Video filtering with FFmpeg

Coding Technology Document

- FFMPEG를 활용한 Visual Studio 내 Filtering–

2017.01.24

작성자 : 김문원, 황진하

목차

[1. Summary 2](#_Toc473100050)

[2. System Environment 2](#_Toc473100051)

[3. Environmet Setting 2](#_Toc473100052)

[4. Coding Guide 3](#_Toc473100053)

[4.1 Header Include 3](#_Toc473100054)

[4.2 Initialize Structure 3](#_Toc473100055)

[4.3 Regist Filter 4](#_Toc473100056)

[4.4 Initialize Filter 4](#_Toc473100057)

[4.5 Insert Filter 6](#_Toc473100058)

[4.6 Set Video Filter 7](#_Toc473100059)

[4.7 Apply Function 8](#_Toc473100060)

[5. Result 9](#_Toc473100061)

[5.1 Video Filtering Result 10](#_Toc473100062)

[6. Reference 10](#_Toc473100063)

[7. Supplement 11](#_Toc473100064)

[7.1 No option Filter 11](#_Toc473100065)

[7.2 Option Filter 12](#_Toc473100066)

# Summary

본 문서는 Video의 Filter에 대한 환경 구축. Filter를 적용하는 방법에 대해 서술한다.

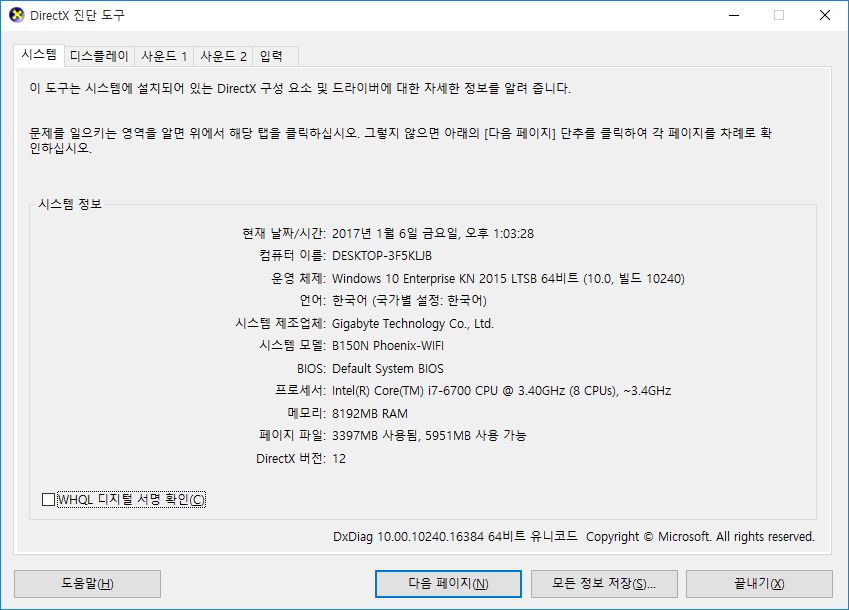
Filter를 적용하는 방법은 FFmepg filter library를 활용한 filter 적용과,

Frame Data를 편집해, 원하는 Filter를 적용하는 방법으로 기술 되어 있다.

# System Environment

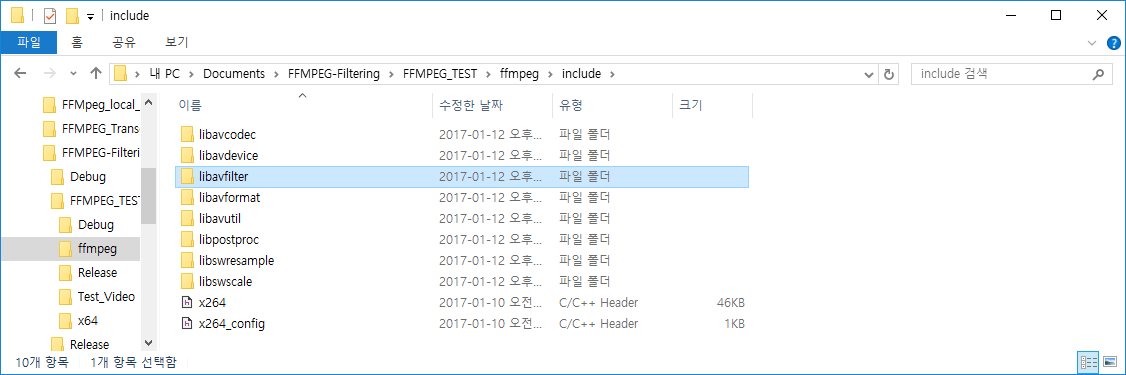
Microsoft Visual Studio 2015 (v140)

