

# 客户端 Deinterlace 库

# API 参考

文档版本 04

发布日期 2008-09-05

BOM编码 N/A

深圳市海思半导体有限公司为客户提供全方位的技术支持,用户可与就近的海思办事处联系,也可直接与公 司总部联系。

### 深圳市海思半导体有限公司

地址: 深圳市龙岗区坂田华为基地华为电气生产中心 邮编: 518129

网址: http://www.hisilicon.com

客户服务电话: 0755-28788858

客户服务传真: 0755-28357515

客户服务邮箱: support@hisilicon.com

#### 版权所有 © 深圳市海思半导体有限公司 2007-2008。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式 传播。

#### 商标声明



(上) THISILICON、海思,均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

#### 注意

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导, 本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。



# 目 录

i

刖 音	
1 概述	1-1
1.1 Deinterlace库概述	
1.2 文档说明	1-2
1.3 Deinterlace开发包	1-2
2 API参考	2-1
2.1 Deinterlace初始化和释放函数	2-2
2.2 Deinterlace主函数	2-2
2.3 接口调用流程	
3 API函数	3-1
3.1 HI_InitDeinterlace	3-2
3.2 HI_ReleaseDeinterlace	3-3
3.3 HI_Deinterlace	3-4
3.4 HI_GetVersion	3-6
3.5 HI_SetOsd	3-7
4 其他信息	4-1
4.1 通用结构及数据类型定义	4-2
4.2 烘湿缸	4.3



# 插图目录

图 2-1 接口调用流程图......2-3



# 前言

## 概述

本节介绍本文档的内容、对应的产品版本、适用的读者对象、行文表达约定、历史修订记录等。

# 产品版本

与本文档相对应的产品版本如下所示。

产品名称	产品版本
Hi3510 通信媒体处理器	V100
Hi3511 H.264 编解码处理器	V100
Hi3512 H.264 编解码处理器	V100

## 读者对象

本参考描述了基于 Deinterlace 库开发的各种参考信息。使用本参考的程序员应该:

- 熟练使用 C++语言
- 掌握基本的 Windows32 调用

# 内容简介

本参考首先简单概述了 Deinterlace 库的功能以及 Deinterlace API 的分类, 然后详细介绍了 Deinterlace API 参考信息及接口函数。本参考内容组织如下。

章节	内容
1 概述	简单介绍 Deinterlace 库的功能、Deinterlace API 的分类以及开发包的组成。



章节	内容
2 API 参考	详细介绍 Deinterlace API 参考信息。
3 API 函数	详细介绍 Deinterlace API 接口函数。
4 其他信息	简单介绍通用结构及数据类型定义和错误码。

# 约定

## 符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

符号	说明
<b>企</b> 危险	以本标志开始的文本表示有高度潜在危险,如果不能避免,会导致人员死亡或严重伤害。
警告	以本标志开始的文本表示有中度或低度潜在危险,如果不 能避免,可能导致人员轻微或中等伤害。
注意	以本标志开始的文本表示有潜在风险,如果忽视这些文本,可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
◎型 窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
□ 说明	以本标志开始的文本是正文的附加信息,是对正文的强调 和补充。

### 通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体,并且在内容前后增加线条与正文隔离。



格式	说明
"Terminal Display"格式	"Terminal Display"格式表示屏幕输出信息。此外,屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。

# 修改记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修改日期	版本	修改说明
2008-09-05	04	增加本文档适用的 Hi3512 产品版本。
2008-04-21	03	增加本文档适用的 Hi3511 产品版本。
2007-11-30	02	<ul> <li>1.3 Deinterlace 开发包: 删除了"Deinterlace 效果比较说明文件"。</li> <li>3 API 函数: 增加了"HI_SetOsd" API 函数。</li> <li>4.1 通用结构及数据类型定义: 增加了"DEINTERLACE_OSDRECT_S"结构体。</li> <li>4.2 错误码: 增加了"HI_ERR_OSDNUM"错误码。</li> </ul>
2007-04-20	01	第 1 次发布

