

## ffds show 源代码分析 2：位图覆盖滤镜（对话框部分Dialog）

2013年10月24日 21:30:44 阅读数：7025

ffds show源代码分析系列文章列表：

[ffds show 源代码分析 1：整体结构](#)

[ffds show 源代码分析 2：位图覆盖滤镜（对话框部分Dialog）](#)

[ffds show 源代码分析 3：位图覆盖滤镜（设置部分Settings）](#)

[ffds show 源代码分析 4：位图覆盖滤镜（滤镜部分Filter）](#)

[ffds show 源代码分析 5：位图覆盖滤镜（总结）](#)

[ffds show 源代码分析 6：对解码器的dll的封装（libavcodec）](#)

[ffds show 源代码分析 7：libavcodec视频解码器类（TvideoCodecLibavcodec）](#)

[ffds show 源代码分析 8：视频解码器类（TvideoCodecDec）](#)

[ffds show 源代码分析 9：编解码器有关类的总结](#)

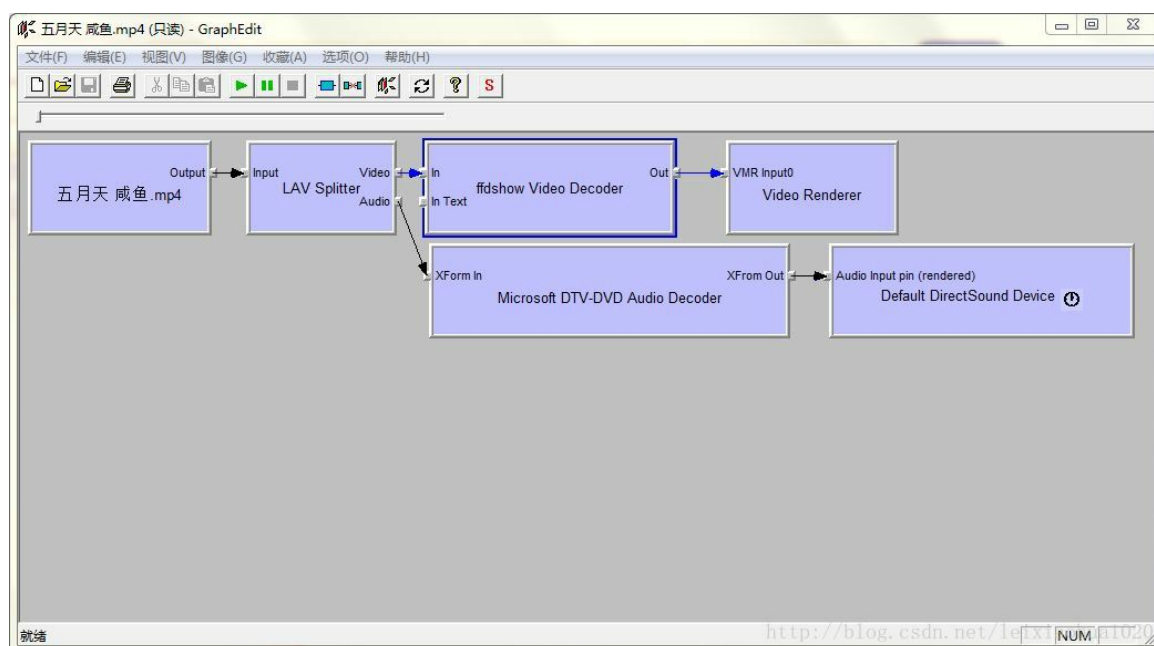


本文我们介绍ffds show的滤镜功能。ffds show支持很多种滤镜，可以支持多种视频和音频的后期效果。例如OSD滤镜支持在视频的左上角显示视频相关的信息。而可视化滤镜则支持显示视频每一帧的运动矢量以及量化参数。在这里我们介绍一种位图覆盖（Bitmap）滤镜（Filter）。

### 效果

编译完ffds show之后，在“项目属性->调试->命令”里面将GraphEdit.exe所在位置设置为调试程序，例如在这里我设置了《终极解码》里面自带Graph Edit.exe，路径为“C:\Program Files\Final Codecs\Codecs\GraphEdit.exe”。这样就可以使用GraphEdit.exe调试ffds show了。

