

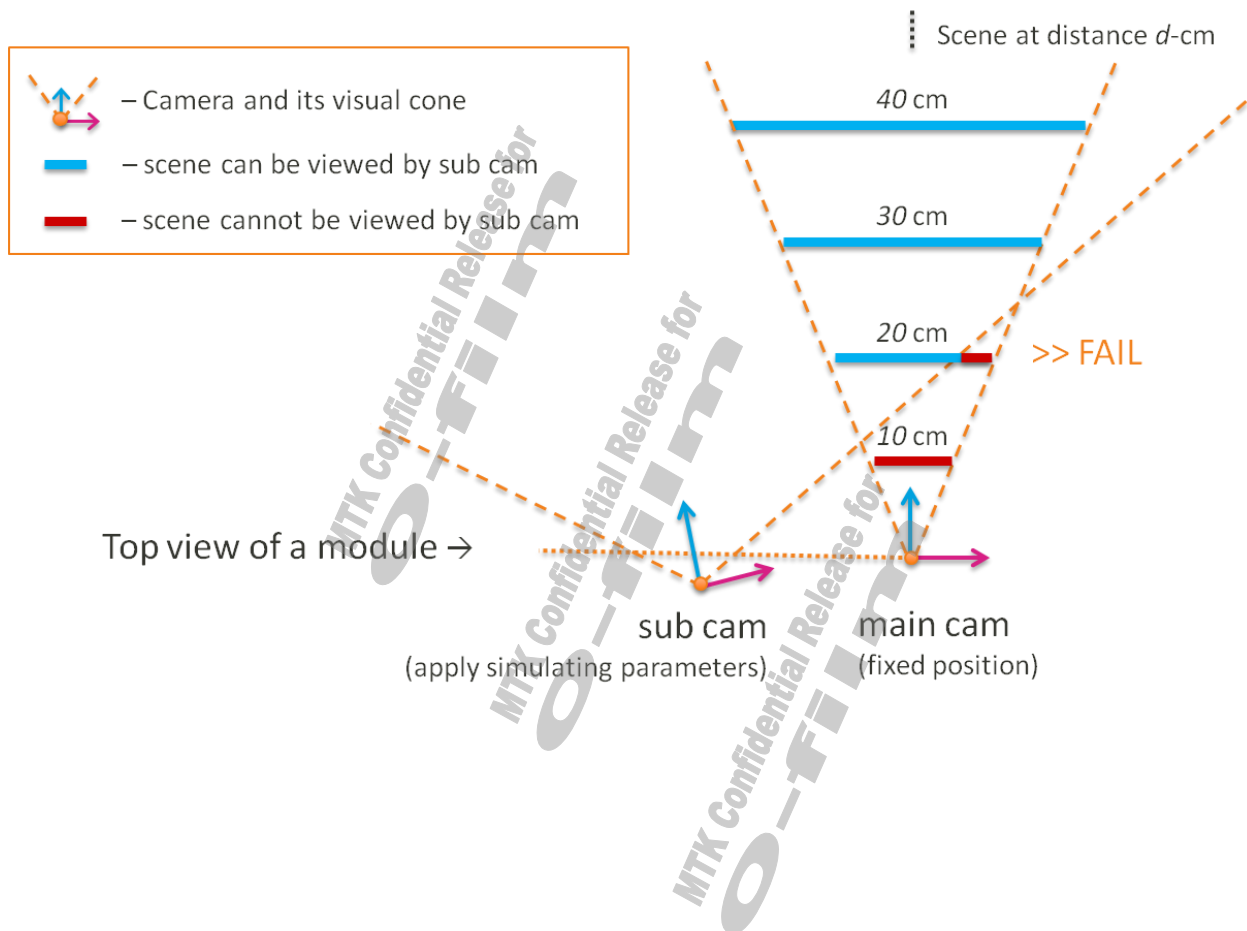
IQC Verification Tool

1. Target

The tool is designed to verify the quality of stereo module whether meet the requirement of error tolerance.

PASS: qualified to pass the requirement of error tolerance

FAIL: unqualified to pass the requirement of error tolerance



2. Verification process

Step0. Please refer to the “[IQC Verification Tool_setup_modulehouse.pdf](#)” to setup the verification environment. Please check the following items:

1. The image center of the main camera needs to be aligned with the center of the designed pattern.
2. The sensor plane needs to be parallel to the plane of the target pattern, i.e. the optical axis needs to be vertical to the plane of the target pattern - no perspective projection distortion in the image space.
3. The captured image needs to avoid planner rotation of the designed pattern in the image space.
4. The captured images from the two distances need to be parallel.

