

360Tools 文件结构

文件包含结构，以360tools_conv.c为例：

```
360tools_conv.c
    360tools.h
        360tools_def.h
        360tools_args.h
        360tools_img.h
            360tools_def.h
        360tools_map.h
            360tools_def.h
```

在src/文件夹下，可以将文件分为两类：

1. mapping逻辑文件

- 360tools_erp.c 360tools_erp.h
- 360tools_isp.c 360tools_isp.h
- 360tools_ohp.c 360tools_ohp.h
- 360tools_cmp.c 360tools_cmp.h
- 360tools_tsp.c 360tools_tsp.h
- 360tools_ssp.c 360tools_ssp.h
- 360tools_cpp.c 360tools_cpp.h

2. 整体程序相关的文件

- 360tools.c 360tools.h
- 360tools_img.c 360tools_img.h
- 360tools_map.c 360tools_map.h
- 360tools_args.h // 处理程序入口输入参数
- 360tools_def.h // 数据结构定义、宏定义
- 360tools_tbl.c

一些关键的数据结构：

```
/* image buffer format =====/
typedef struct
{
    void          * buf_pad[4]; /* adress of each plane with padding */
    void          * buffer[4]; /* address of each plane */
    int            stride[4]; /* buffer stride */
    int            elevation[4]; /* buffer elevation */
    int            width; /* width */
    int            height; /* height */
    int            colorspace; /* color space */
} S360_IMAGE;

/* Spherical PSNR Sampling Location Structure =====/
typedef struct
{
    float lat;
    float lng;
}S360_SPH_COORD;

/* Latlong map for Y, chroma layers =====/
typedef struct
{
    S360_SPH_COORD * layer[2]; /* map for each plane*/
    int            width;
    int            height;
    int            pitch;
    int            yaw;
}S360_MAP;
```

主程序执行流程

- 360tools_conv.c

1. 处理程序入口参数，初始化变量信息，并对参数进行规范化判断。

相关函数：s360_args_parse_all (360tools_args.h)

例如，对于ERP2TSP：

```
if((cfmt == CONV_FMT_ERP_TO_TSP) &&
    ((w_out%4 != 0 || h_out%4 != 0) || (w_out != h_out*2)))
{
    s360_print("Invalid output resolution for TSP, suggested aspect "
              "ratio 2:1 and must be multiple of 4: %dx%d\n", w_out, h_out);
    s360_print("Suggested sample dimension: %dx%d\n", w_out, w_out/2);
    print_usage();
    goto END;
}
```

上面代码规定了输出的视频**宽和高必须是4的整数倍**，且**宽是高的2倍**。

2. 初始化输入输出图像

相关函数：s360_img_create (360tools_img.h)

```
imgi = s360_img_create(w_in, h_in, cs_int, opt);
imgo = s360_img_create(w_out, h_out, cs_int, opt);
```

3. 设置变换函数指针**fn_conv**

例如，对于ERP2TSP：

```
case CONV_FMT_ERP_TO_TSP:
    fn_conv = s360_erp_to_tsp;
    break;
```

4. 设置map （不必需）

<map 的作用？只有下采需要map？面的个数不同需要执行map？>

相关函数：s360_map_create (360tools_map.h)

```
map = s360_map_create(w_in, h_in, w_out, h_out, cfmt, opt, pitch, yaw);
```

5. 对每一帧数据进行处理

将fpi的数据读到imgi中，然后执行**fn_conv**函数（一共支持26种变换），将变换后的数据放在imgo中，然后进行写文件。

360tools_map.c 文件解读

注意：只有一些变换需要设置map

1. 支持的映射

映射名称	映射模型
Equirectangular projection (ERP)	经纬图
Icosahedral projection (ISP)	二十面体
Octahedron projection (OHP)	八面体
Cubemap projection (CMP)	六面体
Truncated Square Pyramid projection (TSP)	梯形金字塔
Segmented Sphere Projection (SSP)	
Reshaped Icosahedral projection (RISP)	
Reshaped Octahedron projection (ROHP)	
Reshaped Cubemap projection (RCMP)	