**1** 概述

# 关于本章

本章描述内容如下表所示。

标题	内容
1.1 Deinterlace 库概述	介绍 Deinterlace 库的功能和特性。
1.2 文档说明	列举此文档包含的 API。
1.3 Deinterlace 开发包	介绍 Deinterlace 开发包包含的文档。

## 1.1 Deinterlace 库概述

目前,电视节目仍采用 Interlace 视频(场交错成帧)信号,视频采集设备也都是隔行扫描方式。由于组成一帧的奇场、偶场存在时间差,Interlace 视频在逐行显示器播放过程中会出现锯齿、拉丝等图像质量问题,这就需要在播放前对视频进行去交错(Deinterlace)处理。

海思采集的 D1 图象按照场进行编码,在 PC 端解码后,需要对图像进行 Deinterlace 处理。海思提供了进行 Deinterlace 处理的库,即 DllDeinterlace.lib。

DllDeinterlace.lib 是 Windows 下的静态库,完成把隔行图像转换为逐行图像(即两场图像形成一帧图像)的功能,输入解码后的 YUV420 场图像数据,每两场输出一帧 YUV420 图像。Deinterlace 库具有如下优势:

#### ● 静止画面清晰

静止画面垂直分辨率达到 500 线强,较好的保持了纹理细节,图像边缘比较清晰锐利,消除了最容易出现的边缘锯齿现象。

● 运动画面无拉丝

运动物体的边缘过渡平滑,锯齿不明显。对于剧烈运动的物体,Deinterlace 库较好的解决了最容易出现的拉丝问题。

• 增加了抗闪烁、抗噪功能

Deinterlace 库增加了抗闪烁功能,图像画面比较稳定;增加了抗噪功能,对于噪点较多的图像,Deinterlace 库的优势会特别明显。

- 较好的解决了丢场问题
  - 一些特殊的专业测试码流,物体影像单独存在于奇场或偶场。以往的一些方案面 对这样的特殊情况会丢失奇场信息,Deinterlace 库较好的解决了这个问题。
- 效率满足工程应用的需求

实际工程应用中,如果侧重图像质量,势必影响图像处理效率;如果侧重图像处理效率,图像质量势必下降。Deinterlace 库在保证图像处理质量的基础上,图像处理效率也达到令人满意的程度。

## 1.2 文档说明

此文档提供了 Deinterlace 一整套 API 的说明,包括:

- 初始化 Deinterlace API。
- 释放 Deinterlace API。
- Deinterlace API.

## 1.3 Deinterlace 开发包

Deinterlace 开发包包含以下 5 个部分:



- 头文件: DllDeinterlace.h
- 静态库文件: DllDeinterlace.lib
- 动态库文件: DllDeinterlace.lib、DllDeinterlace.dll
- API 调用示范代码文件: sample.cpp
- Deinterlace 应用参考文件:客户端 Deinterlace 库 API 参考.pdf



# **2** API 参考

# 关于本章

本章描述内容如下表所示。

标题	内容
2.1 Deinterlace 初始化和释放函数	介绍 Deinterlace 初始化和释放函数的功能。
2.2 Deinterlace 主函数	介绍 Deinterlace 主函数的功能以及注意事项。
2.3 接口调用流程	介绍 Deinterlace 接口的调用流程。



## 2.1 Deinterlace 初始化和释放函数

Deinterlace 初始化函数(HI\_InitDeinterlace)在第一次执行 Deinterlace 处理前,分配 Deinterlace 空间、初始化 Deinterlace 相关变量及状态,设置输入场图像的高宽、跨度信息和输出帧图像的跨度信息。初始化成功则返回 Deinterlace 句柄。

结束 Deinterlace 处理时,Deinterlace 释放函数(HI\_ReleaseDeinterlace)释放 Deinterlace 空间,防止内存泄漏。



#### 注意

调用 Deinterlace 释放函数后,还需把 Deinterlace 句柄置为空,避免后面错误的引用该句柄。

## 2.2 Deinterlace 主函数

Deinterlace 主函数(HI\_Deinterlace)完成两场 YUV420 图像输出一帧 YUV420 图像功能。输入 YUV420 场图像的 Y、U、V 三个分量,输出则分别为存储空间扩大了一倍的 YUV420 帧图像的 Y、U、V 三分量。