FFMPEG 3.1.2



# Environmet Setting

* FFmpeg Library[[1]](#footnote-1)를 참고해, 필터 적용에 필요한 정보를 추가한다.
* Filter와 관련된 Library는 “include\libavfilter” 폴더 내 정리가 되어있다.
* Visual Studio내 에서 사용하기 위해, 해당 Library를 추가한다.



<해당 폴더가 AVfilter의 헤더파일을 포함한다.>

# Coding Guide[[2]](#footnote-2)

## Header Include

* + - FFMPEG를 사용하기 위해, 필요한 헤더 파일들을 포함한다.

#include <libavfilter\avfilter.h> // Main libfilter

#include <libavfilter\buffersink.h> // Sink Filter

#include <libavfilter\buffersrc.h> // Source Filter

## Initialize Structure

* + - 이후 Filter를 관리하기 위해 FilterContext 구조체를 정의한다.

typedef struct \_FilterContext {

AVFilterGraph \*filter\_graph; // Filter's Context

AVFilterContext \*src\_ctx; // Linked Buffer Source

AVFilterContext \*sink\_ctx; // Linked Buffer Sink

AVFilterInOut \*inputs; // Filter Input

AVFilterInOut \*outputs; // Filter Output

int last\_filter\_idx = 0; // Last filter index for Linking

} FilterContext;

## Regist Filter

* + - Filter에 관련 된 모든 정보를 Regist하기 위해 다음 함수를 사용한다.

avfilter\_register\_all();

* + - 위 함수는 Avfilter가 사용 할수 있는 모든 Filter를 등록한다.

## Initialize Filter

* + - Filter를 설정하기 위해 기본적인 Link, in&out put, buffer를 생성, 할당한다.
    - 필터의 초기화는 다음의 순서를 따른다.
    - 이 과정은 static int init\_video\_filter() 함수 내에서 구현 한다.
    - Filter의 context를 저장하는 Filter Graph를 Momery에 할당한다.

// Allocate memory for filter graph

vfilter\_ctx.filter\_graph = avfilter\_graph\_alloc();

if (vfilter\_ctx.filter\_graph == NULL)

{

return -1;

}

* + - Filter Graph에 AVFilterInOut의 input 과 output을 Linking 한다.

// Link input and output with filter graph.

if (avfilter\_graph\_parse2(vfilter\_ctx.filter\_graph, "null", &inputs, &outputs) < 0)

{

printf("Failed to parse video filtergraph\n");

return -2;

}

* + - 모든 준비가 완료되면 필터를 생성한다.
    - 원본 프레임이 들어가는 Buffer Source Filter를 생성하기에 앞서, Buffer Source Filter에 해당 Video 의 정보를 버퍼에 입력한다.

char args[512];

// Video (Time\_base, Size, Pixel\_format, Pixel\_aspect)

snprintf(args, sizeof(args), "time\_base=%d/%d:video\_size=%dx%d:pix\_fmt=%d:pixel\_aspect=%d/%d"

, stream->time\_base.num, stream->time\_base.den

, codec\_ctx->width, codec\_ctx->height

, codec\_ctx->pix\_fmt

, codec\_ctx->sample\_aspect\_ratio.num, codec\_ctx->sample\_aspect\_ratio.den);

* + - * 위, 작업은 원본 Video 해당 Media의 Context가 입력 되어 있어야 한다.
    - 입력된 버퍼를 통해 Buffer Source filter를 생성한다.

