KWANGWOON UNIVERSITY

High Dynamic Range

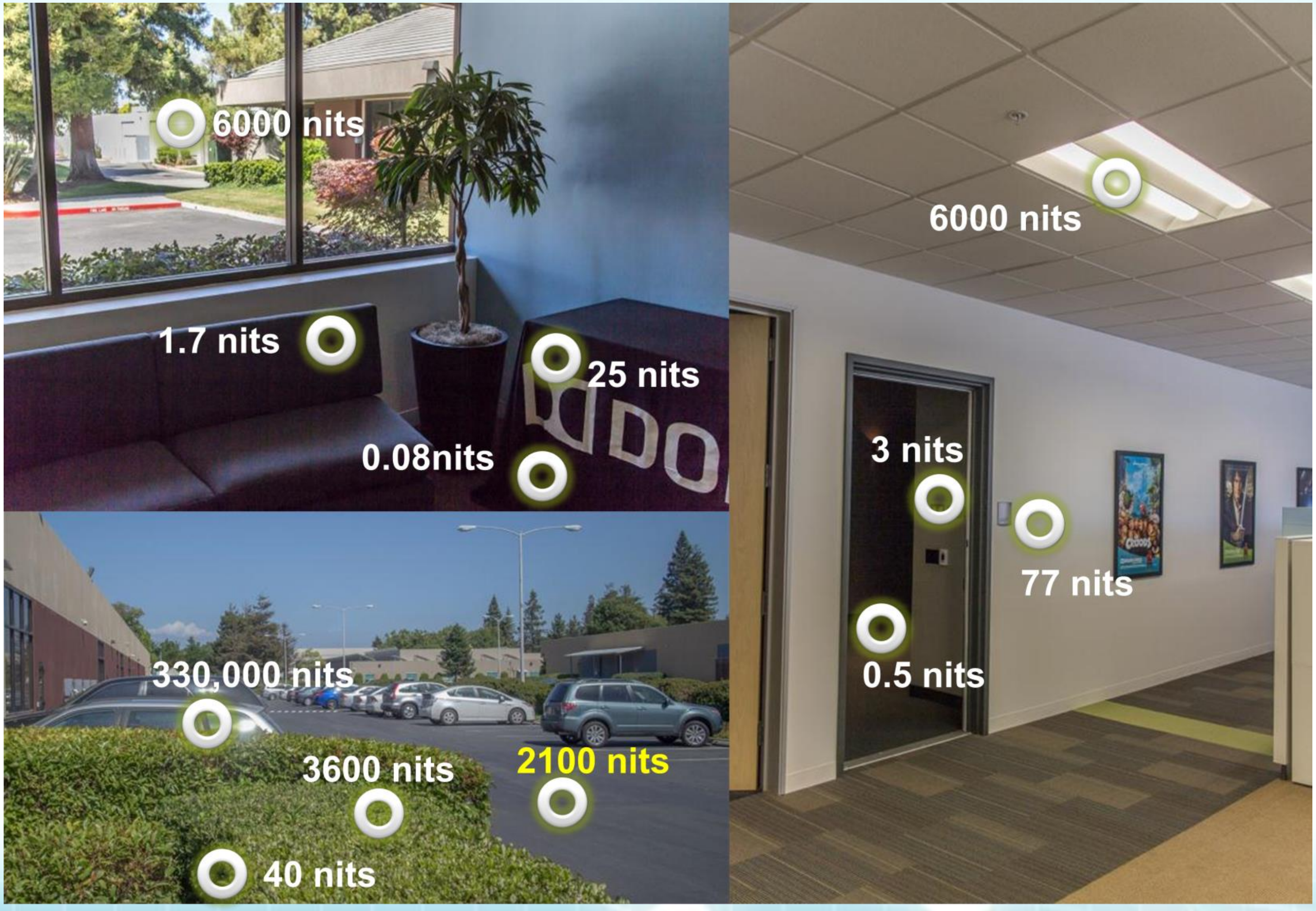
2018.11.27

Seoungjun Oh(sjoh@kw.ac.kr)
Wooju Lee (krosea@kw.ac.kr)

Multimedia LAB

VIA-Multimedia Center, Kwangwoon University

What is HDR?

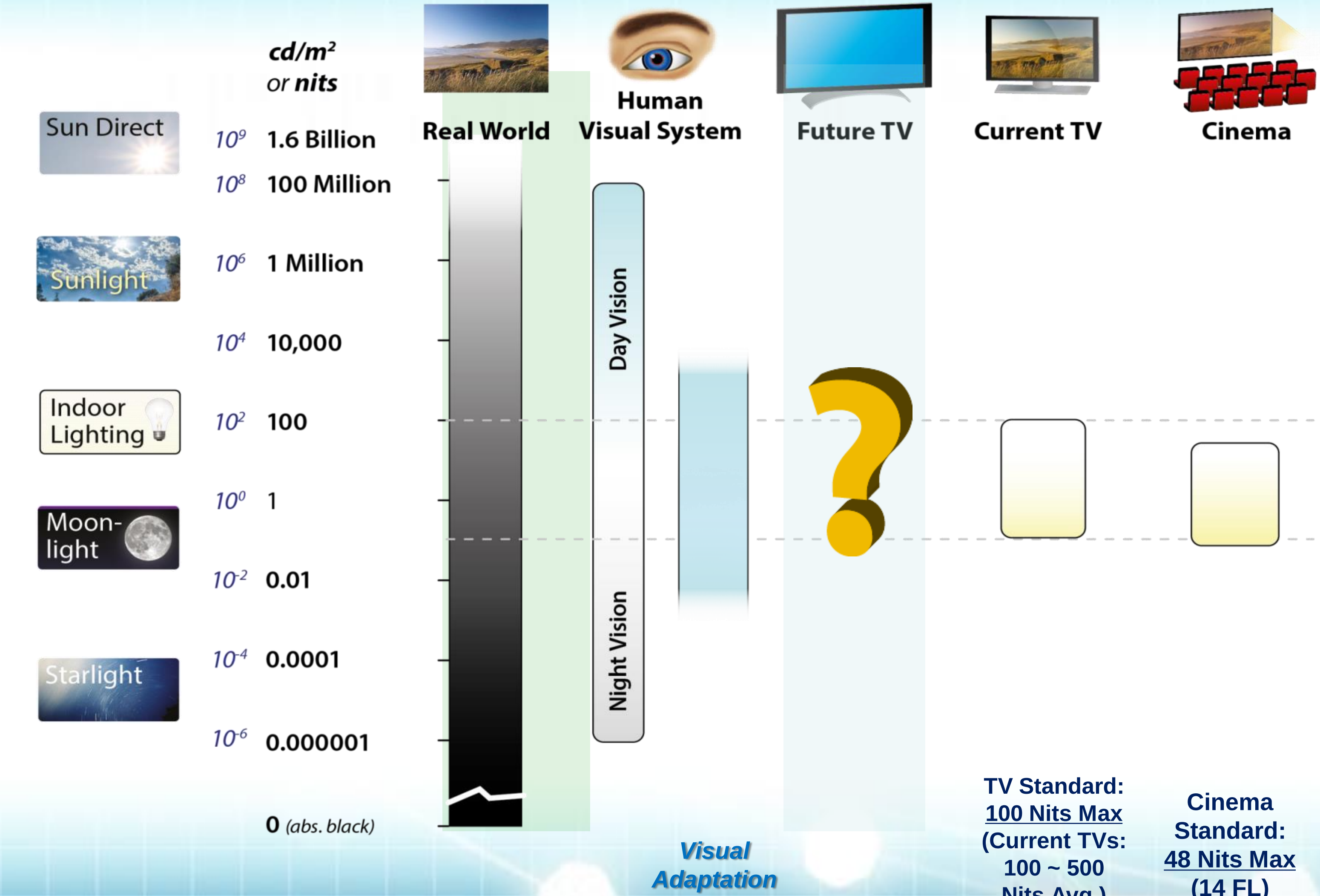


Source: ITU-R, “How much dynamic range for better pixel?,” Dolby, 2015

What is HDR?