解码完成后可直接调用 Deinterlace 主函数,每两场会输出一帧图像。



#### 注音

调用 Deinterlace 主函数前,需给奇、偶场标志赋值,每次循环调用 Deinterlace 接口时需间隔输入奇、偶、奇、偶场标志。

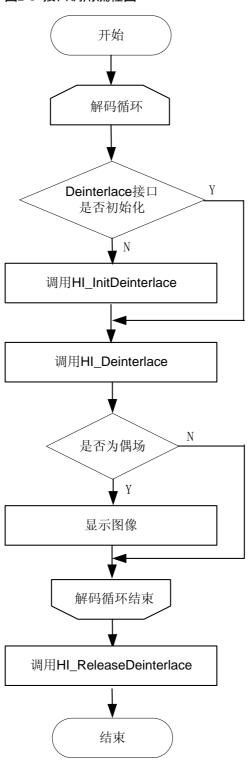
#### □ 说明

- 1. 文中的"解码完成"指的是完成每一场图像解码。
- 2. 奇场也称为顶场, 偶场也称为底场。

## 2.3 接口调用流程

Deinterlace 接口的调用流程如图 2-1 所示。

#### 图2-1 接口调用流程图



若第一次解码完成后执行 Deinterlace 处理,需先调用 Deinterlace 初始化函数,然后再调用 Deinterlace 主函数;否则不需要先调用 Deinterlace 初始化函数,解码完成后可以直接调用 Deinterlace 主函数。



在调用 Deinterlace 初始化函数前,需要把输入的场图像高宽、跨度信息以及输出帧图像的跨度信息配置进输入结构体参数中。Deinterlace 库内部会保存这些信息,并开辟相应的 Deinterlace 空间,供 Deinterlace 主函数使用。

在每次调用 Deinterlace 主函数前,需给奇、偶场标志赋值,Deinterlace 库会根据输入的奇、偶场标志,每偶场输出一帧图像。如果输入的场标志不符合要求,则不做任何处理,直接返回错误码。

结束 Deinterlace 处理时,调用 Deinterlace 释放函数,释放所开辟的 Deinterlace 空间。



#### 注意

在 Deinterlace 库外部开辟的输出帧的 YUV 每个分量的 Buffer 至少应该为相应跨度 × 场高 × 2。

调用 Deinterlace 释放函数此接口后,还需要把 Deinterlace 句柄置为空。



# **3** API 函数

# 关于本章

本章描述内容如下表所示。

标题	内容
3.1 HI_InitDeinterlace	介绍 HI_InitDeinterlace 函数。
3.2 HI_ReleaseDeinterlace	介绍 HI_ReleaseDeinterlace 函数。
3.3 HI_Deinterlace	介绍 HI_Deinterlace 函数。
3.4 HI_GetVersion	介绍 HI_GetVersion 函数。
3.5 HI_SetOsd	介绍 HI_SetOsd 函数。

# 3.1 HI\_InitDeinterlace

#### 【目的】

初始化 Deinterlace 接口。

#### 【语法】

int HI\_InitDeinterlace(void \*\*pHandle, DEINTERLACE\_PARA\_S struPara);

#### 【描述】

- 初始化 Deinterlace 接口。
- 创建 Deinterlace 句柄。
- 保存输入参数。

#### 【参数】

参数名称	描述	输入/输出	全局/局部
**pHandle	Deinterlace 句柄,输入的是指针的指针类型,该句柄要返回给用户使用,后面的Deinterlace 主函数和释放函数都会用到此句柄。该句柄的作用是用来区分各个Deinterlace 库,防止播放多路 D1 时各个库的串扰。	输出	局部
struPara	配置输入 YUV420 场图像的高宽、跨度参数,配置输出 YUV420 帧图像的跨度参数。在库中,要求输入场图像宽度至少为128,输入场图像高度至少为4,输入的 Y分量跨度至少为图像的宽度,UV 分量的跨度至少为图像宽度的一半,输出图像的跨度要求同上。详细信息请参见DEINTERLACE_PARA_S 结构体。	输入	局部

