预测机制。同时,此阶段还提供了延迟释放机制,延迟释放的持续时间同语音的似然值相关。



### I.3 检测阶段

VAD 检测采用的参数是语音开始时的能量加速度,该参数具有较好噪声鲁棒性。这种加速度可以通过以下三种方法来计算:

a) 全带频谱检测法

全带频谱测量法采用的参数,是通过两阶段维纳滤波器中第一阶段所得出的 Mel 域维纳滤波器系数(见 J.8)。对 Mel 域维纳滤波器系数求和后再平方的值作为输入值 input。

每帧的处理步骤如下所示:

```

if(frame<15&&Acceleration<2.5)
    tracker= MAX(tracker,input)
if(input<tracker×UpperBound&&input>tracker×LowerBound)
    tracker=a×tracker+(1-a) ×input
if(input<tracker×Floor)
    tracker=b×tracker+(1-b) ×input
if(input>tracker×threshold)
    return true
else
    return false

```

式中:

```

a=0.8;
b=0.97;
UpperBound=1.5;
LowerBound=0.75;

```

Floor=0.5;  
 threshold=1.65;  
 tracker —— 噪声能量估计值;  
 Acceleration —— 测量的加速度,可以通过连续输入的二阶差分来估计,但本检测算法通过跟踪连续输入的两个平均数  $0 \times mean + 1 \times input$  和  $((frame - 1) \times mean + 1 \times input) / frame$  的比率来估计。

b) 子带频谱检测法

子带频谱检测法的输入是由方法 1 中产生的第二、第三和第四 Mel 域维纳滤波器系数的平均值。检测器对每帧的处理步骤如下:

```
input = p × currentInput + (1 - p) × PreviousInput ;
if(Frame < 15)  tracker = MAX(tracker, input);
if(input < tracker × UpperBound  &&  input > tracker × LowerBound)
    tracker = a × tracker + (1 - a) × input ;
if(input < tracker × Floor)
    tracker = b × tracker + (1 - b) × input ;
if(input > tracker × threshold)
    return true ;
else
    return false ;
```

式中:

$p = 0.75$ ;  
 $threshold = 3.25$ ,其他参数和方法 1 中相同。

c) 频谱方差检测法

频谱方差检测法的输入部分是由每帧全带范围内线性频率维纳滤波器系数的方差构成。方差的计算公式为:

$$\frac{1}{N_{\text{SPEC}}} \sum_{bin=0}^{N_{\text{SPEC}}-1} (H_2(bin))^2 - \left( \sum_{bin=0}^{N_{\text{SPEC}}-1} H_2(bin) \right)^2 / N_{\text{SPEC}}^2 \quad \dots\dots\dots (I.1)$$

式中:

$N_{\text{SPEC}} = N_{\text{FFT}} / 4$ ;

$H_2(bin)$ ——线性频率维纳滤波器系数。

方法 3 的第一步同方法 2 的 b);第二步到第四步同方法 1 的 b)~d),其中  $LowerBound = 0.85$ ,  $Floor = 0.25$ ,其他参数不变。

#### I.4 决策阶段

VAD 决策算法输入为 I.3 讨论的三种方法输出的结果。这三种方法得到结果 true 或 false(T 或 F),并将其存储在缓冲中。连续帧得出的结果会不断填充缓冲。该过程提供了缓冲模式的上下文分析。只有缓冲填满了有效结果之后,VAD 决策算法才会进行输出。该过程导致了值为缓冲长度减一的帧延迟。

对于一个  $N = 7$  帧的缓冲,最新的结果存放在第  $N$  个位置。有后续结果进入时,缓冲中的值向左平移,如图 I.1 所示。



图 I.1 VAD 缓冲区示意图

VAD 决策算法处理步骤如下所示：

a)  $V_N = \text{Measurement 1} \text{ or } \text{Measurement 2} \text{ or } \text{Measurement 3}$

由全带频谱检测法、子带频谱检测法和频谱方差检测法得出的值有一个为 true 时,  $V_N$  的结果也是 true, 并将  $V_N$  存储到位置为 N 的缓冲中。

$$M = \text{MAX} \left\{ \begin{array}{l} C++ , \quad V_i = \text{true} \\ C=0 , \quad V_i = \text{false} \end{array} \right. , \quad 1 < i < N \right\}_c \dots \dots \dots \quad (\text{I.2})$$

b) 决策算法分析缓冲中的结果, 并寻找出缓冲内值为 true 的最长连续序列。在寻找过程中, 如果下一个值为 true 时, 算子 C 加 1, 反之下一个值为 false 时, C 清零。整个缓冲扫描结束后, 将算子 C 的最大值赋给 M。例如, 序列 T T F T T T F 扫描后, M 的值为 3。

c) *if*( $M \geq S_p$   $\&\&$   $T < L_s$ )