Luminance Levels

Light units are in candela/m^2, mor e conveniently s poken - **“Nits”**

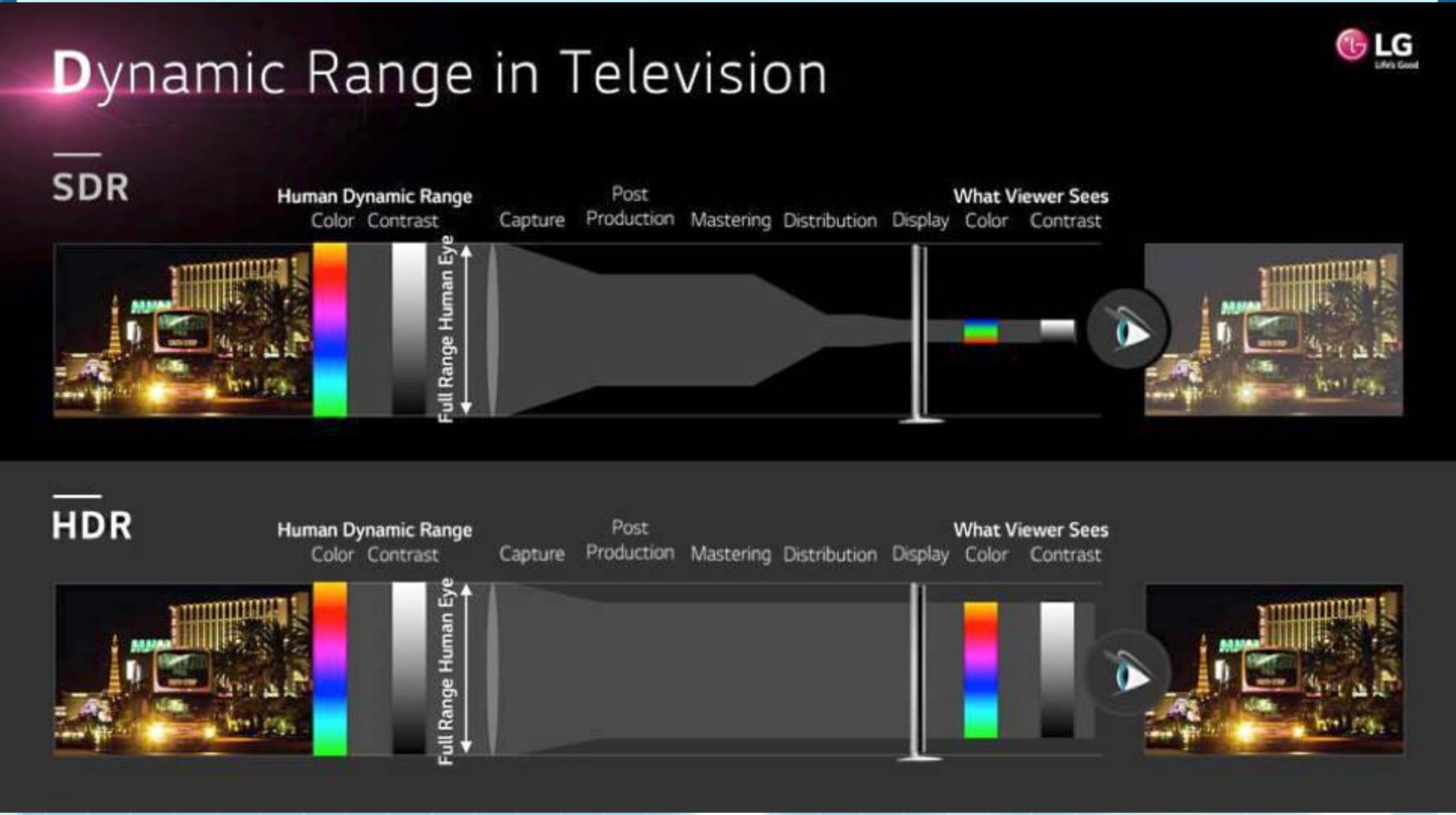


TV Standard: 100 Nits Max (Current TVs: 100 ~ 500 Nits Avg.)

Cinema Standard: 48 Nits Max (14 FL)

Source: SMPTE annual Conference 2013, Pat Griffis, Making Better Pixel

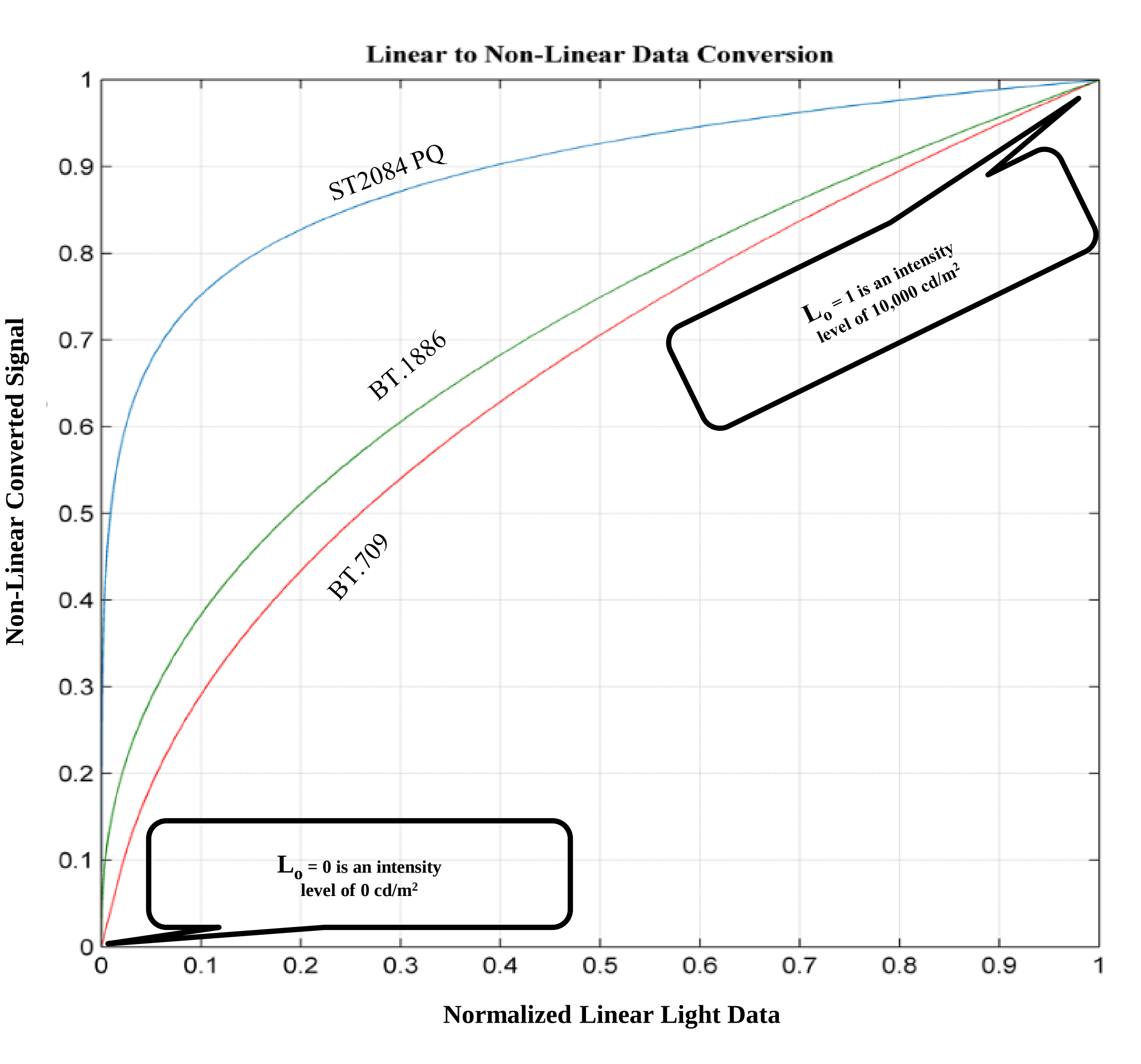
What is HDR?


Dynamic Range in Television
LG
Life's Good
SDR
Human Dynamic Range
Color
Contrast
Full Range Human Eye
Capture
Post Production
Mastering
Distribution
Display
What Viewer Sees
Color
Contrast
HDR
Human Dynamic Range
Color
Contrast
Full Range Human Eye
Capture
Post Production
Mastering
Distribution
Display
What Viewer Sees
Color
Contrast

What is HDR?

- HDR
  - Dynamic Range of a scene = max / min light intensity
  - Classification
    - SDR(Standard Dynamic Range) : < 1000:1
    - EDR (Enhanced Dynamic Range) : 1000:1 ~ 100,000:1
    - HDR (High Dynamic Range) : > 100,000:1
  - Standard for HDR display device
    - HDR10
    - Transfer function (Optical to Electrical)
      - PQ(Perceptual Quantization)
      - HLG(Hybrid Log-Gamma)