// Create Buffer Source

if (avfilter\_graph\_create\_filter(

&vfilter\_ctx.src\_ctx

, avfilter\_get\_by\_name("buffer")

, "in", args, NULL, vfilter\_ctx.filter\_graph) < 0)

{

printf("Failed to create video buffer source\n");

return -3;

}

* + - 생성한 Buffer source 와 Input Filter를 Link 한다.

// Link Buffer Source with input filter

if (avfilter\_link(vfilter\_ctx.src\_ctx, 0, inputs->filter\_ctx, 0) < 0)

{

printf("Failed to link video buffer source\n");

return -4;

}

* + - 이어서 Output이 연결 될 Buffer Sink를 생성한다.

// Create Buffer Sink

if (avfilter\_graph\_create\_filter(

&vfilter\_ctx.sink\_ctx

, avfilter\_get\_by\_name("buffersink")

, "out", NULL, NULL, vfilter\_ctx.filter\_graph) < 0)

{

printf("Failed to create video buffer sink\n");

return -3;

}

* + - * Buffer Sink는 현재 Filter를 삽입하기위해, Output을 Link 하지 않는다.
      * 이후 Output과의 Link는 Filter를 마무리하는 set\_video\_filter 함수 내 에서 구현한다.

## Insert Filter

* + - Filter의 종류와, Command를 입력받아 해당하는 필터를 Buffer Source에 연결한다.

int insert\_filter(FilterContext\* F\_ctx, char\* filter\_name, const char\* command, int flag);

* + - FilterContext\* F\_ctx 🡪 Filter의 Context를 저장한다.
    - char\* filter\_name 🡪 생성 할 Filter의 Name을 저장한다.
    - const char\* command 🡪 Filter의 command 를 저장한다.
    - int flag 🡪 위 함수의 동작 여부를 저장한다.
    - Filter Context의 내의 last\_filter\_idx를 이용해, 마지막으로 연결 된 Filter Context를 가져온다.

int last\_filter\_idx = F\_ctx->last\_filter\_idx;

AVFilterContext\* last\_filter = F\_ctx->filter\_graph->filters[last\_filter\_idx];

* + - 예외 처리를 위해, last\_filter\_idx가 Filter\_default\_idx일 경우 output과 link 한다.

if (F\_ctx->last\_filter\_idx == Filter\_default\_idx) {

if (avfilter\_link(F\_ctx->outputs->filter\_ctx, 0, filter\_ctx, 0) < 0)

{

rintf("Failed to link video format filter\n");

return -4;

}

return 0

}

* + - 예외 처리를 하면, 입력 된 filter\_name 과 command 를 파라미터로 Filter를 생성한다.

snprintf(args, sizeof(args), "%s", command); // write command to buffer

if (avfilter\_graph\_create\_filter(&filter\_ctx, avfilter\_get\_by\_name(filter\_name), filter\_name, args, NULL, F\_ctx->filter\_graph) < 0)

{

printf("Failed to create video scale filter\n");

return -4;

}

* + - filter를 생성 한 후에, last filter와 현재 생성 한 filter를 Link 한다.

if (avfilter\_link(last\_filter, 0, filter\_ctx, 0) < 0)

{

printf("Failed to link video format filter\n");

return -4;

}

## Set Video Filter

* + - 원하는 Filter를 생성 하고 Linking 한 후, Buffer Sink와 연결 하기 위해 다음의 함수를 이용한다.

static int set\_video\_filter(FilterContext\* F\_ctx)

* + - Last Filter와 Buffer Sink Filter를 Link 한다.

if (avfilter\_link(last\_filter, 0, F\_ctx->sink\_ctx, 0) < 0)

{

printf("Failed to link video format filter\n");

return -4;

}

* + - Buffer Sink의 Frame Size를 설정한다.

av\_buffersink\_set\_frame\_size(F\_ctx->sink\_ctx, codec\_ctx->frame\_size);

* + - Link가 모두 완료 되었으면, 모든 Filter를 Configure 한다.

if (avfilter\_graph\_config(F\_ctx->filter\_graph, NULL) < 0)

{

printf("Failed to configure video filter context\n");

return -5;

}

## Apply Function

* + - Main에서 실제로 구현 되는 Filter 함수는 Initailze, insert, set 의 단계를 따른다.
    - 기본적인 Filter의 설정은 다음의 예시를 통해 확인 할 수 있다.