$T = L_s$ ;

$S_p$  为“可能是语音”的阈值, 对应着第二步得出的 true 值连续序列最大值  $M \geq 3$  的情况。如果延迟释放计时器 T 小于  $L_s$ , 则给 T 赋值短延迟释放时间( $L_s = 5$  帧)。

d) *if*( $M \geq S_L$   $\&\&$   $F > F_s$ )

$T = L_M$ ;

*else if*( $M \geq S_L$ )

$T = L_L$ ;

$S_L$  为“近似为语音”的阈值, 对应着第二步得出的 true 值连续序列最大值 M 大于等于 4 的情况。如果当前帧序号 F 在初始导入安全周期  $F_s$ (35 帧)之外, 则给 T 赋值中延迟释放时间 T ( $L_M = 23$  帧); 否则, 给 T 赋值长延迟释放时间 T( $L_L = 50$  帧), 这样做的目的是, 防止语音过早出现引起检测器的初始化噪声估计值太大。

e) *if*( $M < S_p$   $\&\&$   $T > 0$ )

$T--$ ;

如果 M 没有达到阈值  $S_p$  时, 将 T 减一。因此 T 只有在语音不存在的情况下才会减少。

f) *if*( $T > 0$ )

*return* true;

*else*

*return* false;

如果 T 大于 0 时, 输出为 true(语音帧); 否则, 输出为 false(非语音帧)。

g) 在下一帧到达之前, 缓冲区左移以接受新的输入帧。

从上面的 VAD 决策过程来看, 输出的语音或非语音判决应用于即将离开缓冲区的帧, 相应的预测机制如图 I.2 所示。

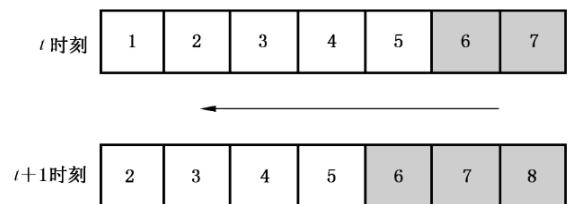


图 I.2 预测机制下缓冲区示意图

在  $t$  时刻时, 缓冲区已经被 7 帧数据填满, 第 6 帧和第 7 帧的  $V_N$  为 true。基于上述决策算法, 第一帧的决策结果为 false(非语音帧)。在  $t+1$  时刻时, 缓冲区左移并移出第一帧, 新的第 8 帧  $V_N$  结果为 true。应用决策算法, 第二帧的决策结果为 true(语音帧)。当有新帧到达时, 对于第 3,4,5 帧也会得到同样的结果。这样就形成了一个 4 帧的短时预测(使用第 6、7、8 帧的结果对第 2~5 帧进行预测)。

## 附录 J

(资料性附录)