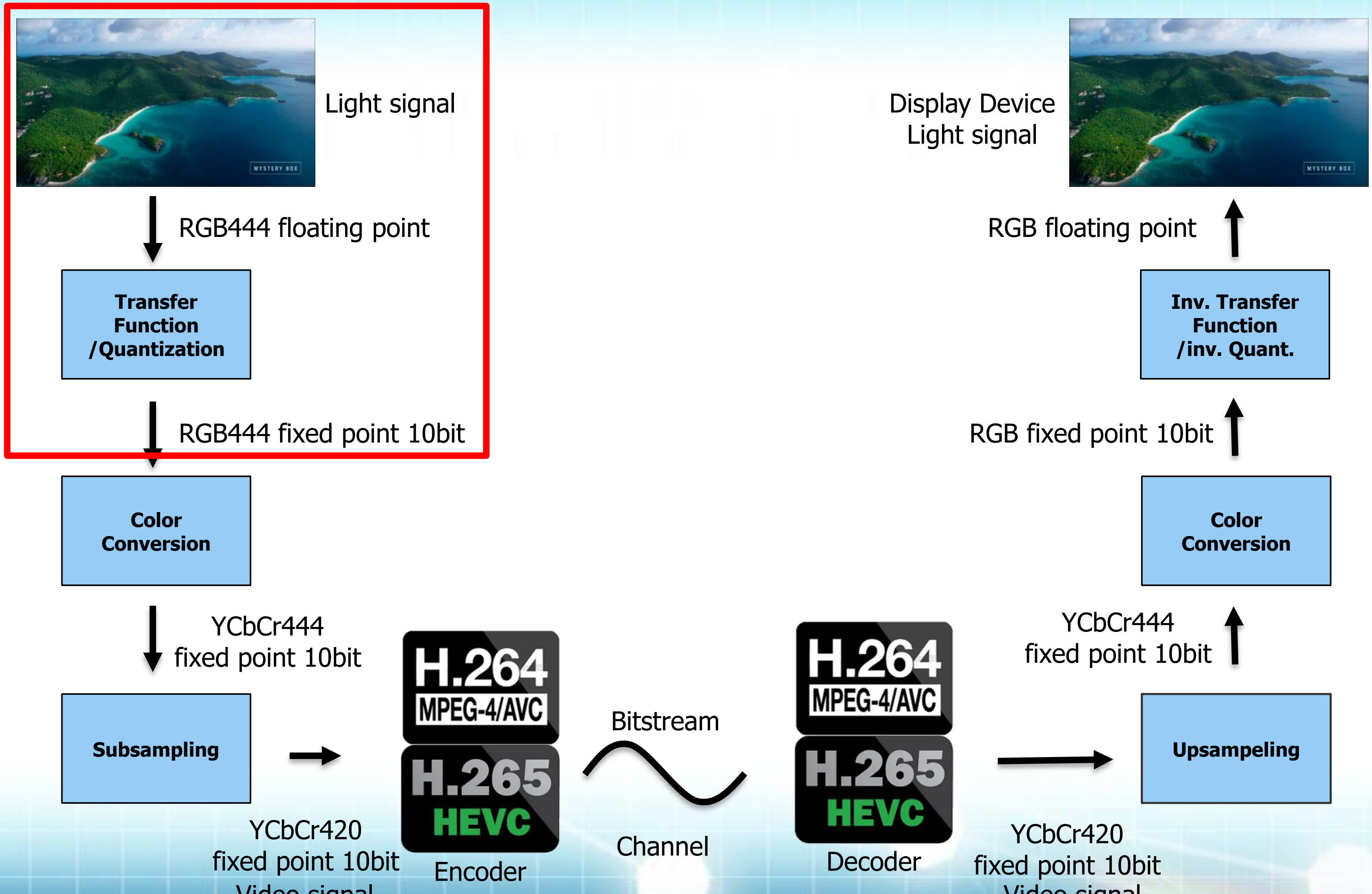
Perceptual Quantization



L_o = 1 is an intensity level of 10,000 cd/m^2

L_o = 0 is an intensity level of 0 cd/m^2

HDR in CODEC



Light signal

RGB444 floating point

Transfer Function /Quantization

RGB444 fixed point 10bit

Color Conversion

YCbCr444 fixed point 10bit

Subsampling

YCbCr420 fixed point 10bit Video signal

Encoder

Bitstream

Channel

Decoder

YCbCr420 fixed point 10bit Video signal

Upsampeling

YCbCr444 fixed point 10bit

Color Conversion

RGB fixed point 10bit

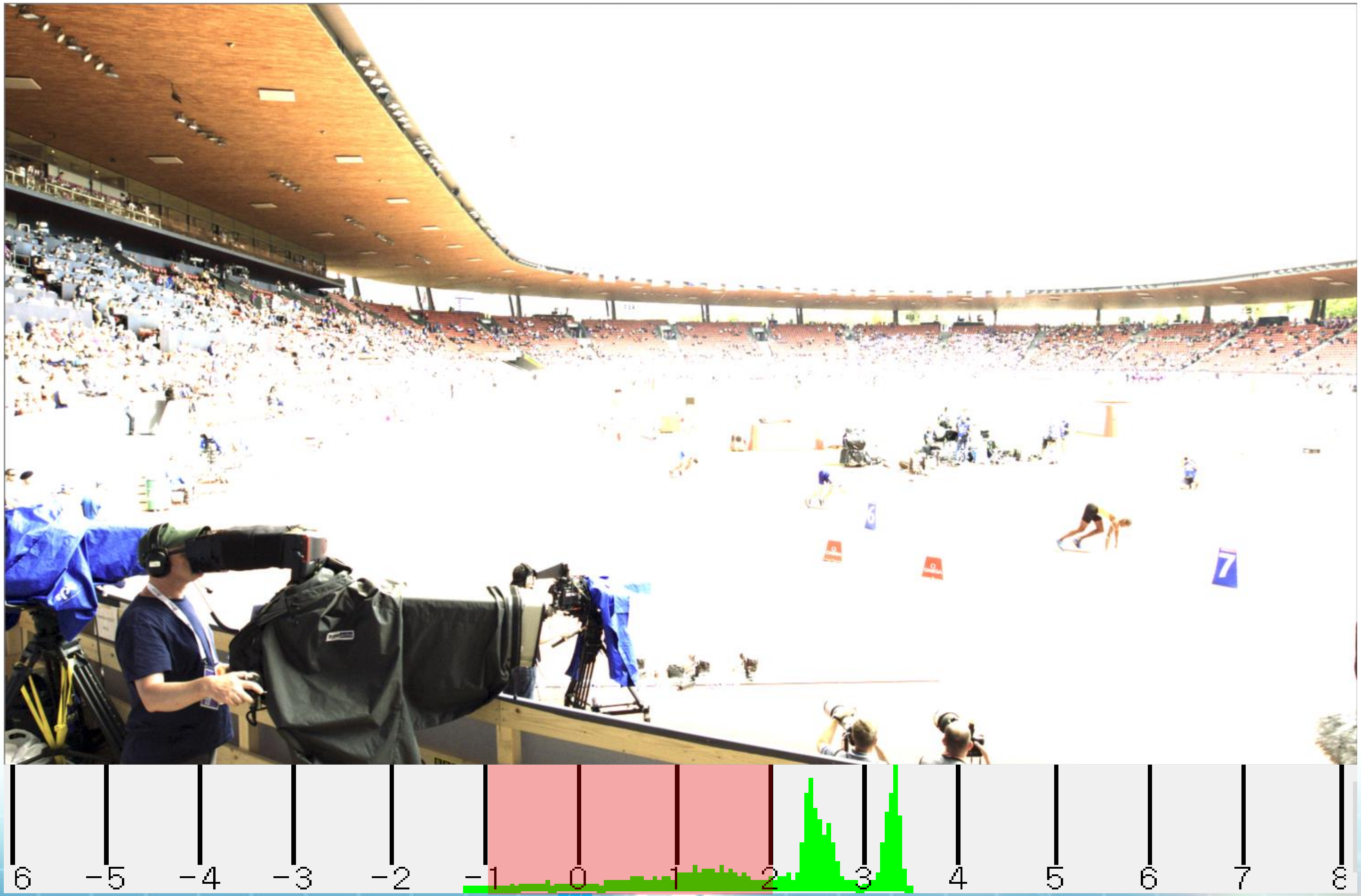
Inv. Transfer Function /inv. Quant.