向GraphEdit.exe里面拖入一个文件“五月天 咸鱼.mp4”，结果如下图所示：



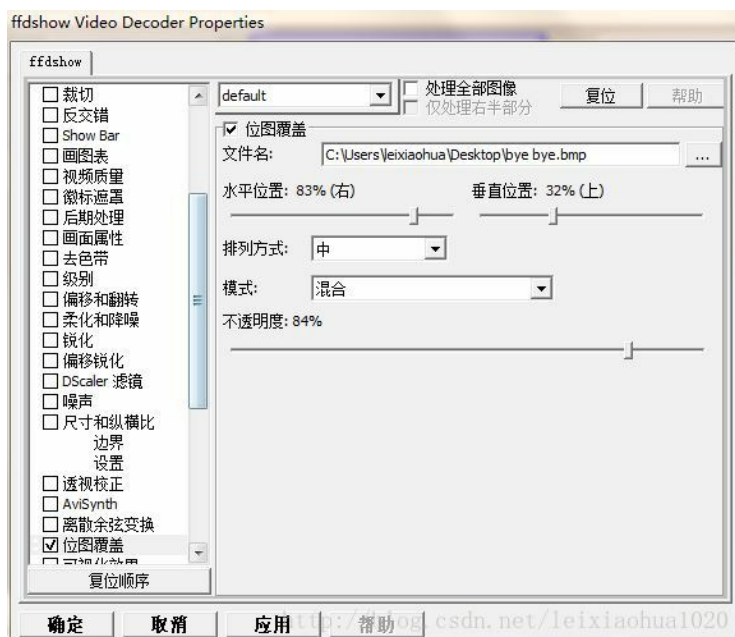
注：有的时候默认的视频解码器可能不是ffds show，可能是CoreAVC等，可以先删除视频解码器然后添加ffds show。

点击绿色三角形按钮就可以开始播放视频。

右键点击ffdshow组件，打开属性对话框之后，可以看见右边栏中有很多的滤镜。

勾选“位图覆盖”滤镜，然后选择一张用于覆盖的图片（在这里我选择了一张bmp格式的专辑封面）。

注：可以调整位图所在的水平位置，垂直位置，不透明度，并且可以修改位图叠加模式（在这里用混合）。



添加了该滤镜之后，播放窗口的显示内容为：

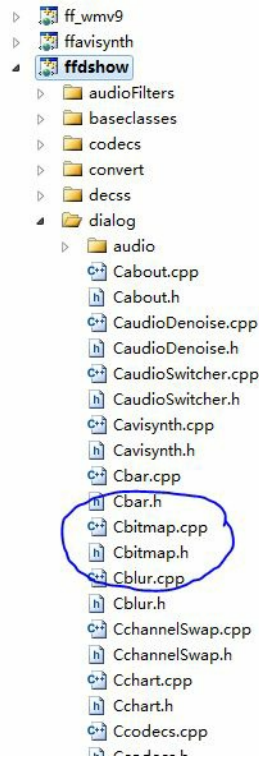


可见在右上角显示出了叠加的位图。

## 源代码分析

### 1.对话框部分

与位图覆盖（Bitmap）滤镜的对话框有关的类位于dialog目录下的Cbitmap.cpp和Cbitmap.h文件中。



先来看看Cbitmap.h中类的声明：

需要注意的是，里面类的名字居然叫TbitmapPage，而没有和头文件名字一致。= =

```
[cpp]
1.  #ifndef _CBITMAPPAGE_H_
2.  #define _CBITMAPPAGE_H_
3.
4.  #include "TconfPageDecVideo.h"
5.  //Bitmap配置页面
6.  class TbitmapPage : public TconfPageDecVideo
7.  {
8.  private:
9.      void pos2dlg(void), opacity2dlg(void);
10.     //设置文件路径
11.     void onFlm(void);
12. protected:
13.     virtual INT_PTR msgProc(UINT uMsg, WPARAM wParam, LPARAM lParam);
14. public:
15.     //构造函数
16.     TbitmapPage(TffdshowPageDec *Iparent, const TfilterIDFF *idff);
17.     //初始化
18.     virtual void init(void);
19.     //配置数据传入到对话框界面
20.     virtual void cfg2dlg(void);
21.     virtual void translate(void);
22. };
23.
24. #endif
```