## 噪声消除

## J.1 概述

噪声消除算法主要作用是降低背景噪声,提高信号的信噪比。无论语音识别还是声纹识别算法,噪声对识别结果影响很大。因此在识别特征参数提取之前,应先对信号进行降噪处理。

## J.2 Mel 域两阶段维纳(Wiener)滤波器

基于维纳滤波器的噪声消除算法由两阶段组成,见图 J.1。输入信号通过第一阶段的降噪处理后,在第二阶段根据处理后信号的信噪比进行噪声消除。

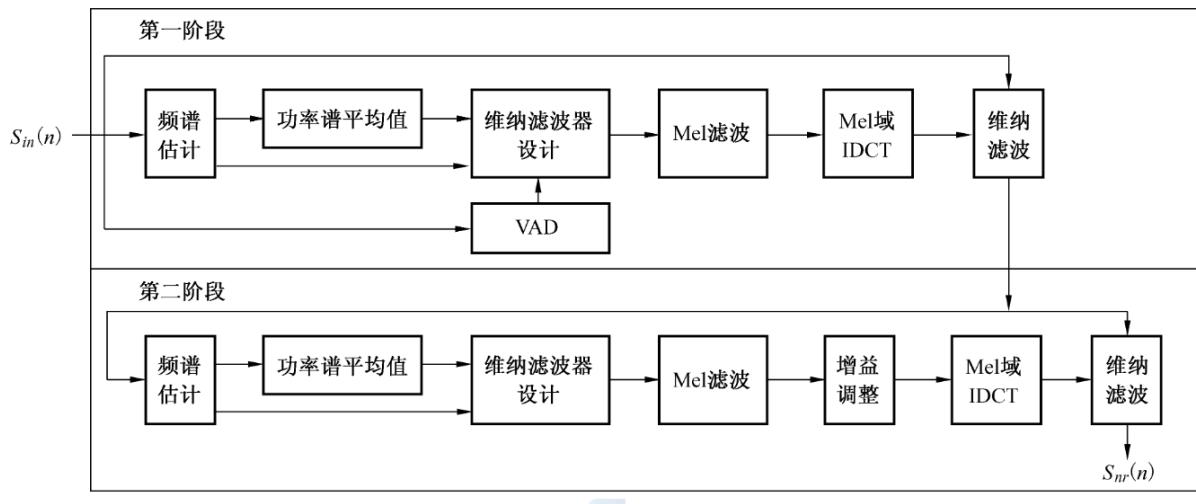


图 J.1 噪声消除流程图

输入信号首先按照帧的大小进行切分,然后在频谱估计模块中计算出每帧的线性频谱估计。在功率谱平均值模块内,对信号频谱按帧进行平滑处理。在维纳滤波器设计模块中,根据当前帧的频谱估计和噪声谱估计(噪声谱估计是通过 VAD 检测的噪声帧进行估计),计算出频域维纳滤波器系数。线性维纳滤波器系数经过 Mel 滤波器组进行滤波处理,得到了 Mel 域维纳滤波器。然后通过 Mel 域 IDCT 计算 Mel 域维纳滤波器的脉冲响应。最后,将每阶段的输入信号通过维纳滤波器进行滤波。在图 J.1 中,第二阶段的输入信号就是第一阶段的输出信号。此外,第二阶段中增益调整模块的主要功能是对噪声消除的增益进行控制。

## J.3 缓冲

噪声消除模块中输入信号以帧为单位,每帧长度为 10 ms(160 个样本)。噪声消除过程中的每阶段都需要一个大小为四帧的缓冲区(frame0~frame3)。当有新帧输入时,这两个缓冲区依次移动一帧。新输入帧被放置在第一个缓冲区的 frame3 位置上。

首先,对第一个缓冲区中的 frame1(样本 160~319)进行降噪,并把降噪后的帧放在第二个缓冲区

中的 frame3 位置上。然后,对第二个缓冲区中的 frame1 作降噪处理,此帧是噪声消除模块的输出。因此,噪声消除的每阶段都有 2 帧(20ms)的延迟。在每阶段中,频谱估计的窗长为 25 ms(样本 120~519)。图 J.2 给出两阶段噪声消除过程的缓冲区示意图。

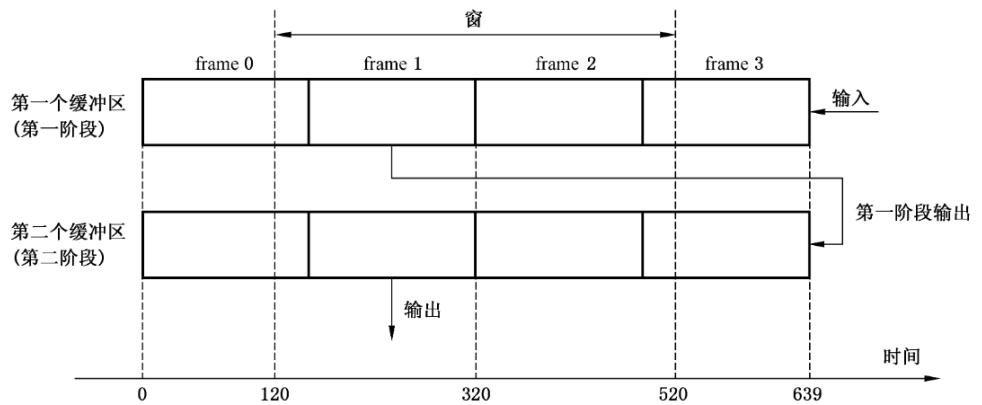


图 J.2 两阶段噪声消除过程的缓冲区示意图

## J.4 频谱估计

输入信号被分成  $N_{in}$  个样本的重叠帧, 帧长为 25 ms ( $N_{in}=400$ ) 并且有 10 ms (160 个样本) 的帧移。每帧  $s_{in}(n)$  都要作加窗处理, 采用长为  $N_{in}$  的汉宁 (Hanning) 窗, 见式 J.1 所示。