RGB floating point

Display Device Light signal

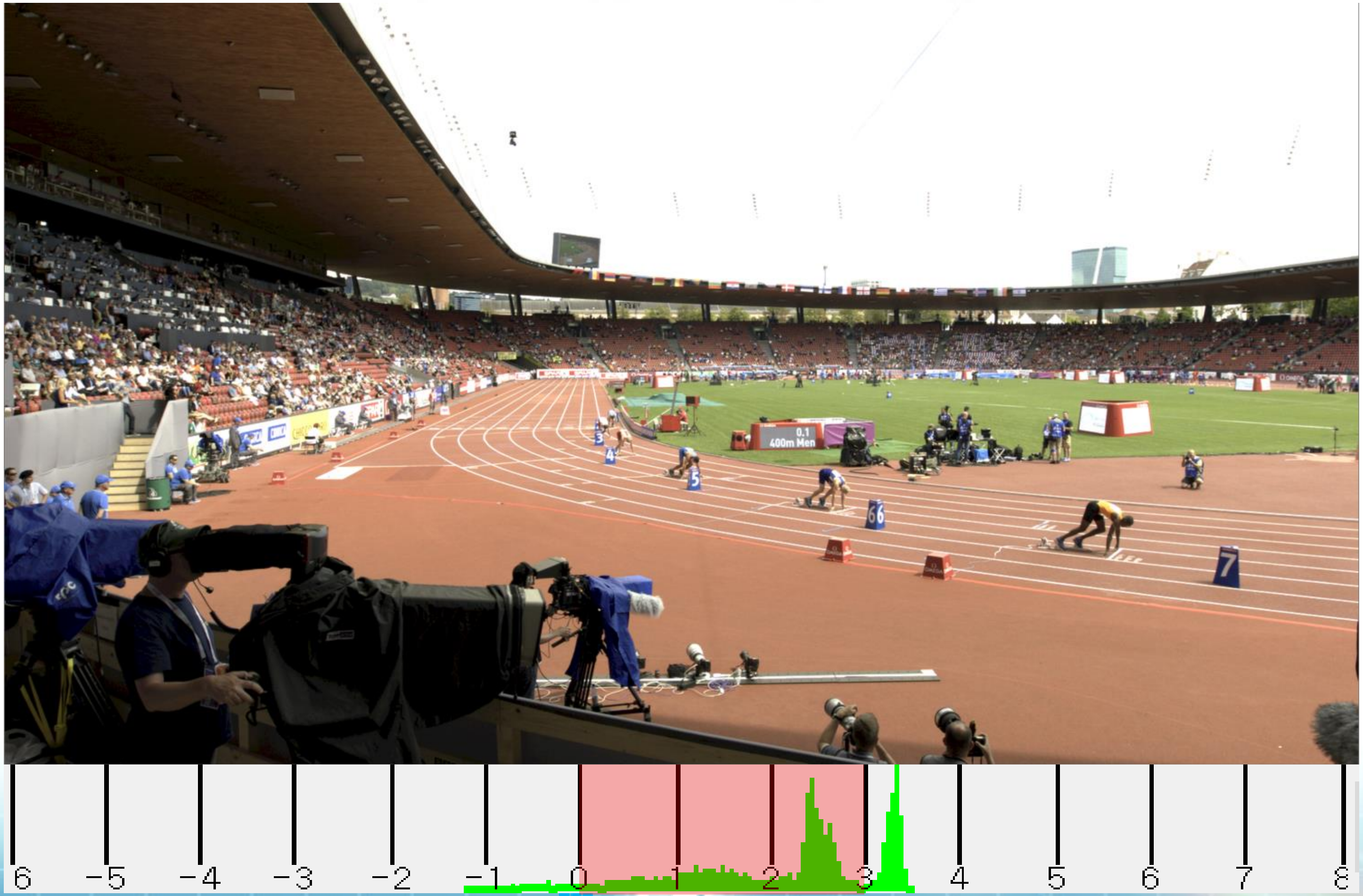
SDR vs HDR

- SDR



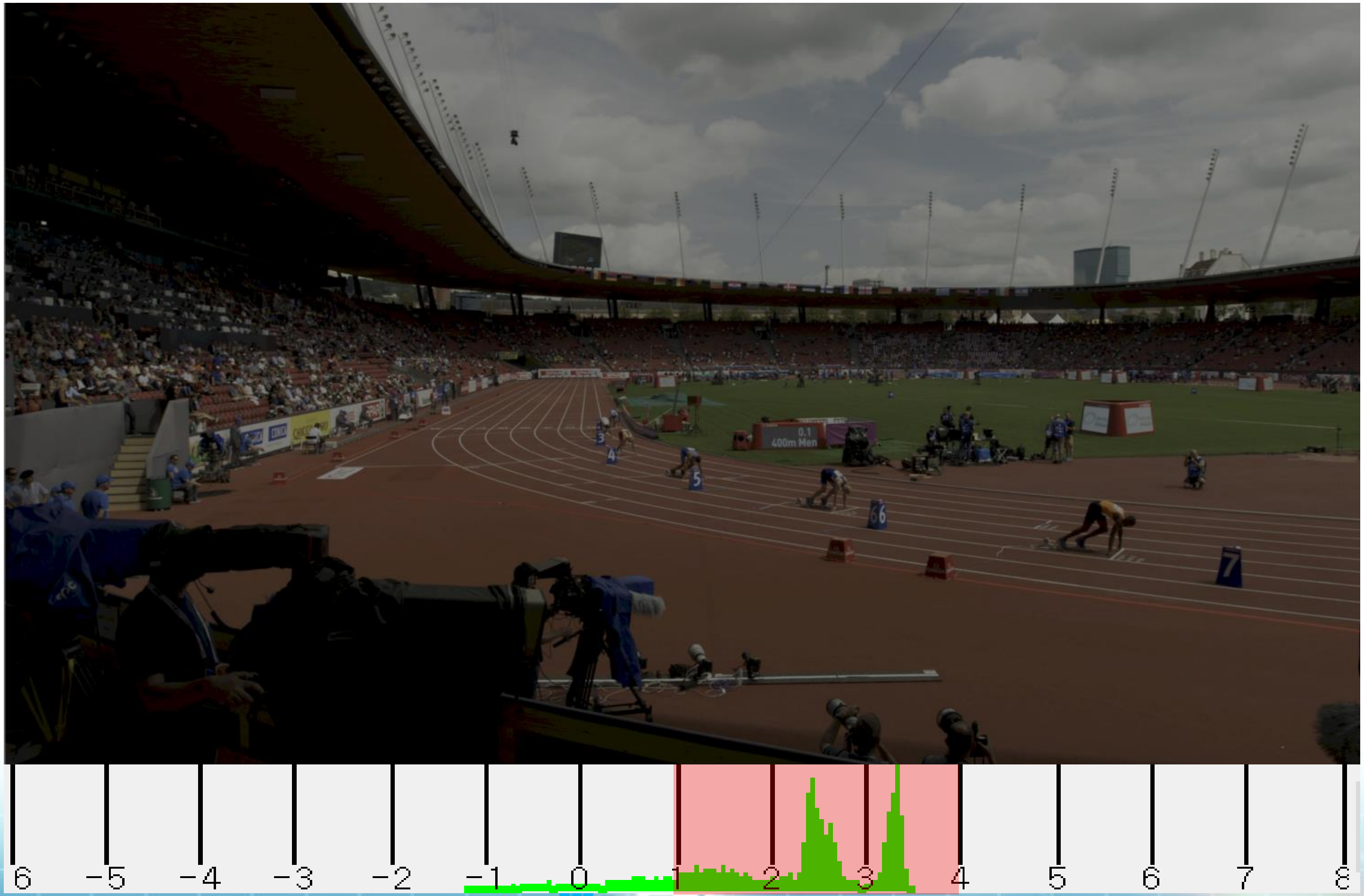
SDR vs HDR

- SDR



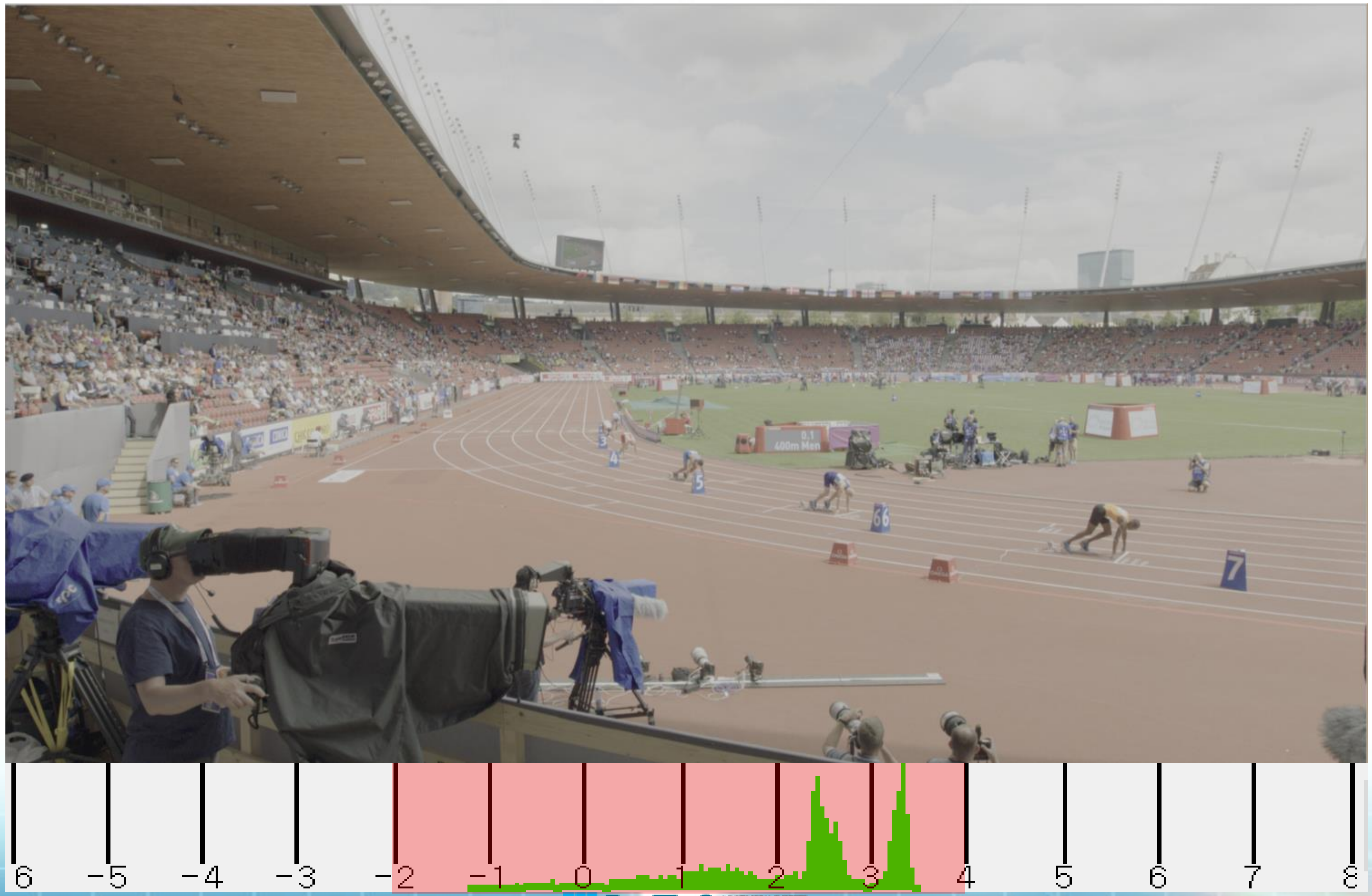
SDR vs HDR

- SDR



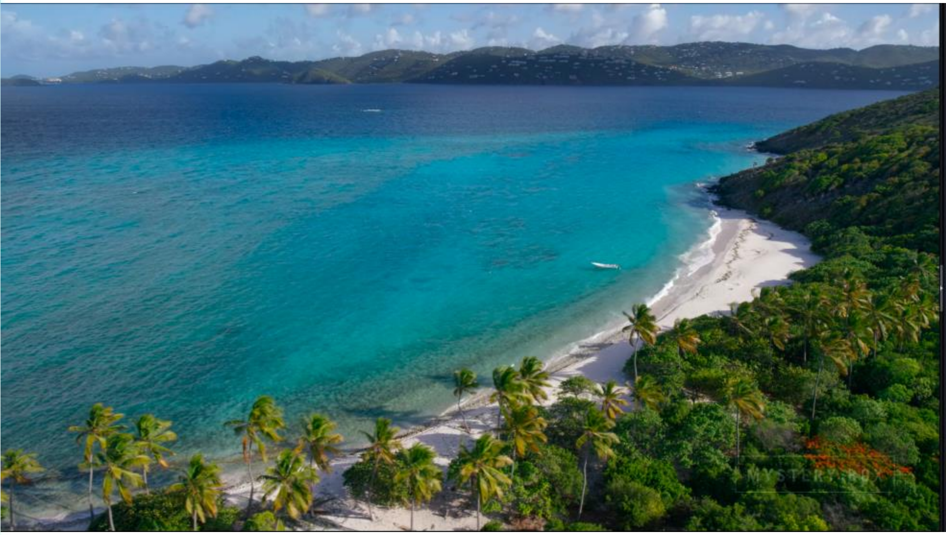
SDR vs HDR

- HDR

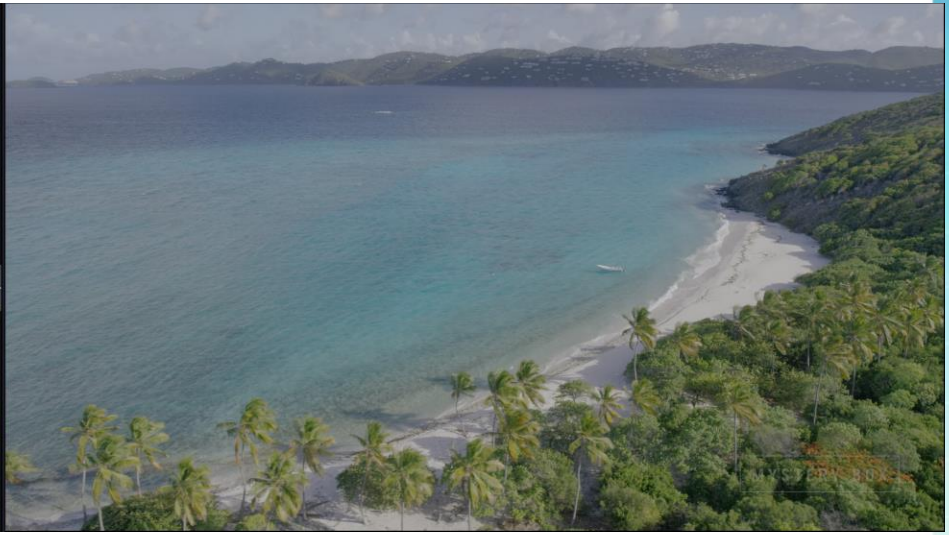


SDR vs HDR

- HDR