#### 【返回值】

返回值	描述
0	成功
非 0	失败,其值为错误码

#### 【错误码】



接口返回值	含义
HI_SUCCESS_DEINTERLACE	成功
HI_ERR_MALLOC	创建空间失败
HI_ERR_PITCH	输入、输出图像的高宽跨度设置不符合要求

#### 【需求】

头文件: DllDeinterlace.h库文件: DllDeinterlace.lib

#### 【注意】

调用此函数前 Deinterlace 句柄要置为空。

#### 【举例】

请参考 sample.cpp 中的代码。

# 3.2 HI\_ReleaseDeinterlace

#### 【目的】

释放 Deinterlace 接口。

#### 【语法】

int HI\_ReleaseDeinterlace(void \*pHandle);

#### 【描述】

释放 Deinterlace 接口,完成 Deinterlace 空间的释放。

#### 【参数】

参数名称	描述	输入/输出	全局/局部
*pHandle	Deinterlace 句柄(即初始化时创建的句柄,下同),用来区分各个 Deinterlace 库,防止播放多路 D1 时各个库的串扰。	输入	局部

#### 【返回值】

返回值	描述
0	成功
非 0	失败, 其值为错误码

#### 【错误码】

接口返回值	含义
HI_SUCCESS_DEINTERLACE	成功
HI_ERR_POINT_NULL	输入指针为空

#### 【需求】

● 头文件: DllDeinterlace.h

• 库文件: DllDeinterlace.lib

#### 【注意】

调用此函数后 Deinterlace 句柄要置为空。

#### 【举例】

请参考 sample.cpp 中的代码。

## 3.3 HI\_Deinterlace

#### 【目的】

完成两场输出一帧图像功能,输入 YUV420 场图像,输出 YUV420 帧图像。

#### 【语法】

```
int HI_Deinterlace(
    void *pHandle,
    DEINTERLACE_FRAME_S struDstFrame,
    unsigned char *pszSrcY,
    unsigned char *pszSrcU,
    unsigned char *pszSrcV,
    PIC_TYPE_E tFieldFlag
);
```

#### 【描述】

执行 Deinterlace 处理,即完成两场输出一帧图像功能,输入 YUV420 场图像,输出 YUV420 帧图像。解码完成后可直接调用此函数。

#### □ 说明

针对显卡支持的图像格式,在视频显示中可以把 YUV420 转换为相应的图像格式,其转换目的空间可直接为显存。

#### 【参数】



参数名称	描述	输入/输出	全局/局部
*pHandle	Deinterlace 句柄,用来区分各个 Deinterlace 库,防止播放多路 D1 时各个 库的串扰。	输入	局部
struDstFrame	输出帧图像。 三个成员分别是输出帧图像的 Y、U、V 分量地址。详细信息请参见 DEINTERLACE_FRAME_S 结构体。	输出	局部
*pszSrcY	输入 YUV420 场图像的 Y 分量地址。	输入	局部
*pszSrcU	输入 YUV420 场图像的 U 分量地址。	输入	局部
*pszSrcV	输入 YUV420 场图像的 V 分量地址。	输入	局部
tFieldFlag	PIC_TYPE_E 枚举类型。 奇、偶场标志,输入底场时才有帧图像的 输出,顶场不做处理,直接返回,不输出 图像。	输入	局部

#### 【返回值】

返回值	描述
0	成功
非 0	失败,其值为错误码

#### 【错误码】

接口返回值	含义
HI_SUCCESS_DEINTERLACE	成功
HI_ERR_POINT_NULL	输入指针为空
HI_ERR_FIELD_FLAG	输入的不是奇、偶场标志

#### 【需求】

头文件: DllDeinterlace.h库文件: DllDeinterlace.lib

#### 【注意】

调用此函数前需给奇、偶场标志赋值。

#### 【举例】

请参考 sample.cpp 中的代码。

# 3.4 HI\_GetVersion

#### 【目的】

获取 Deinterlace 版本号。

#### 【语法】

int HI\_GetVersion(char \*\*pszVersion);