$N_{in}$  和  $N_{FFT} - 1$  之间的样本补零,  $N_{FFT} = 512$  是 FFT 长度:

对  $s_{\text{FFT}}(n)$  进行 FFT, 以求出频谱:

式中：

*bin*——FFT 频率索引。

计算功率谱  $P(bin)$ :

$$P(bin) = |X(bin)|^2, \quad 0 \leqslant bin \leqslant N_{FFT}/2 \quad \dots \dots \dots \quad (J.5)$$

再对功率谱  $P(bin)$  进行平滑处理：

$$P_{in}(bin) = \frac{P(2 \times bin) + P(2 \times bin + 1)}{2}, \quad 0 \leqslant bin \leqslant N_{\text{FFT}}/4 \quad \dots \dots \dots (16)$$

$$P_{in}(N_{\text{FET}}/4) = P(N_{\text{FET}}/2)$$

平滑处理后,功率谱的长度缩短为  $N_{\text{SPEC}} = N_{\text{FFT}}/4 + 1$ 。

### 3.5 功率谱平均值

对连续  $T_{\text{PSD}}$  帧求其功率谱  $P_{in}(bin)$  的平均值, 见图 J.3。

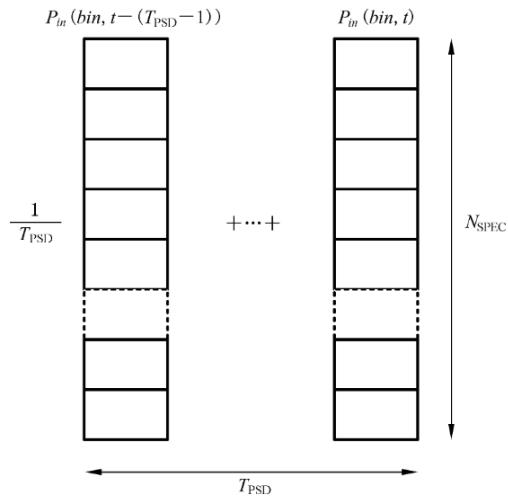


图 J.3 功率谱平均值

功率谱平均值为：

$$P_{in\_PSD}(bin, t) = \frac{1}{T_{PSD}} \sum_{i=0}^{T_{PSD}-1} P_{in}(bin, t-i), \quad 0 \leq bin \leq N_{SPEC} - 1 \quad \dots \dots \dots \quad (J.7)$$

式中：

$T_{PSD}=2$ ；

$t$ ——帧索引。

## J.6 噪声估计的 VAD

根据帧索引  $t$ ，计算出每帧的遗忘因子  $lambdaLTE$ ：

```
if( $t < NB\_FRAME\_THRESHOLD\_LTE$ )
     $lambdaLTE = 1 - 1/t$ ; .....(J.8)
else
     $lambdaLTE = LAMBDA\_LTE$ ;
```

式中：

$NB\_FRAME\_THRESHOLD\_LTE = 10$ ；

$LAMBDA\_LTE = 0.97$ 。

输入信号  $s_{in}(n)$  的连续  $M$  个 ( $M=160$ ) 样本的对数能量  $frameEn$  为：

$$frameEn = 0.5 + \frac{16}{\ln 2} \times \ln \left( \frac{\left( 64 + \sum_{i=0}^{M-1} s_{in}(n)^2 \right)}{64} \right) \quad \dots \dots \dots \quad (J.9)$$

用  $frameEn$  更新  $meanEn$ ：

```

if((frameEn-meanEn)<SNR_THRESHOLD_UPD_LTE) || (t<MIN_FRAME))
{
    if((frameEn<meanEn) || (t<MIN_FRAME))
        meanEn=meanEn+(1-lambdaLTE)*(frameEn-meanEn);
    else
        meanEn=meanEn+(1-lambdaLTEhigherE)*(frameEn-meanEn);
    if(meanEn<ENERGY_FLOOR)
        meanEn=ENERGY_FLOOR;
}

```

式中：

$SNR\_THRESHOLD\_UPD\_LTE = 20;$   
 $ENERGY\_FLOOR = 80;$   
 $MIN\_FRAME = 10;$   
 $\lambda_{LTEhigher} = 0.99.$