HDR rendering



SDR rendering

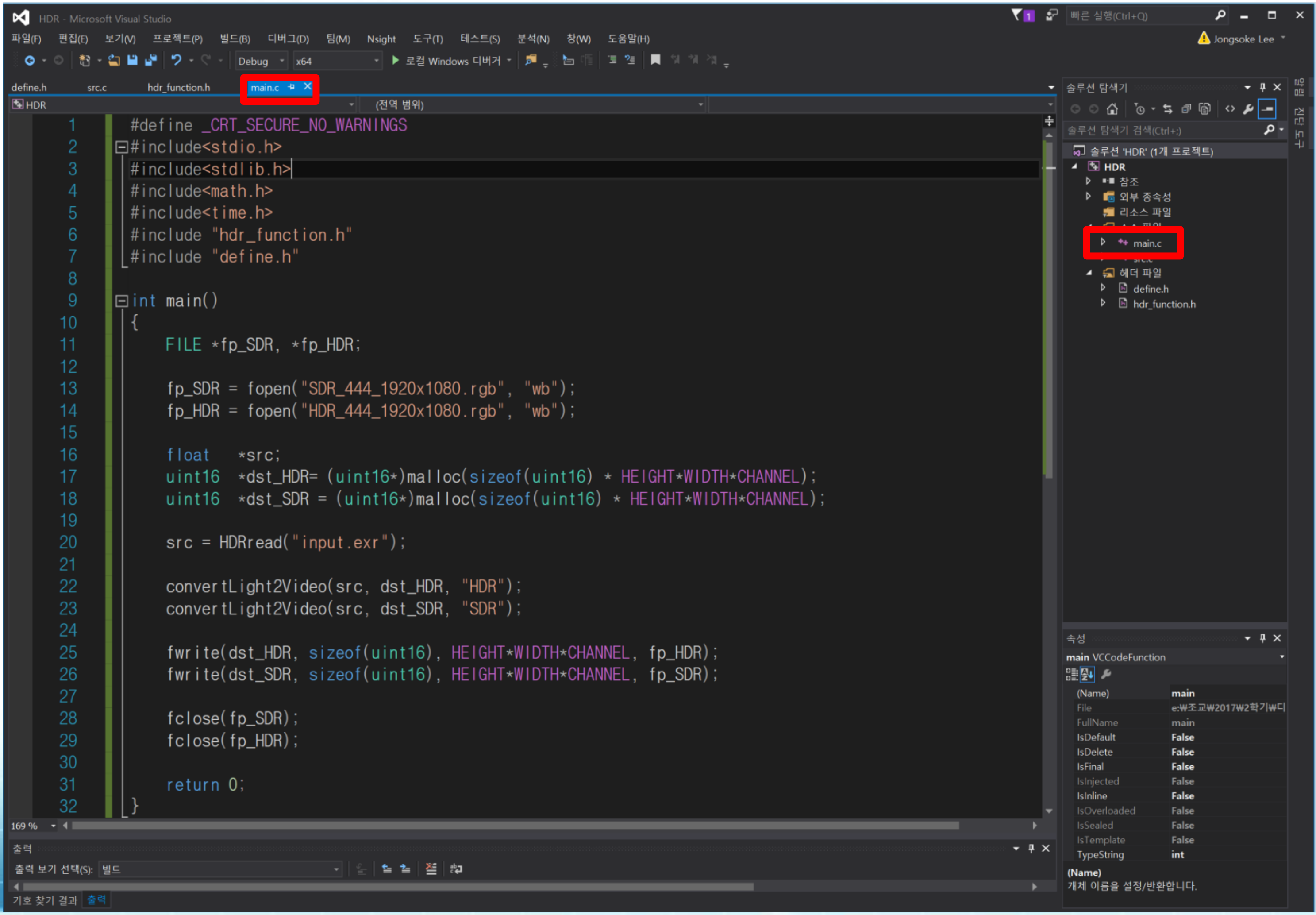
Assignment

- Assignment#1
  - Implement SDR(sRGB) Transfer function
    - V is linear light signal, [0, 1]
    - L is video signal, [0, 1]
    - $$L = \begin{cases} 12.92\ \times V & if\ V \leq 0.0031308 \\ \left((1 + 0.055) \times V\right)^{1/2.4} - 0.055 & if\ V > 0.0031308 \end{cases}$$