// Video Filter Setting

init\_video\_filter(); // Initialize Filter

insert\_filter(&vfilter\_ctx, "negate", "", 0); // Apply Negative Filter

insert\_filter(&vfilter\_ctx, "rotate", "(t/30)/PI", 0); // Apply Rotate Filter

insert\_filter(&vfilter\_ctx, "null", "", 0); // Apply Null Filter

set\_video\_filter(&vfilter\_ctx); // Link Output Filter

※ 위 필터는 색상을 반전하고, 시계 방향으로 회전하고, Null Filter를 적용 한 Filter Setting의 예 이다.

* + - 이렇게 생성한 필터를 이용해 Decoding 된 Frame에 Filter effect 를 적용 할 수 있다.
    - 먼저 Video Stream에 대한 Filter를 할당한다.

FilterContext\* filter\_ctx;

if (read\_pkt.stream\_index == fmt\_ctx->nVSI)

filter\_ctx = &vfilter\_ctx;

* + - 이 후 av\_buffersrc\_add\_frame을 이용해 Buffer Source에 Frame 을 넣고, av\_buffersink\_get\_frame로 Buffer Sink에 저장 된 Filtered Frame을 가져오면 된다.

※ 이 때 Filter의 성격에 따라, 한 개의 Frame을 Add 했다고 해서, 반드시 한 개의 Frame이 Return 된다는 보장이 없으므로, 이에 유의 해야한다.

// Put frame into filter.

if (av\_buffersrc\_add\_frame(filter\_ctx->src\_ctx, pVFrame) < 0)

{

printf("Error occurred when putting frame into filter context\n");

break;

}

while (1)

{

// Get frame from filter, if it returns < 0 then filter is currently empty.

if (av\_buffersink\_get\_frame(filter\_ctx->sink\_ctx, filtered\_frame) < 0)

{

break;