根据  $frameEn$  和  $meanEn$  这两个参数，确定当前帧是语音帧 ( $flagVAD_{nest} = 1$ ) 还是噪声帧 ( $flagVAD_{nest} = 0$ )：

```

if(t > 4)
{
    if((frameEn - meanEn) > SNR_THRESHOLD_VAD)
    {
        flagVAD_nest = 1;
        nbSpeechFrame = nbSpeechFrame + 1;
    }
    else
    {
        if(nbSpeechFrame > MIN_SPEECH_FRAME_HANGOVER)
            hangOver = HANGOVER;                                .....( J.11 )
        nbSpeechFrame = 0;
        if(hangOver != 0)
        {
            hangOver = hangOver - 1;
            flagVAD_nest = 1;
        }
        else
            flagVAD_nest = 0;
    }
}

```

式中：

$SNR\_THRESHOLD\_VAD = 15;$   
 $MIN\_SPEECH\_FRAME\_HANGOVER = 4;$   
 $HANGOVER = 15;$   
 $nbSpeechFrame, meanEn, flagVAD_{nest}, hangOver$  初始化为 0。



$$H(bin, t) = \frac{\sqrt{\eta(bin, t)}}{1 + \sqrt{\eta(bin, t)}} \quad \dots \dots \dots \quad (J.16)$$

已知  $H(bin, t)$ , 便可对降噪信号谱估计进行更新:

更新后的先验信噪比  $\eta_2(bin, t)$  为：

式中：

$\eta_{TH} = 0.079\ 432\ 823$ (对应的 SNR 为-22 dB)。

相应地,滤波器传递函数  $H_2(bin, t)$  更新为:

$$H_2(bin, t) = \frac{\sqrt{\eta_2(bin, t)}}{1 + \sqrt{\eta_2(bin, t)}}, 0 \leq bin \leq N_{\text{SPEC}} - 1 \quad \dots \dots \dots \quad (\text{J.19})$$

根据  $H_2(bin, t)$  得出降噪信号谱  $P_{den_3}^{1/2}(bin, t)$ :

## J.8 Mel 濾波

首先对线性频率维纳滤波器系数  $H_2(bin)$ ,  $0 \leqslant bin \leqslant N_{\text{SPEC}} - 1$ , 作平滑处理, 之后转化为 Mel 频率刻度。通过对  $H_2(bin)$  作半重叠三角形频率窗处理后, 估计出 Mel 域维纳滤波器系数  $H_{2\_mel}(k)$ 。为了得出 Mel 子带的中心频率  $bin_{centr}(k)$ , 线性频率表  $f_{lin}$  通过下面的公式转化为 Mel 刻度:

$$MEL\{f_{lin}\} = 2595 \times \log_{10}(1 + f_{lin}/700) \quad \dots \dots \dots \quad (J.21)$$

第  $k$  子带的中心频率  $f_{mel}(k)$ :

$$f_{centr}(k) = 700 \times (10^{f_{mel}(k)/2.595} - 1), 1 \leq k \leq K_{FB} \quad \dots \dots \dots \quad (J.22)$$

式中：

$K_{FB}=32$ , 并且:

式中：

$f_{lin\_samp} = 16$  kHz——采样频率。

两个边缘子带的中心频率  $f_{centr}(0)$  和  $f_{centr}(K_{FB}+1) = f_{lin\_samp}/2$  加在  $K_{FB}=32$  子带上。因此，一共要计算  $K_{FB}+2=34$  个 Mel 域维纳滤波器系数。中心频率所对应的 FFT 频率为：

$$bin_{centr}(k) = round\left(\frac{f_{centr}(k)}{f_{lin\_samp}} \times 2 \times (N_{SPEC} - 1)\right) \quad \dots \dots \dots \quad (J.24)$$

下面计算三角形频率窗  $W(k, i)$ 。

$1 \leq k \leq K_{FB}$  的窗函数计算如下：

$$W(k, i) = \frac{i - bin_{centr}(k-1)}{bin_{centr}(k) - bin_{centr}(k-1)}, bin_{centr}(k-1) + 1 \leq i \leq bin_{centr}(k) \quad \dots\dots\dots (J.25)$$

$i$  取其他值时  $W(k,i) = 0$ .