- Assignment#2
  - Decide SDR maximum intensity

Assignment

- Code



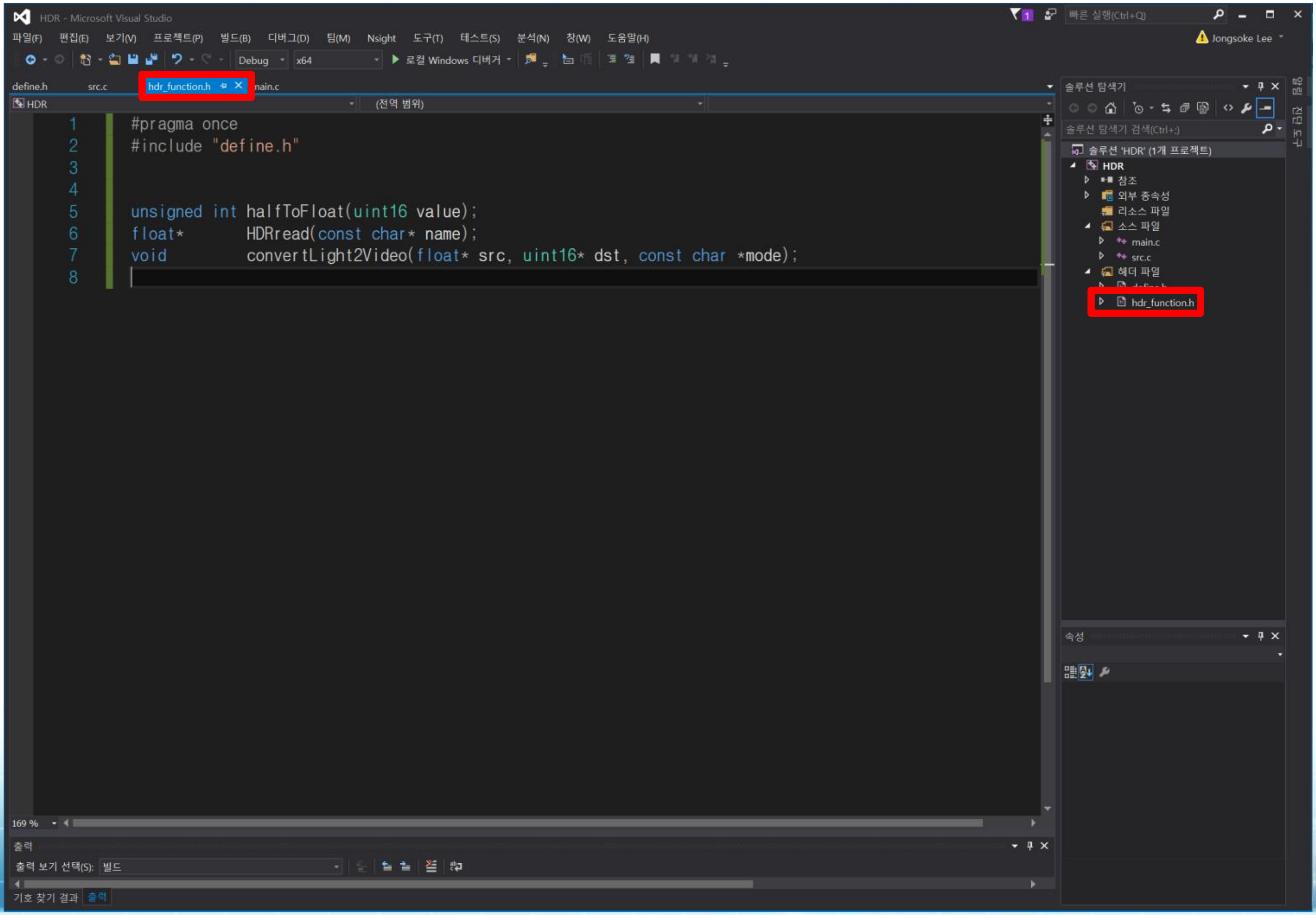
```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include "hdr_function.h"
#include "define.h"

int main()
{
    FILE *fp_SDR, *fp_HDR;

    fp_SDR = fopen("SDR_444_1920x1080.rgb", "wb");
    fp_HDR = fopen("HDR_444_1920x1080.rgb", "wb");

    float   *src;
    uint16  *dst_HDR= (uint16*)malloc(sizeof(uint16) * HEIGHT*WIDTH*CHANNEL);
    uint16  *dst_SDR = (uint16*)malloc(sizeof(uint16) * HEIGHT*WIDTH*CHANNEL);

    src = HDRread("input.exr");

    convertLight2Video(src, dst_HDR, "HDR");
    convertLight2Video(src, dst_SDR, "SDR");

    fwrite(dst_HDR, sizeof(uint16), HEIGHT*WIDTH*CHANNEL, fp_HDR);
    fwrite(dst_SDR, sizeof(uint16), HEIGHT*WIDTH*CHANNEL, fp_SDR);

    fclose(fp_SDR);
    fclose(fp_HDR);

    return 0;
}
```

(Name)	main
File	e:₩조교₩2017₩2학기₩디
FullName	main
IsDefault	False
IsDelete	False
IsFinal	False
IsInjected	False
IsInline	False
IsOverloaded	False
IsSealed	False
IsTemplate	False
TypeString	int

(Name)
개체 이름을 설정/반환합니다.

Assignment

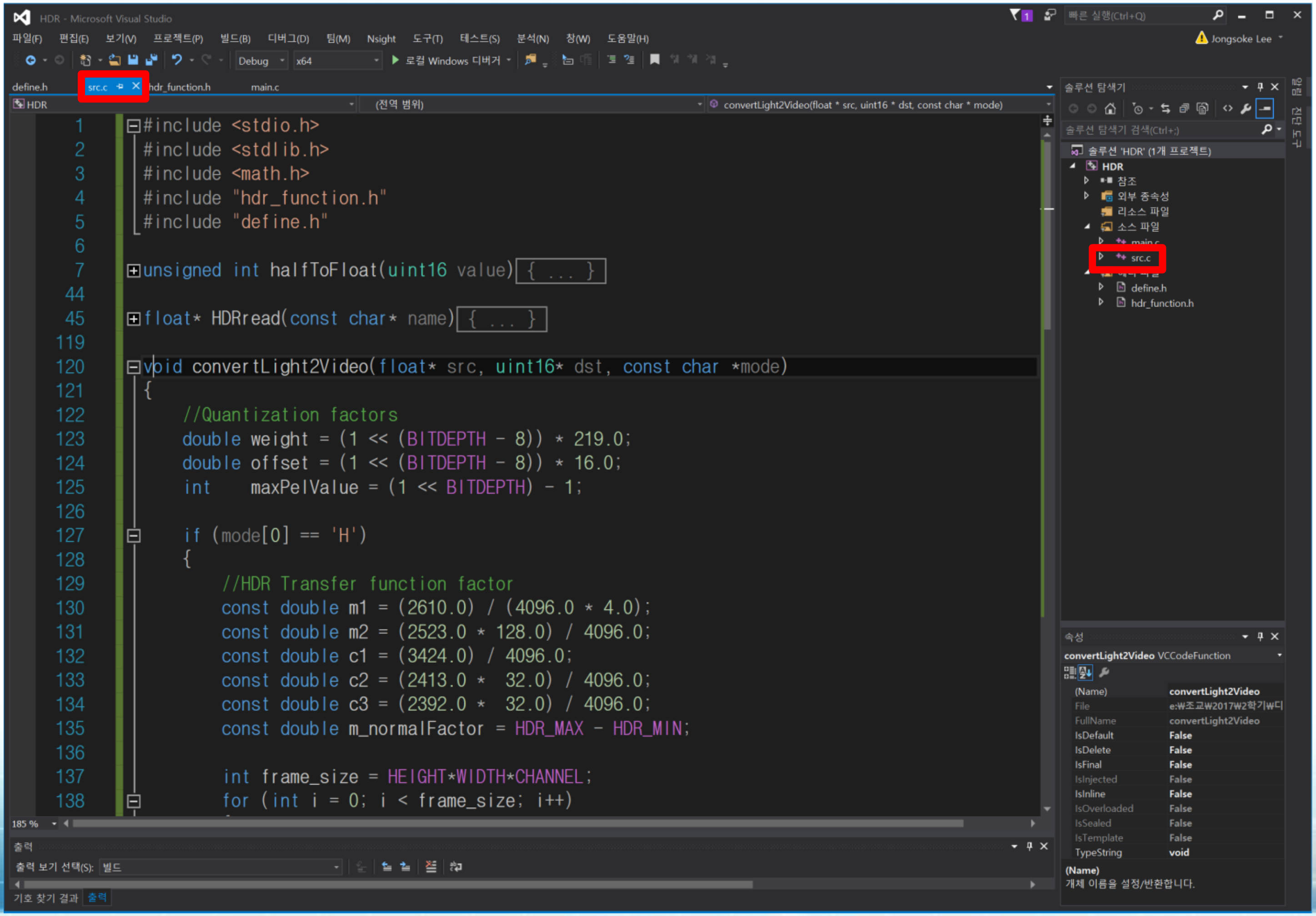
- Code



```c
#pragma once
#include "define.h"


unsigned int halfToFloat(uint16 value);
float*       HDRread(const char* name);
void         convertLight2Video(float* src, uint16* dst, const char *mode);