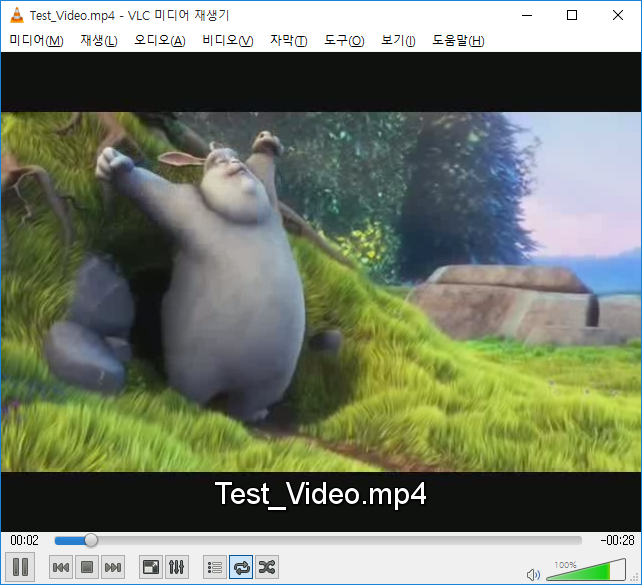
}

}

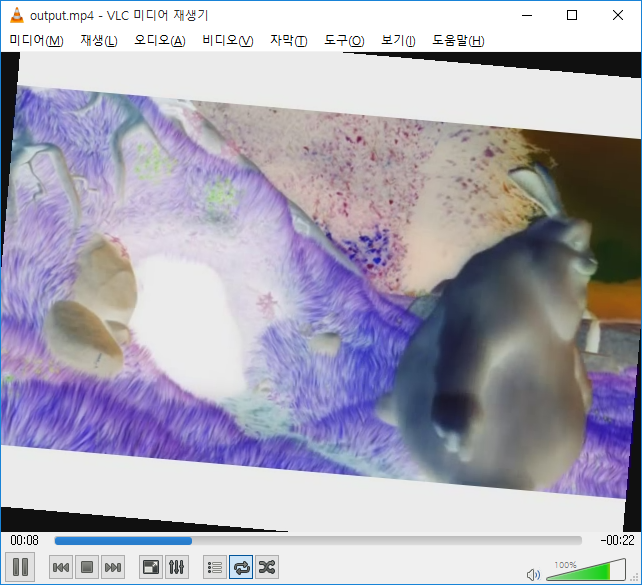
# Result

Filter Effect가 적용된 Media를 확인 할 수 있다.

## Video Filtering Result



<Original Video>



<Rotate, Negative Filtered Video>

# Reference

|  |  |
| --- | --- |
| 1 | Google, “FFMPEG Filtering” https://ffmpeg.org/ffmpeg-filters.html (2017.01.24) |
| FFmpeg에서 제공하는 Filter Library의 Content를 확인 할수 있다. |
| 2 | Google, “FFMPEG Lib” <https://www.ffmpeg.org/doxygen/2.1/index.html> (2017.01.24) |
| FFMPEG Library의 Document를 확인 할 수 있다. |
| 3 | Google, “FFMPEG Filter” https://trac.ffmpeg.org/wiki/FilteringGuide (2017.01.24) |
| FFmpeg의 Filtering Guide에 대해 나와있다. |
| 4 | Google, “FFmpeg filtering”  https://www.ffmpeg.org/doxygen/trunk/filtering\_8c-source.html (2017.01.24) |
| ‘C’로 구현된 Filter 적용 예시 코드를 확인 할 수 있다. |
| 5 | 이기곤, 『FFmpeg 라이브러리 코덱과 영상 변환을 중심으로』, 한빛미디어(2015) |
| 전반적인 FFMPEG Library와 Coding Guide 를 확인 할 수 있다. |

본 문서의 소스 코드는

Github에 등록 되어있다.

https://github.com/Phigaro/FFMPEG-Filtering

# Supplement

* FFmepg 내 적용 가능한 Filter.
* 본 부록에서는 Filter의 이름과, 예시코드를 첨부한다.
* 더욱 다양한 필터, 필터 옵션은 FFmpeg 공식 홈페이지의 Filter Table을 통해 확인 할 수 있다.[[3]](#footnote-3)

## No option Filter

|  |  |
| --- | --- |
| 필터 이름 | null |
| 필터 설명 | 아무것도 하지않음 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "null", "", 0); |

|  |  |
| --- | --- |
| 필터 이름 | hfilp |
| 필터 설명 | Video 상하 반전 필터 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "hflip", NULL, 0); |

|  |  |
| --- | --- |
| 필터 이름 | vfilp |
| 필터 설명 | Video 좌우 반전 필터 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "vflip", NULL, 0); |

|  |  |
| --- | --- |
| 필터 이름 | negate |
| 필터 설명 | Video 색상 반전 필터 (네거티브 효과) |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "negate", NULL, 0); |

|  |  |
| --- | --- |
| 필터 이름 | deband |
| 필터 설명 | Banding Artifact 제거 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "deband","", 0); |

|  |  |
| --- | --- |
| 필터 이름 | hqdn3d |
| 필터 설명 | 3D 영상의 Denoise filter |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "deband","", 0); |

|  |  |
| --- | --- |
| 필터 이름 | Inflate |
| 필터 설명 | Video에 Inflate 효과를 적용 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "inflate","", 0); |

|  |  |
| --- | --- |
| 필터 이름 | mcdeint |
| 필터 설명 | 모션 압축을 통한 디인터레이스 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "mcdeint","", 0); |

|  |  |
| --- | --- |
| 필터 이름 | showinfo |
| 필터 설명 | Input video frame의 정보를 표시함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "showinfo", "", 0); |

|  |  |
| --- | --- |
| 필터 이름 | swapuv |
| 필터 설명 | U & V 의 Plane을 Swap함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "swapuv", "", 0); |