$k=0$  的窗函数计算如下：

$$W(0, i) = 1 - \frac{i}{bin_{centr}(1) - bin_{centr}(0)}, 0 \leq i \leq bin_{centr}(1) - bin_{centr}(0) - 1 \quad \dots\dots\dots (J.27)$$

$i$  取其他值时  $W(0, i) = 0$ 。

$k = K_{FB} + 1$  的窗函数计算如下：

$$W(K_{FB} + 1, i) = \frac{i - bin_{centr}(K_{FB})}{bin_{centr}(K_{FB} + 1) - bin_{centr}(K_{FB})}, bin_{centr}(K_{FB}) + 1 \leq i \leq bin_{centr}(K_{FB} + 1) \quad \dots\dots\dots (J.28)$$

$i$  取其他值时  $W(K_{FB} + 1, i) = 0$ 。

Mel 域维纳滤波器系数  $H_{2\_mel}(k)$  在  $0 \leq k \leq K_{FB} + 1$  时, 计算公式如下:

$$H_{2\_mel}(k) = \frac{1}{\sum_{i=0}^{N_{SPEC}-1} W(k, i)} \sum_{i=0}^{N_{SPEC}-1} W(k, i) \times H_2(i) \quad \dots\dots\dots (J.29)$$

## J.9 增益调整



第一阶段降噪处理中, 根据降噪信号的功率谱  $P_{den3}(bin, t)$  计算降噪信号的能量  $E_{den}(t)$  如下:

$$E_{den}(t) = \sum_{bin=0}^{N_{SPEC}-1} P_{den3}^{1/2}(bin, t) \quad \dots\dots\dots (J.30)$$

第二阶段降噪处理中, 根据噪声功率谱  $P_{noise}(bin, t)$  计算噪声能量:

$$E_{noise}(t) = \sum_{bin=0}^{N_{SPEC}-1} P_{noise}^{1/2}(bin, t) \quad \dots\dots\dots (J.31)$$

通过连续三帧的降噪信号的能量和噪声能量可估计出平滑后的信噪比:

$$\begin{aligned} Ratio &= \frac{E_{den}(t-2) \times E_{den}(t-1) \times E_{den}(t)}{E_{noise}(t) \times E_{noise}(t) \times E_{noise}(t)}; \\ &if(Ratio > 0.000\ 01) \\ &\quad SNR_{over}(t) = 20/3 \times \log_{10}(Ratio); \\ &else \\ &\quad SNR_{over}(t) = -100/3; \end{aligned} \quad \dots\dots\dots (J.32)$$

为了估计第二阶段的降噪增益, 低信噪比跟踪值  $SNR_{low\_track}$  计算如下:

$$\begin{aligned} &if(((SNR_{over}(t) - SNR_{lower\_track}(t-1)) < 10 || (t < 10)) \\ &\quad SNR_{low\_track}(t) = \lambda_{SNR}(t) \times SNR_{low\_track}(t-1) + (1 - \lambda_{SNR}(t)) \times SNR_{over}(t); \\ &else \end{aligned} \quad \dots\dots\dots (J.33)$$

$$SNR_{low\_track}(t) = SNR_{low\_track}(t-1);$$

式中:

$SNR_{low\_track}$  —— 初始化为 0;

$\lambda_{SNR}(t)$  —— 遗忘因子, 计算如下:

$$\begin{aligned} &if(t < 10) \\ &\quad \lambda_{SNR}(t) = 1 - 1/t; \\ &else \\ &\quad if(SNR_{over}(t) < SNR_{low\_track}(t)) \\ &\quad \quad \lambda_{SNR}(t) = 0.95; \\ &\quad else \\ &\quad \quad \lambda_{SNR}(t) = 0.99; \end{aligned} \quad \dots\dots\dots (J.34)$$

增益调整主要目的在于：当处理纯噪声帧时，需采用相对较大的降噪增益；而处理包含语音的噪声帧时，需要采用相对较小的降噪增益。对当前信噪比估计  $SNR_{over}(t)$  和低信噪比跟踪值  $SNR_{low\_track}(t)$  进行比较，同时更新维纳滤波器增益调整系数  $\alpha_{GF}(t)$ ，计算如下：

$$\begin{aligned}
 & if(E_{den}(t) > 100) \\
 & \quad \{ \\
 & \quad \quad if(SNR_{over}(t) < (SNR_{low\_track}(t) + 3.5)) \\
 & \quad \quad \{ \\
 & \quad \quad \quad \alpha_{GF}(t) = \alpha_{GF}(t-1) + 0.15; \\
 & \quad \quad \quad if(\alpha_{GF}(t) > 0.8) \\
 & \quad \quad \quad \quad \alpha_{GF}(t) = 0.8; \\
 & \quad \quad \} \\
 & \quad \quad else \\
 & \quad \quad \{ \\
 & \quad \quad \quad \alpha_{GF}(t) = \alpha_{GF}(t-1) - 0.3; \\
 & \quad \quad \quad if(\alpha_{GF}(t) < 0.1) \\
 & \quad \quad \quad \quad \alpha_{GF}(t) = 0.1; \\
 & \quad \quad \} \\
 & \quad \} \\
 & \quad .....(J.35)
 \end{aligned}$$