2. 支持的变换

变换格式 Conversion Format	对应程序入口参数 -f (num)
CONV_FMT_ERP_TO_CPP	0
CONV_FMT_ERP_TO_ISP	1
CONV_FMT_ISP_TO_ERP	2
CONV_FMT_ERP_TO_CMP	3
CONV_FMT_CMP_TO_ERP	4
CONV_FMT_ERP_TO_OHP	5
CONV_FMT_OHP_TO_ERP	6
CONV_FMT_ERP_TO_TSP	7
CONV_FMT_TSP_TO_ERP	8
CONV_FMT_ERP_TO_SSP	9
CONV_FMT_SSP_TO_ERP	10
CONV_FMT_ISP_TO_RISP	11
CONV_FMT_RISP_TO_ISP	12
CONV_FMT_CMP_TO_RCMP	13
CONV_FMT_RCMP_TO_CMP	14

CONV_FMT_OHP_TO_ROHP	15
CONV_FMT_ROHP_TO_OHP	16
CONV_FMT_ERP_TO_RISP1	21
CONV_FMT_RISP1_TO_ERP	22
CONV_FMT_ERP_TO_COHP	25
CONV_FMT_COHP_TO_ERP	26
CONV_FMT_CPP_TO_ERP	31
CONV_FMT_CPP_TO_ISP	32
CONV_FMT_CPP_TO_CMP	33
CONV_FMT_CPP_TO_OHP	34
CONV_FMT_CPP_TO_TSP	35
CONV_FMT_CPP_TO_SSP	36
CONV_FMT_ERP_TO_MERP	37
CONV_FMT_CPP_BYPASS	38
CONV_FMT_MAX	39

3. 在360tools_map.c中实现的变换函数fn_map（变换格式的子集）

函数名	功能	-f (num)
map_erp_cpp_to_cmp	ERP/CPP to CMP	3、31
map_erp_cpp_to_isp	ERP/CPP to ISP	1、32
map_erp_to_tsp	ERP to TSP	7
map_risp2	ISP to RISP RISP to ISP	11、 12
map_to_ohp	ERP/ROHP/CPP to OHP OHP to ROHP	5、16、34 15
map_erp_to_risp1	ERP to RISP1	21
map_erp_to_cohp	ERP to COHP	25
map_erp_to_merp	ERP to MERP	

4. 在360tools_map.c中实现的所有函数（包括了上面变换函数）

函数名	功能	被调用
map_init		
s360_map_delete		
erp2ohp_triangle		
erp2ohp_triangle_rev		

general_mapping		
general_mapping_rev		
erp2cohp_triangle		
erp2cohp_triangle_rev		
init_sph2ohp_map		
init_sph2cohp_map		
map_to_ohp	ERP to OHP	
map_erp_to_cohp		
erp2isp_triangle		
erp2isp_triangle_rev		
init_sph2isp_map		
erp2isp_triangle_half		
erp2isp_triangle_rev_half		
init_sph_to_risp1_map		
map_risp2	RISP2	
map_erp_cpp_to_isp	ERP to ISP	
erp2cmp_squ		
init_sph2cmp		
map_erp_to_risp1	ERP to RISP1	
map_erp_cpp_to_cmp	ERP to CMP	
rotate_sample	ERP to TSP	
erp2tsp_sample	ERP to TSP	
init_sph2tsp	ERP to TSP	
map_erp_to_tsp	ERP to TSP	

map_erp_to_tsp 函数解析

函数功能：计算TSP平面点 (x, y) 到球面点 (lat, lng) 的对应关系

1. w_squ

计算正方形的边长，取大于等于 w_dst 二分之一最近的偶数，与输出必须是偶数有关：

w_squ = NEAREST_EVEN(w_dst / 2.0); // 但是后面程序里面似乎有bug?