#### 【描述】

获取 Deinterlace 版本号接口,输出为 Deinterlace 版本号。

#### 【参数】

参数名称	描述	输入/输 出	全局/局部
**pszVersion	输出 Deinterlace 版本号字符串。	输出	局部

#### 【返回值】

返回值	描述
0	成功
非 0	失败,其值为错误码

#### 【错误码】

接口返回值	含义
HI_SUCCESS_DEINTERLACE	成功
HI_ERR_POINT_NULL	输入指针为空

#### 【需求】

头文件: DllDeinterlace.h库文件: DllDeinterlace.lib

#### 【注意】

输入参数为字符串指针的地址。



#### 【举例】

char \*pszVersion = NULL;
pszVersion = new char[256];
HI\_GetVersion((char \*\*)&pszVersion);

#### □ 说明

详细信息可参考 sample.cpp 代码。

## 3.5 HI\_SetOsd

#### 【目的】

设置 OSD 矩形区域。

#### 【语法】

int HI SetOsd(void \*pHandle, DEINTERLACE OSDRECT S rcOsd[], int iOsdNum);

#### 【描述】

- 调用 HI SetOsd 接口时, OSD 矩形区域必须同时满足以下 4 个条件:
  - 设置的 OSD 矩形区域必须为半透明或非透明。
  - 从板端 OSD 矩形区域传递 OSD 矩形区域的位置信息,并且 OSD 矩形区域放置在视频的左上角或右上角。
  - OSD 矩形区域必须是文字或时间 OSD。
  - OSD 矩形区域不能太大, 否则会影响 Deinterlace 性能。
- 当设置好 OSD 矩形区域,Deinterlace 会针对该区域进行专门的处理,达到图像质量优化的效果。
- 最多支持 4 个 OSD, 且 OSD 矩形区域必须在视频图像区域内。

#### 【参数】

参数名称	描述	输入/输出	全局/局部
*pHandle	Deinterlace 句柄。	输入	局部
rcOsd [ ]	输入 OSD 矩形区域数组。详细信息请参见 DEINTERLACE_OSDRECT_S 结构体。	输入	局部
iOsdNum	输入 OSD 矩形区域数组个数,最多支持 4 个。	输入	局部

#### 【返回值】

返回值	描述
0	成功



返回值	描述
非 0	失败,其值为错误码

#### 【错误码】

接口返回值	含义
HI_SUCCESS_DEINTERLACE	成功
HI_ERR_POINT_NULL	输入指针为空
HI_ERR_OSDNUM	输入的 OSD 个数不符合要求

#### 【需求】

头文件: DllDeinterlace.h库文件: DllDeinterlace.lib

#### 【举例】

```
DEINTERLACE_OSDRECT_S rcOsd[1];
rcOsd[0].x = 515;
rcOsd[0].y = 0;
rcOsd[0].w = 185;
rcOsd[0].h = 20;
HI_SetOsd(m_pHandle, rcOsd, 1);
```



# 4 其他信息

# 关于本章

本章描述内容如下表所示。

标题	内容
4.1 通用结构及数据类型定义	详细描述了 API 中涉及到的通用结构及数据类型的定义。
4.2 错误码	列举了 API 中涉及到的错误码。

#### □ 说明

详细信息请参考头文件 DllDeinterlace.h。



## 4.1 通用结构及数据类型定义

```
PIC_TYPE_E
```

/\*\*\*\*\*\*\*\*\*\*\*\*\*

#### DEINTERLACE\_PARA\_S

```
*此结构体提供初始化Deinterlace时的输入参数,成员值都需调用者配置。
```

- \*成员包括输入场图像的高宽、YUV分量的跨度信息。
- \*成员包括输出帧图像YUV分量的跨度信息。

#### DEINTERLACE FRAME S



```
unsigned char *pszV; //V分量地址 }DEINTERLACE FRAME S;
```

#### DEINTERLACE\_OSDRECT\_S

## 4.2 错误码