```

Assignment

❖ Code


HDR - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) Nsight 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
Debug
x64
▶ 로컬 Windows 디버거
빠른 실행(Ctrl+Q)
Jongsoke Lee
define.h
src.c
hdr_function.h
main.c
HDR
(전역 범위)
convertLight2Video(float * src, uint16 * dst, const char * mode)
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'HDR' (1개 프로젝트)
HDR
참조
외부 종속성
리소스 파일
소스 파일
main.c
src.c
헤더 파일
define.h
hdr_function.h
속성
185 %
출력
출력 보기 선택(S): 빌드
기호 찾기 결과
출력

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "hdr_function.h"
#include "define.h"

unsigned int halfToFloat(uint16 value){ ... }

float* HDRread(const char* name){ ... }

void convertLight2Video(float* src, uint16* dst, const char *mode)
{
    //Quantization factors
    double weight = (1 << (BITDEPTH - 8)) * 219.0;
    double offset = (1 << (BITDEPTH - 8)) * 16.0;
    int    maxPelValue = (1 << BITDEPTH) - 1;

    if (mode[0] == 'H')
    {
        //HDR Transfer function factor
        const double m1 = (2610.0) / (4096.0 * 4.0);
        const double m2 = (2523.0 * 128.0) / 4096.0;
        const double c1 = (3424.0) / 4096.0;
        const double c2 = (2413.0 *  32.0) / 4096.0;
        const double c3 = (2392.0 *  32.0) / 4096.0;
        const double m_normalFactor = HDR_MAX - HDR_MIN;

        int frame_size = HEIGHT*WIDTH*CHANNEL;
        for (int i = 0; i < frame_size; i++)
```

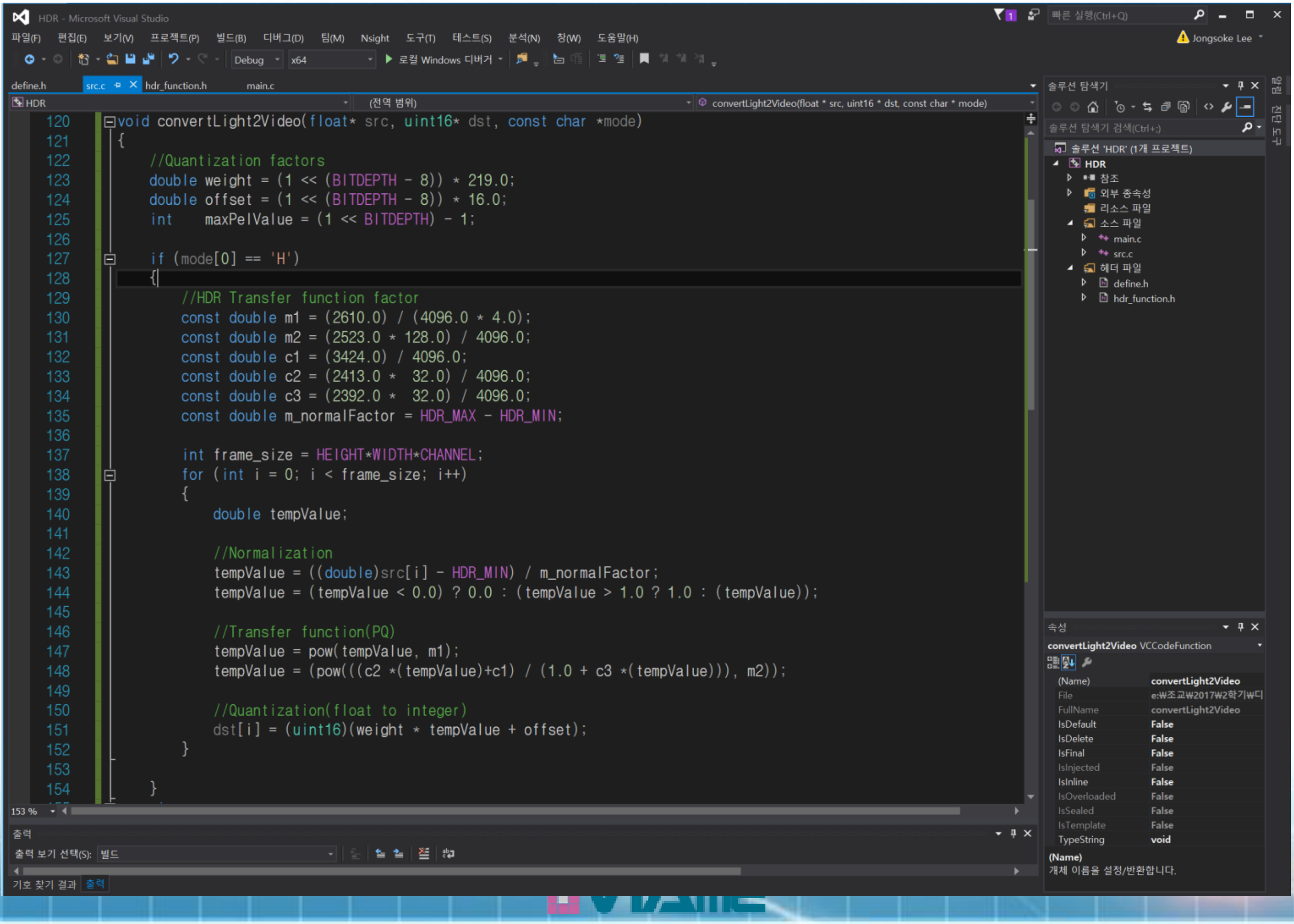
convertLight2Video VCCodeFunction

(Name)	convertLight2Video
File	e:₩조교₩2017₩2학기₩디
FullName	convertLight2Video
IsDefault	False
IsDelete	False
IsFinal	False
IsInjected	False
IsInline	False
IsOverloaded	False
IsSealed	False
IsTemplate	False
TypeString	void

(Name)
개체 이름을 설정/반환합니다.

Assignment

- Code


HDR - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) Nsight 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
빠른 실행(Ctrl+Q)
Jongsoke Lee
Debug
x64
로컬 Windows 디버거
define.h
src.c
hdr_function.h
main.c
HDR
(전역 범위)
convertLight2Video(float * src, uint16 * dst, const char * mode)
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'HDR' (1개 프로젝트)
HDR
참조
외부 종속성
리소스 파일
소스 파일
main.c
src.c
헤더 파일
define.h
hdr_function.h
153 %
출력
출력 보기 선택(S): 빌드
기호 찾기 결과
출력

```
void convertLight2Video(float* src, uint16* dst, const char *mode)
{
    //Quantization factors
    double weight = (1 << (BITDEPTH - 8)) * 219.0;
    double offset = (1 << (BITDEPTH - 8)) * 16.0;
    int    maxPelValue = (1 << BITDEPTH) - 1;

    if (mode[0] == 'H')
    {
        //HDR Transfer function factor
        const double m1 = (2610.0) / (4096.0 * 4.0);
        const double m2 = (2523.0 * 128.0) / 4096.0;
        const double c1 = (3424.0) / 4096.0;
        const double c2 = (2413.0 *  32.0) / 4096.0;
        const double c3 = (2392.0 *  32.0) / 4096.0;
        const double m_normalFactor = HDR_MAX - HDR_MIN;

        int frame_size = HEIGHT*WIDTH*CHANNEL;
        for (int i = 0; i < frame_size; i++)
        {
            double tempValue;

            //Normalization
            tempValue = ((double)src[i] - HDR_MIN) / m_normalFactor;
            tempValue = (tempValue < 0.0) ? 0.0 : (tempValue > 1.0 ? 1.0 : (tempValue));

            //Transfer function(PQ)
            tempValue = pow(tempValue, m1);
            tempValue = (pow(((c2 *(tempValue)+c1) / (1.0 + c3 *(tempValue))), m2));

            //Quantization(float to integer)
            dst[i] = (uint16)(weight * tempValue + offset);
        }