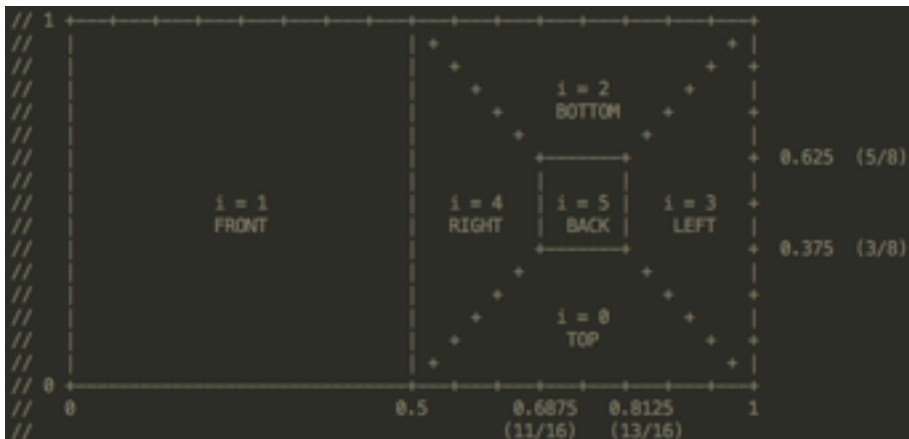
2. 分别计算Y、U、V对应的平面到球坐标的对应关系

```
/*for Y*/
init_sph2tsp(map->layer[0], w_dst, h_dst, w_squ, map->pitch, map->yaw);
/*for U & V*/
init_sph2tsp(map->layer[1], (w_dst >> 1), (h_dst >> 1), (w_squ >> 1), map->pitch, map->yaw);
```

3. init_sph2tsp

设置面和坐标的关系

Layout of TSP



执行erp2tsp_sample函数，计算map[mx]的值，也就是计算二维平面点 (mx, my) 对应到球面的位置 (lat, lng) 。

4. erp2tsp_sample 坐标的矩阵变换

顶点对应关系？：

```
const int tbl_vidx_erp2tsp[6][4] =
{
    { 4, 7, 1, 0 }, // Top
    { 7, 6, 0, 2 }, // Front
    { 6, 5, 2, 3 }, // Bottom
    { 4, 5, 7, 6 }, // Left
    { 0, 2, 1, 3 }, // Right
    { 1, 3, 4, 5 }, // Back
};
```

```

static void erp2tsp_sample(int squ_idx, S360_SPH_COORD *map, double j, double i, int pitch, int yaw)
{
    double vector_12[3], vector_13[3];
    double d12, d13, d12_scl, d13_scl;
    double xyz[3];
    int v_1_3d, v_2_3d, v_3_3d, v_4_3d;
    S360_SPH_COORD coord;

    v_1_3d = tbl_vidx_erp2tsp[squ_idx][0];
    v_2_3d = tbl_vidx_erp2tsp[squ_idx][1];
    v_3_3d = tbl_vidx_erp2tsp[squ_idx][2];
    v_4_3d = tbl_vidx_erp2tsp[squ_idx][3];

    v3d_sub(tbl_squ_xyz[v_2_3d], tbl_squ_xyz[v_1_3d], vector_12);
    d12 = v3d_norm(vector_12);
    v3d_sub(tbl_squ_xyz[v_3_3d], tbl_squ_xyz[v_1_3d], vector_13);
    d13 = v3d_norm(vector_13);
    d12_scl = d12 * i;
    d13_scl = d13 * j;
    v3d_affine(vector_12, d12_scl, tbl_squ_xyz[v_1_3d], xyz);
    v3d_affine(vector_13, d13_scl, xyz, xyz);
    rotate_sample(xyz, pitch, yaw);
    v3d_norm(xyz);
    cart_to_sph(xyz[0], xyz[2], xyz[1], &coord);
    *map = coord;
}

```

5. 三维坐标xyz，映射到二位球面经纬度 cart_to_sph

map_erp_to_cmp 函数解析

函数功能：计算CMP平面点 (x, y) 到球面点 (lat, lng) 的对应关系

1. 计算w_squ，即正方形的边长

计算正方形的边长，取大于等于 w_dst 四分之一最近的偶数，与输出必须是偶数有关：

w_squ = NEAREST_EVEN(w_dst / 2.0); // 但是后面程序里面似乎有bug?

2. init_sph2cmp

由于是规则的图形，所以对6个面直接计算：

```
static void init_sph2cmp(S360_SPH_COORD * map, int w_dst, int h_dst, int w_squ)
{
    int i;
    for (i = 0; i < 6; i++)
    {
        erp2cmp_squ(w_dst, h_dst, i, map, w_squ);
    }
}
```

3. 坐标的矩阵变换

4. 三维坐标xyz，映射到二位球面经纬度 cart_to_sph