再看看Cbitmap.cpp文件吧。关键的代码都已经加上了注释。

```
[cpp]
1.  /*
2.   * Copyright (c) 2004-2006 Milan Cutka
3.   *
4.   * This program is free software; you can redistribute it and/or modify
5.   * it under the terms of the GNU General Public License as published by
6.   * the Free Software Foundation; either version 2 of the License, or
7.   * (at your option) any later version.
8.   *
9.   * This program is distributed in the hope that it will be useful,
10.   * but WITHOUT ANY WARRANTY; without even the implied warranty of
11.   * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12.   * GNU General Public License for more details.
13.   *
14.   * You should have received a copy of the GNU General Public License
15.   * along with this program; if not, write to the Free Software
16.   * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
17.   */
18. //Bitmap配置页面
19. #include "stdafx.h"
20. #include "TsubtitlesSettings.h"
```

```

21. #include "tbitmapSettings.h"
22. #include "Cbitmap.h"
23. //初始化
24. void TbitmapPage::init(void)
25. {
26.     //设置滑动条范围
27.     edLimitText(IDC_ED_BITMAP_FLNM, MAX_PATH);
28.     tbrSetRange(IDC_TBR_BITMAP_POSX, 0, 100, 10);
29.     tbrSetRange(IDC_TBR_BITMAP_POSY, 0, 100, 10);
30.     tbrSetRange(IDC_TBR_BITMAP_OPACITY, 0, 256);
31. }
32. //配置数据传入到对话框界面
33. void TbitmapPage::cfg2dlg(void)
34. {
35.     //各种设置
36.     //EditControl设置
37.     setDlgItemText(m_hwnd, IDC_ED_BITMAP_FLNM, cfgGetStr(IDFF_bitmapFlnm));
38.     pos2dlg();
39.     cbxSetCurSel(IDC_CBX_BITMAP_ALIGN, cfgGet(IDFF_bitmapAlign));
40.     cbxSetCurSel(IDC_CBX_BITMAP_MODE, cfgGet(IDFF_bitmapMode));
41.     opacity2dlg();
42. }
43. //Bitmap位置信息
44. void TbitmapPage::pos2dlg(void)
45. {
46.     char_t s[260];
47.     int x;
48.     //获取
49.     x = cfgGet(IDFF_bitmapPosx);
50.     TsubtitlesSettings::getPosHoriz(x, s, this, IDC_LBL_BITMAP_POSX, countof(s));
51.     setDlgItemText(m_hwnd, IDC_LBL_BITMAP_POSX, s);
52.     //设置
53.     tbrSet(IDC_TBR_BITMAP_POSX, x);
54.
55.     x = cfgGet(IDFF_bitmapPosy);
56.     TsubtitlesSettings::getPosVert(x, s, this, IDC_LBL_BITMAP_POSY, countof(s));
57.     setDlgItemText(m_hwnd, IDC_LBL_BITMAP_POSY, s);
58.     tbrSet(IDC_TBR_BITMAP_POSY, x);
59. }
60. void TbitmapPage::opacity2dlg(void)
61. {
62.     int o = cfgGet(IDFF_bitmapStrength);
63.     tbrSet(IDC_TBR_BITMAP_OPACITY, o);
64.     setText(IDC_LBL_BITMAP_OPACITY, _L("%s %i%%"), _(IDC_LBL_BITMAP_OPACITY), 100 * o / 256);
65. }
66.
67. INT_PTR TbitmapPage::msgProc(UINT uMsg, WPARAM wParam, LPARAM lParam)
68. {
69.     switch (uMsg) {
70.         case WM_COMMAND:
71.             switch (LOWORD(wParam)) {
72.                 case IDC_ED_BITMAP_FLNM:
73.                     if (HIWORD(wParam) == EN_CHANGE && !isSetWindowText) {
74.                         char_t flnm[MAX_PATH];
75.                         GetDlgItemText(m_hwnd, IDC_ED_BITMAP_FLNM, flnm, MAX_PATH);
76.                         cfgSet(IDFF_bitmapFlnm, flnm);
77.                     }
78.                     return TRUE;
79.             }
80.             break;
81.         }
82.     return TconfPageDecVideo::msgProc(uMsg, wParam, lParam);
83. }
84. //设置文件路径
85. void TbitmapPage::onFlnm(void)
86. {
87.     char_t flnm[MAX_PATH];
88.     cfgGet(IDFF_bitmapFlnm, flnm, MAX_PATH);
89.     if (dlgGetFile(false, m_hwnd, _L(
ID3_BITMAP, _L("Load image file")), _L("All supported (*.jpg;*.bmp;*.gif;*.png)\0*.bmp;*.jpg;*.jpeg;*.gif;*.png\0Windows Bitmap (*.bm
\0*.bmp\0JPEG (*.jpg)\0*.jpg\0CompuServe Graphics Interchange (*.gif)\0*.gif\0Portable Network Graphics (*.png)\0*.png"), _L("bmp"),
flnm, _L("."), 0)) {
90.         setDlgItemText(m_hwnd, IDC_ED_BITMAP_FLNM, flnm);
91.         //设置
92.         cfgSet(IDFF_bitmapFlnm, flnm);
93.     }
94. }
95.
96. void TbitmapPage::translate(void)
97. {
98.     TconfPageBase::translate();
99.
100.     cbxTranslate(IDC_CBX_BITMAP_ALIGN, TsubtitlesSettings::alignments);
101.     cbxTranslate(IDC_CBX_BITMAP_MODE, TbitmapSettings::modes);
102. }
103. //构造函数
104. TbitmapPage::TbitmapPage(TfddshowPageDec *Iparent, const TfilterIDFF *idff): TconfPageDecVideo(Iparent, idff)
105. {
106.     //各种绑定
107.     resInter = IDC_CHB_BITMAP;
108.     static const TbindTrackbar<TbitmapPage> htbr[] = {
109.         IDC_TBR_BITMAP_POSX, IDFF_bitmapPosx, &TbitmapPage::pos2dlg,