## Option Filter

|  |  |
| --- | --- |
| 필터 이름 | rotate |
| 필터 설명 | Video를 회전함 |
| * 90[°](https://search.naver.com/search.naver?where=nexearch&query=%C2%B0&ie=utf8&sm=tab_she&qdt=0) 회전 | str = “90\* PI / 180”  insert\_filter(&vfilter\_ctx, "rotate", str, 0); |
| * 1[°](https://search.naver.com/search.naver?where=nexearch&query=%C2%B0&ie=utf8&sm=tab_she&qdt=0)/s 회전 | insert\_filter(&vfilter\_ctx, "rotate", "(t\*PI)/180", 0); |

|  |  |
| --- | --- |
| 필터 이름 | crop |
| 필터 설명 | Video를 자름 |
| * (12.34) cood 100x100 crop | str = “100:100:12:34”  insert\_filter(&vfilter\_ctx, "crop", str, 0); |

|  |  |
| --- | --- |
| 필터 이름 | avgblur |
| 필터 설명 | Avgblur effect 적용 |
| * 예시 코드 | insert\_filter(&vfilter\_ctx, "avgblur", "sizeX = 10", 0); |
| * sizeX | Horizontal kernel size |
| * sizeY | Vertical kernel size |

|  |  |
| --- | --- |
| 필터 이름 | boxblur |
| 필터 설명 | box blur effect 적용 |
| * 예시 코드 | insert\_filter(&vfilter\_ctx,"boxblur"," boxblur=2:1:cr=0:ar=0); |
| * lr | Luma radius |
| * lp | Luma power |
| * cr | Chroma radius |
| * cp | Chroma power |
| * ar | Alpha radius |
| * ap | Alpha power |

|  |  |
| --- | --- |
| 필터 이름 | convolution |
| 필터 설명 | 주변 화소를 이용해 Video를 편집함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "convolution","command", 0); |
| Blur | "1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1/9:1/9:1/9:1/9" |
| Edge enhance | "0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:5:1:1:1:0:128:128:128" |
| Edge detect | "0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:5:5:5:1:0:128:128:128" |
| Emboss | "-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2" |
| sharpen | "0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0" |

|  |  |
| --- | --- |
| 필터 이름 | delogo |
| 필터 설명 | Logo를 지움 (특정 부분을 blur 처리함) |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "delogo", "x=30:y=30:w=30:h=30:t=100", 0); |
| x, y | Blur 좌상단 좌표 |
| w, h | Blur 우하단 좌표 |
| t | fuzzy edge의 두께 |

|  |  |
| --- | --- |
| 필터 이름 | deshake |
| 필터 설명 | 카메라의 흔들림을 편집 할 때 사용. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "deshake", "x=50:y=100:w=300:h=300:rx=64:ry=64", 0); |
| rx, ry | Video 이동의 최대 범위 |

|  |  |
| --- | --- |
| 필터 이름 | drawbox |
| 필터 설명 | Video에 특정 box를 Draw |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "drawbox", "x=10:y=360:w=620:h=20:color=pink@0.3:t=max", 0); |
| color | Box의 color를 설정 [color=(color)@(transparency)] |
| t | Box의 외각선의 두께를 설정 (~max) |

|  |  |
| --- | --- |
| 필터 이름 | drawgrid |
| 필터 설명 | Video에 Grid effect를 적용 |
| 예시 코드 | insert\_filter(&vfilter\_ctx,"drawgrid","width=10:height=10:thickness=2:color=white@0.55",0); |