    }
```

속성

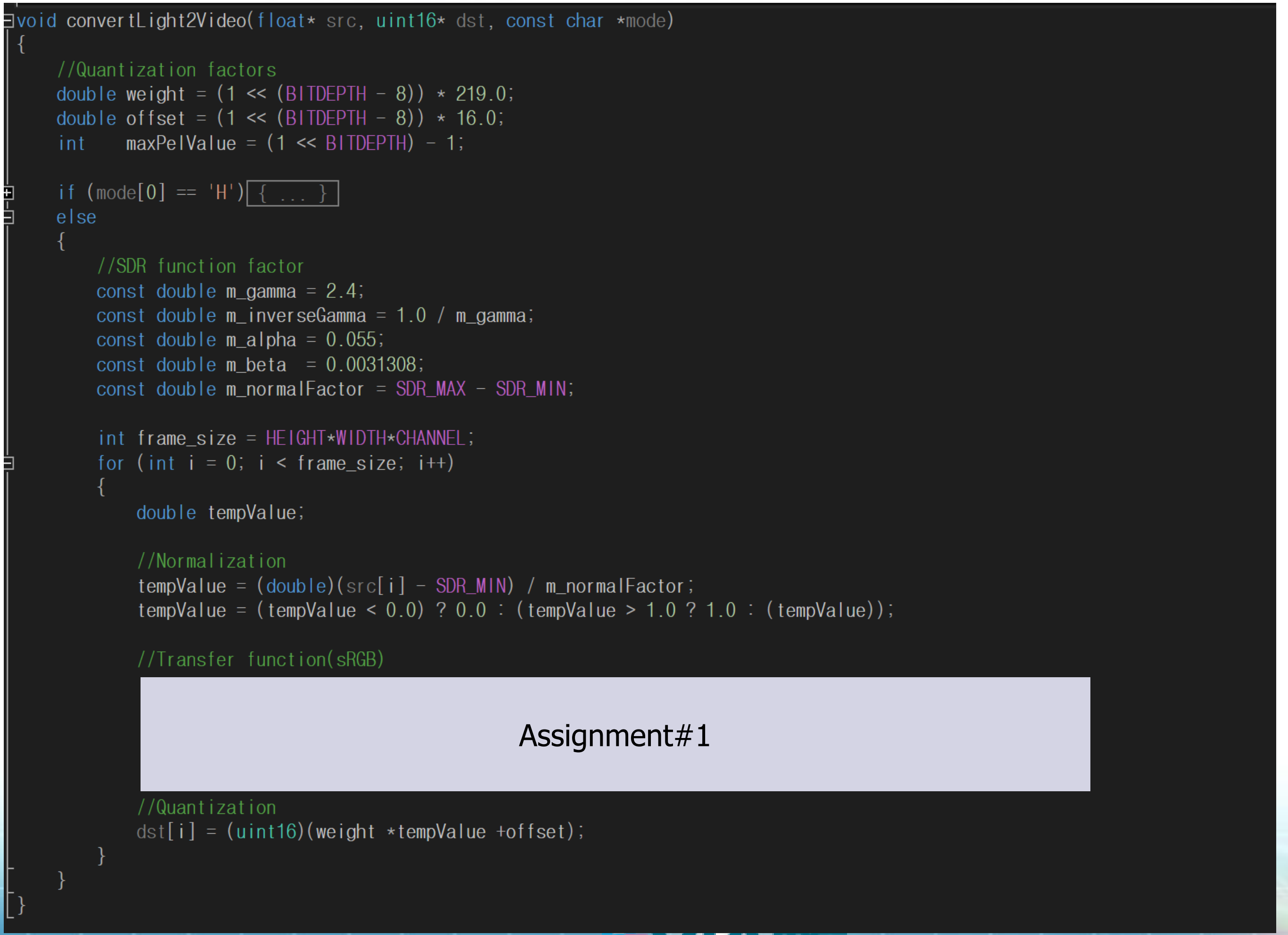
convertLight2Video VCCodeFunction

(Name)	convertLight2Video
File	e:₩조교₩2017₩2학기₩디
FullName	convertLight2Video
IsDefault	False
IsDelete	False
IsFinal	False
IsInjected	False
IsInline	False
IsOverloaded	False
IsSealed	False
IsTemplate	False
TypeString	void

(Name)
개체 이름을 설정/반환합니다.

Assignment

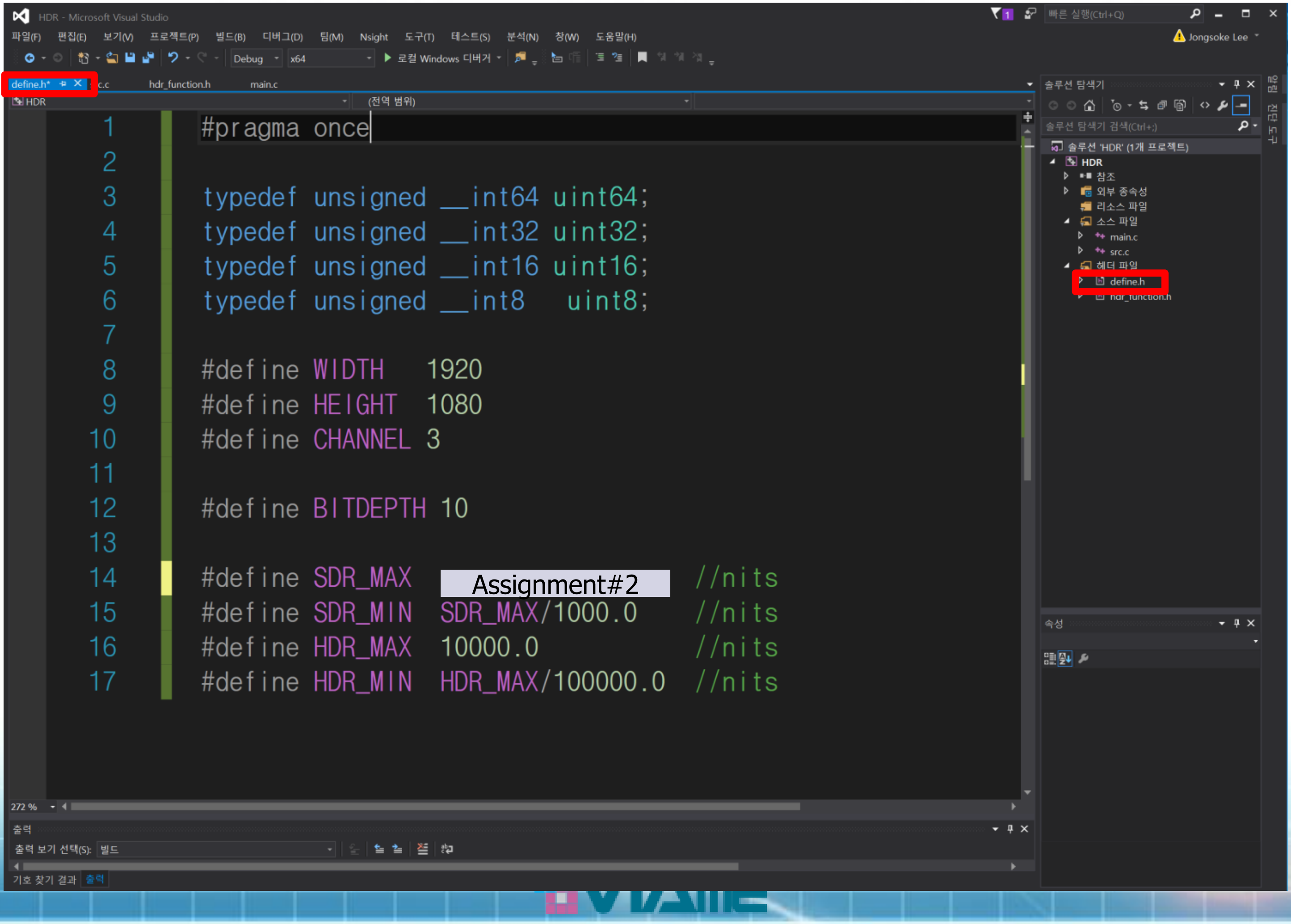
- Code



```
void convertLight2Video(float* src, uint16* dst, const char *mode)
{
    //Quantization factors
    double weight = (1 << (BITDEPTH - 8)) * 219.0;
    double offset = (1 << (BITDEPTH - 8)) * 16.0;
    int    maxPelValue = (1 << BITDEPTH) - 1;

    if (mode[0] == 'H'){ ... }
    else
    {
        //SDR function factor
        const double m_gamma = 2.4;
        const double m_inverseGamma = 1.0 / m_gamma;
        const double m_alpha = 0.055;
        const double m_beta  = 0.0031308;
        const double m_normalFactor = SDR_MAX - SDR_MIN;

        int frame_size = HEIGHT*WIDTH*CHANNEL;
        for (int i = 0; i < frame_size; i++)
        {
            double tempValue;

            //Normalization
            tempValue = (double)(src[i] - SDR_MIN) / m_normalFactor;
            tempValue = (tempValue < 0.0) ? 0.0 : (tempValue > 1.0 ? 1.0 : (tempValue));

            //Transfer function(sRGB)

                                Assignment#1

            //Quantization
            dst[i] = (uint16)(weight *tempValue +offset);
        }
    }
}
```

Assignment

- Code



```c
#pragma once

typedef unsigned __int64 uint64;
typedef unsigned __int32 uint32;
typedef unsigned __int16 uint16;
typedef unsigned __int8   uint8;

#define WIDTH   1920
#define HEIGHT  1080
#define CHANNEL 3

#define BITDEPTH 10

#define SDR_MAX  Assignment#2       //nits
#define SDR_MIN  SDR_MAX/1000.0     //nits
#define HDR_MAX  10000.0            //nits
#define HDR_MIN  HDR_MAX/100000.0   //nits
```