式中：

$$\alpha_{GF}(0) = 0.8。$$

第二阶段的维纳滤波器系数乘以增益调整系数  $\alpha_{GF}(t)$ ：

$$H_{2\_mel\_GF}(k, t) = (1 - \alpha_{GF}(t)) + \alpha_{GF}(t) \times H_{2\_mel}(k, t), \quad 0 \leq k \leq K_{FB} + 1 \quad .....(J.36)$$

式中，系数  $\alpha_{GF}(t)$  的取值在 0.1~0.8 之间。

## J.10 Mel 域 IDCT

维纳滤波器的时域脉冲响应  $h_{WF}(n)$  通过 Mel 域维纳滤波器系数  $H_{2\_mel}(k)$  [第二阶段为  $H_{2\_mel\_GF}(k)$ ，见式(J.36)] 进行 Mel 域 IDCT 得到：

$$h_{WF}(n) = \sum_{k=0}^{K_{FB}+1} H_{2\_mel}(k) \times IDCT_{mel}(k, n), \quad 0 \leq n \leq K_{FB} + 1 \quad .....(J.37)$$

式中：

$IDCT_{mel}(k, n)$ ——Mel 域 IDCT 函数。

具体推导如下：

首先， $1 \leq k \leq K_{FB}$  频带的各自中心频率为：

$$f_{centr}(k) = \frac{1}{N_{SPEC}-1} \sum_{i=0}^{N_{SPEC}-1} W(k, i) \times i \times \frac{f_{samp}}{2 \times (N_{SPEC} - 1)} \quad .....(J.38)$$

式中：

$f_{samp}$ ——采样频率， $f_{samp} = 16$  kHz；

$$f_{centr}(0) = 0$$
 kHz；

$$f_{centr}(K_{FB} + 1) = f_{samp}/2。$$

则， $IDCT_{mel}(k, n)$  为：

$$IDCT_{mel}(k, n) = \cos\left(\frac{2 \times \pi \times n \times f_{centr}(k)}{f_{samp}}\right) \times df(k), 0 \leq k \leq K_{FB} + 1, \quad 0 \leq n \leq K_{FB} + 1$$

.....( J.39 )

式中：

$f_{centr}(k)$ ——Mel 子带  $k$  所对应的中心频率。

$df(k)$  为：

维纳滤波器的脉冲响应扩展到  $0 \leq k \leq 2 \times (K_{FB} + 1)$ :

$$h_{WF\_mirr}(n) = \begin{cases} h_{WF}(n), & 0 \leq n \leq K_{FB} + 1 \\ h_{WF}(2 \times (K_{FB} + 1) + 1 - n), & K_{FB} + 2 \leq n \leq 2 \times (K_{FB} + 1) \end{cases} \dots\dots (J.41)$$

J.11 维纳滤波

根据  $h_{WF\_mirr}(n)$  得出因果的脉冲响应  $h_{WF\_caus}(n, t)$ :

截断后脉冲响应  $F_{WF\_trunc(n,t)}$  为：

$$h_{WF-trunc}(n, t) = h_{WF-caus}(n + K_{FB} + 1 - (FL - 1)/2, t), \quad n = 0, \dots, FL - 1 \quad \dots\dots (J.43)$$

滤波器长度  $FL$  等于 17。截断后脉冲响应加汉宁窗处理：

$$h_{WF\_w}(n, t) = \left\{ 0.5 - 0.5 \times \cos\left(\frac{2 \times \pi \times (n + 0.5)}{FL}\right) \right\} \times h_{WF\_trunc}(n, t), \quad 0 \leq n \leq FL - 1$$

.....( J.44 )

这样,输入信号  $s_{in}$  经过脉冲响应  $h_{WF_w}(n, t)$  的维纳滤波器后就得到降噪信号  $s_{nr}$ :

$$s_{nr}(n) = \sum_{i=-(FL-1)/2}^{(FL-1)/2} h_{WF_w}(i + (FL - 1)/2) \times s_{in}(n - i), \quad 0 \leq n \leq M - 1 \quad .....( J.45 )$$