|  |  |
| --- | --- |
| 필터 이름 | eq |
| 필터 설명 | 밝기, 대비, 채도, 감마등을 수정함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "inflate","", 0); |
| contrast | 대비 (-2.0 ~ 2.0) |
| brightness | 밝기 (-1.0 ~ 1.0) |
| saturation | 채도 ( 0.0 ~ 3.0) |
| gamma | 감마 ( 0.1 ~ 10.0) |

|  |  |
| --- | --- |
| 필터 이름 | fade |
| 필터 설명 | Video에 fade (in & out) 효과를 적용 |
| fade in | insert\_filter(&vfilter\_ctx, "fade", "in:s=0:n=300", 0); |
| fade out | insert\_filter(&vfilter\_ctx, "fade", "out:s=60:n=30", 0); |

|  |  |
| --- | --- |
| 필터 이름 | fftfilt |
| 필터 설명 | Video에 fftfilt 필터를 적용 (Y U V 값을 이용) |
| 예시 코드 | insert\_filter(&vfilter\_ctx, " fftfilt ", "command", 0); |
| high-pass | dc\_Y=128:weight\_Y='squish(1-(Y+X)/100)' |
| low-pass | dc\_Y=0:weight\_Y='squish((Y+X)/100-1)' |
| sharpen | dc\_Y=0:weight\_Y='1+squish(1-(Y+X)/100)' |
| blur | dc\_Y=0:weight\_Y='exp(-4 \* ((Y+X)/(W+H)))' |

|  |  |
| --- | --- |
| 필터 이름 | format |
| 필터 설명 | Pixel Format을 변경 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "format", "pix\_fmts=yuvj420p", 0); |

|  |  |
| --- | --- |
| 필터 이름 | framestep |
| 필터 설명 | N 번째 Frame만을 선택함. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "framestep", "2", 0); 🡪 2번째 프레임만을 출력 |

|  |  |
| --- | --- |
| 필터 이름 | gblur |
| 필터 설명 | Gaussian blur 를 적용함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "gblur", "sigma=1:sigmaV=3:steps=6", 0); |
| sigma | horizontal sigma |
| sigmaV | Vertical sigma |
| steps | number of steps |

|  |  |
| --- | --- |
| 필터 이름 | gep |
| 필터 설명 | gep Filter를 적용함 (YUV, RGB color expression을 편집) |
| 예시 코드 | insert\_filter(&vfilter\_ctx,"geq"," p(W-X\,Y)", 0); 🡪 hflip |
| Sine wave | 128 + 100\*sin(2\*(PI/100)\*(cos(PI/3)\*(X-50\*T) + sin(PI/3)\*Y)):128:128 |

|  |  |
| --- | --- |
| 필터 이름 | gradfun |
| 필터 설명 | Bandinf artifact를 제거함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "gradfun", "3.5:8", 0); 🡪 3.5 strength, 8 radius |

|  |  |
| --- | --- |
| 필터 이름 | histogarm |
| 필터 설명 | color distribution histogarm을 보여준다. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "histogram", "levels\_mode=logarithmic", 0); |
|  | Default Output의 해상도를 256 x 636 으로 맞춘다. |

|  |  |
| --- | --- |
| 필터 이름 | hue |
| 필터 설명 | 색조, 채도 등을 수정한다. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "hue", "s=min(t/3\,1)", 0); |
| 채도 90[°](https://search.naver.com/search.naver?where=nexearch&query=%C2%B0&ie=utf8&sm=tab_she&qdt=0) 회전 | hue=H=PI/2:s=1 |
| Fade-in적용 | hue="s=min(t/3\,1)" |

|  |  |
| --- | --- |
| 필터 이름 | kerndeint |
| 필터 설명 | Deinterlace 한다. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "kerndeint", "thresh=0:map=1", 0); |
| Default | thresh=10:map=0:order=0:sharp=0:twoway=0 |
| White Pixel로 Mapping | thresh=0:map=1 |