```

```

109.         IDC_TBR_BITMAP_POSX, IDFF_bitmapPosX, &tbitmapPage::pos2dlg,
110.         IDC_TBR_BITMAP_POSY, IDFF_bitmapPosy, &tbitmapPage::pos2dlg,
111.         IDC_TBR_BITMAP_OPACITY, IDFF_bitmapStrength, &tbitmapPage::opacity2dlg,
112.         0, 0, NULL
113.     };
114.     bindHtracks(htbr);
115.     static const TbindCombobox<TbitmapPage> cbx[] = {
116.         IDC_CBX_BITMAP_ALIGN, IDFF_bitmapAlign, BINDCBX_SEL, NULL,
117.         IDC_CBX_BITMAP_MODE, IDFF_bitmapMode, BINDCBX_SEL, NULL,
118.         0
119.     };
120.     bindComboboxes(cbx);
121.     static const TbindButton<TbitmapPage> bt[] = {
122.         IDC_BT_BITMAP_FLNM, &tbitmapPage::onFlnm,
123.         0, NULL
124.     };
125.     bindButtons(bt);
126. }

```

看ffdshow源代码的时候，开始会比较费劲。为什么？因为它使用了大量自己写的API函数，以及自己定义的结构体。这些API函数的种类繁多，如果一个一个都看完，估计就精疲力竭了。经过一段时间的学习之后，我发现最方便的方法还是根据函数名字推测其用法。因此我就不深入剖析ffdshow的API函数了。

以上源代码中包含以下API（大致按出现先后次序，可能没有例举全，在这里只是举例子）：

```

1.  edLimitText();//限制输入字符串长度
2.  tbrSetRange();//设置滑动条范围
3.  setDlgItemText();//设置组件名称
4.  cbxSetCurSel();//设置下拉框当前选项
5.  cfgGet();//从注册表中读取变量的值
6.  tbrSet();//设置滑动条的值
7.  bindHtracks();//绑定注册表变量和滑动条
8.  bindComboboxes();//绑定注册表变量和下拉框
9.  bindButtons();//绑定函数和按钮

```

从以上函数大致可以看出tbr\*\*\*()基本上都是操作滑动条的，cbx\*\*\*()基本上都是操作下拉框的，函数基本上可以从名称上理解其意思。bind\*\*\*()就是绑定注册表变量和控件的。注意ffdshow里面有注册表变量这么一个概念。这些变量的值存在系统的注册表里面，不会因为程序结束运行而消失。就目前我的观察来看，绝大部分注册表变量存的是一个整数值。这些注册表变量都以IDFF\_xxx的名称预编译定义在ffdshow\_constants.h头文件中。与MFC控件可以直接与CString，int等变量绑定不同，ffdshow控件只可以和注册表变量绑定。即每次运行的时候都从注册表加载变量的值到界面上。存储的时候把界面上的值存储到注册表中。

注：注册表变量如下所示（截取了一小段）

```

1.  #define IDFF_filterBitmap 1650
2.  #define IDFF_isBitmap 1651
3.  #define IDFF_showBitmap 1652
4.  #define IDFF_orderBitmap 1653
5.  #define IDFF_fullBitmap 1654
6.  #define IDFF_bitmapFlnm 1655
7.  #define IDFF_bitmapPosx 1656
8.  #define IDFF_bitmapPosy 1657
9.  #define IDFF_bitmapPosmode 1658
10. #define IDFF_bitmapAlign 1659
11. #define IDFF_bitmapMode 1660
12. #define IDFF_bitmapStrength 1661

```

此外需要注意的是，ffdshow尽管包含了图形化的属性界面，却没有使用MFC类库，因而MFC的很多函数都不能使用，对此我还不甚了解为什么要这样，以后有机会要探究探究。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/12981725>

文章标签：[ffdshow](#) [滤镜](#) [位图叠加](#) [directshow](#) [源代码](#)

个人分类：[ffdshow](#)

所属专栏：[开源多媒体项目源代码分析](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com