式中：

$FL = 17$  —— 滤波器长度；

$M=160$ ——帧移样本数。

## 参 考 文 献

- [1] GB/T 20090.10 信息技术 先进音视频编码 第10部分：移动语音与音频编码
- [2] 3GPP TS 26.290, Version 7.0.0, "Extended Adaptive Multi-Rate—Wideband (AMR-WB+) codec; Transcoding functions", Mar. 2007.
- [3] 3GPP TS 26.190, Version 7.0.0, " AMR Wideband speech codec; Transcoding functions ", Jun. 2007.
- [4] ETSI ES 202 050, Version 1.1.5, "Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm", Jan. 2007.
- [5] ETSI ES 202 212, Version 1.1.2,"Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm", Nov. 2005.
- [6] ISO/IEC IS 14496-1.Information Technology—Generic coding of audio-visual objects.Part 1: Systems. Nov.1998.
- [7] ISO/IEC IS 14496-2.Information Technology—Generic coding of audio-visual objects.Part 2: Visual. Nov.1998.
- [8] ISO/IETC JCT1/SC29 WG11 N3342. Overview of MPEG-7 standard. Maui,1999.
- [9] ISO/IEC JCT1/SC29 WG11 and ITU-T SG26 Q.6 (JVT-K051, Version3 of ISO/IEC 14496-10E). 12th Meeting Redmond, U.S.A, Jul.2004.
- [10] 姚天任, 孙洪. 现代数字信号处理[M]. 武汉: 华中理工大学出版社,1999.
- [11] A.V.奥本海姆, R.W. 谢弗. 离散时间信号处理[M]. 2 版. 刘树棠, 黄建国, 译. 西安: 西安交通大学出版社,2001.
- [12] 鲍长春. 低比特率数字语音通信编码基础[M]. 北京: 北京工业大学出版社,2001.
- [13] 张贤达, 现代信号处理[M]. 2 版. 北京: 清华大学出版社,2002.
- [14] 赵力. 语音信号处理[M]. 北京: 机械工业出版社,2003.
- [15] 夸特尔瑞. 离散时间语音信号处理——原理与应用[M]. 赵胜辉等, 译. 北京: 电子工业出版社,2004.
- [16] 王炳锡, 王洪. 变速率语音编码[M]. 西安: 西安电子科技大学出版社,2004.
- [17] 胡航. 语音信号处理[M]. 3 版. 哈尔滨: 哈尔滨工业大学出版社,2005.
- [18] 程佩清. 数字信号处理教程[M]. 3 版. 北京: 清华大学出版社,2007.
- [19] 吴家安, 现代语音编码技术[M]. 北京: 科学出版社,2007.
- [20] 万帅, 杨付正. 新一代高效视频编码 H.265/HEVC[M]. 北京: 电子工业出版社,2014.
- [21] Kenneth.R.Castleman. Digital Image Processing. 北京: 清华大学出版社. 1998.
- [22] ITU-T Draft Recommendation and Final Draft International Standard of JointVideo Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). 7th Meeting: Pattaya, Thailand, 7-14 Mar.2003.
- [23] Abdul H. Sadka. Compressed Video Communications. Hohn Wiley & Sons, Ltd. England. 2002.
- [24] Iain E.G.Richardson. H.264 and MPEG-4 Video Compression. Hohn Wiley & Sons, Ltd. England. 2002.
- [25] J.H. Conway and N.J.A. Sloane, "A fast encoding method for lattice codes and quantizers," IEEE Trans. Inform. Theory, vol. IT-29, no. 6, pp. 820-824, Nov. 1983.

- [26] F. Jabloun and A.E. Cetin, "The Teager Energy Based Feature Parameters for Robust Speech Recognition in Noise," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Mar. 1999.
- [27] L.B. Almeida and F.M. Silva, "Variable—Frequency Synthesis: An Improved Harmonic Coding Scheme," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, San Diego, CA, May 1984.
- [28] M. Xie and J. P. Adoul, " Embedded algebraic vector quantization (EAVQ) with application to wideband audio coding," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Atlanta, GA, U.S.A, vol. 1, pp. 240-243, 1996.
- [29] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations on the speaker verification task," 10th International Conference on Speech and Computer (SPECOM), Vol. 1, pp. 191-194, 2005.
- [30] Adrian Grange, Peter de Rivaz, and Jonathan Hunt, "VP9 Bitstream & Decoding Process Specification", 2016.