|  |  |
| --- | --- |
| 필터 이름 | lenscorrection |
| 필터 설명 | Video에 렌즈 굴곡 효과를 적용함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "lenscorrection", "cx=0.5:cy=0.5:k1=0.3:k2=0.3", 0); |
| cx,cy | Image focal point의 x, y coordinate |
| k1 | 렌즈 굴곡식의 2차 보정 계수 |
| k2 | 렌즈 굴곡식의 이중 2차 보정 계수 |
| formula | r\_src = r\_tgt \* (1 + k1 \* (r\_tgt / r\_0)^2 + k2 \* (r\_tgt / r\_0)^4) |

|  |  |
| --- | --- |
| 필터 이름 | mpdecimate |
| 필터 설명 | frame rate를 줄이기 위해 특정 frame을 버림 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, " mpdecimate ","", 0); |
| max | Drop 될 수 있는 frame의 Maximum number |

|  |  |
| --- | --- |
| 필터 이름 | noformat |
| 필터 설명 | libavfilter가 특정한 pixel format을 사용하지 못하게함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "noformat", "yuv420p", 0); |
|  | “yuv420p|yuv444p|yuv410p” 🡪 해당 포맷을 사용하지 못하게함 |

|  |  |
| --- | --- |
| 필터 이름 | noise |
| 필터 설명 | Noise Effect를 추가함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "noise", "alls=20:allf=t+u", 0); |
| a | Averaged temporal noise |
| p | mix random noise |
| t | temporal noise |
| u | uniform noise |

|  |  |
| --- | --- |
| 필터 이름 | pad |
| 필터 설명 | padding을 추가함. |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "pad", "ih\*16/9:ih:(ow-iw)/2:(oh-ih)/2", 0); 🡪 **16:9 padding** |
|  | **640x480 Add paddings colored “violet”** |
|  | width=640:height=480:x=0:y=40:color=violet |

|  |  |
| --- | --- |
| 필터 이름 | perspective |
| 필터 설명 | Video에 perspective 효과를 적용 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "perspective", "0:0:W+300:0:0:H:W-300:H", 0); |
| Default | 0:0:W:0:0:H:W:H 🡪 (x0 y0 x1 y1 x2 y2 x3 y3) |
| View  Right\_bottom corner | 0:0:W+300:0:0:H:W-300:H |

|  |  |
| --- | --- |
| 필터 이름 | prewitt |
| 필터 설명 | prewitt operator를 적용함 🡪 edge detecting |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "prewitt", "", 0); |

|  |  |
| --- | --- |
| 필터 이름 | random |
| 필터 설명 | 캐시에 저장된 Frame을 Random Order로 Flush 함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "random", "frames=3", 0); |
| frame | internal cache의 Size ( # of frame ) |

|  |  |
| --- | --- |
| 필터 이름 | removegrain |
| 필터 설명 | Denoiser for Progressive |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "removegrain", "m0=20", 0); |
| m(0,1,2,3) | Plane을 설정함 🡪 mode의 Range는 (0~24) |

|  |  |
| --- | --- |
| 필터 이름 | setdar |
| 필터 설명 | DAR를 설정함 (Display Aspect Ratio) |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "setdar", "dar=16/9", 0); |

|  |  |
| --- | --- |
| 필터 이름 | setsar |
| 필터 설명 | SAR을 설정함 (Sample Aspect Ratio) |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "setsar", "sar=16/9", 0); |

|  |  |
| --- | --- |
| 필터 이름 | vignette |
| 필터 설명 | vignetting effect를 적용함 |
| 예시 코드 | insert\_filter(&vfilter\_ctx, "vignette", "", 0); |
| simple effect | vignette=PI/4 |
| Flickering | vignette='PI/4+random(1)\*PI/50':eval=frame |

1. Google, “FFMPEG Lib” <https://www.ffmpeg.org/doxygen/2.1/index.html> (2017.01.10) [↑](#footnote-ref-1)
2. Google, “FFmpeg filtering” https://www.ffmpeg.org/doxygen/trunk/filtering\_8c-source.html (2017.01.24) [↑](#footnote-ref-2)
3. Google, “FFMPEG Filtering” https://ffmpeg.org/ffmpeg-filters.html (2017.01.24) [↑](#footnote-ref-3)