Step1. Adjust the focus settings to make the captured images focused on the designed pattern from 20 cm, and take the first stereo image.

Step2. Adjust the focus settings to make the captured images focused on the designed pattern from 90 cm, and take the second stereo image.

Step3. Input the processed stereo raw images from two distances to run the verification tool.

Step4. Output the results.



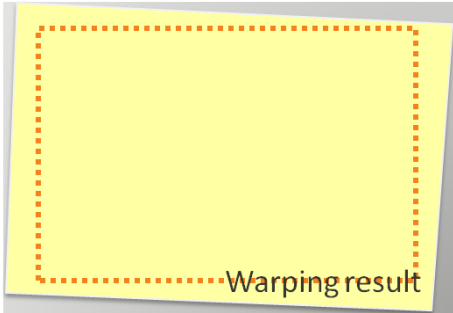
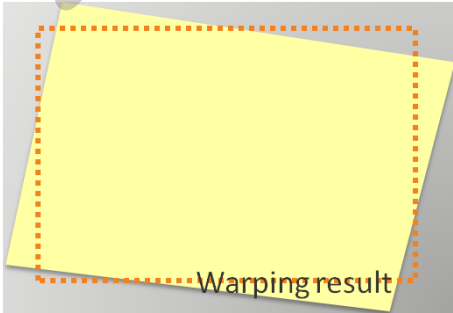
3. Functions

3.1 Function – IQC

Function interface:

`int IQC (int *gVerify_Items, unsigned short *main_raw_near, unsigned short *main_raw_far, int main_w, int main_h, unsigned short *sub_raw_near, unsigned short *sub_raw_far, int sub_w, int sub_h, char configname[])`

Input	Type	Description
gVerify_Items	int *	Store the output information. The array length is equal to 180.
main_raw_near	unsigned short *	The raw image (bit depth: 16) of main camera captured from 20 cm.
main_raw_far	unsigned short *	The raw image (bit depth: 16) of main camera captured from 90 cm.
main_w	int	The width of image in main camera.
main_h	int	The height of image in main camera.
sub_raw_near	unsigned short *	The raw image (bit depth: 16) of sub camera captured from 20 cm.
sub_raw_far	unsigned short *	The raw image (bit depth: 16) of sub camera captured from 90 cm.
sub_w	int	The width of image in sub camera.
sub_h	int	The height of image in sub camera.
configname	char []	The file name of the configuration file.

Output	Description
return	0: Error 1: Success
gVerify_Items[0]	Show the status of this stereo camera according to the following information (gVerify_Items[1-18]). 0: PASS \geq 1: FAIL
gVerify_Items[1]	Geometric quality level. (20 cm) 0: PASS \geq 1: FAIL
<div><div>PASS case</div><div><p>Warping result</p></div></div> <div><div>FAIL case</div><div><p>Warping result</p></div></div>	
Pixel distance from an image boundary to the FAIL region (#) gVerify_Items[2-5]	

<div> <div> <div>PASS case</div> <div>WARPING RESULT</div> </div> <div> <div>FAIL case</div> <div>WARPING RESULT</div> </div> </div>	
gVerify_Items[2]	# pixels from an image top boundary to the FAIL region. (20 cm)
gVerify_Items[3]	# pixels from an image right boundary to the FAIL region. (20 cm)
gVerify_Items[4]	# pixels from an image bottom boundary to the FAIL region. (20 cm)
gVerify_Items[5]	# pixels from an image left boundary to the FAIL region. (20 cm)
gVerify_Items[6]	The average intensity of main camera. (≥ 40) (20 cm)
gVerify_Items[7]	The difference of average intensity between main camera and sub camera. (20 cm)
gVerify_Items[8]	The captured pattern if the MTK designed chart. (20 cm) 0: MTK chart 1: Not MTK chart
gVerify_Items[9]	For judging a scene is a MTK chart nor not. (20 cm)
gVerify_Items[10-18]	gVerify_Items[10-18] are the results of data at 90 cm and the meaning of gVerify_Items[10-18] are the same to the gVerify_Items[1-9].

3.2 Sample code

```
// parameters
char ConfigName[] = "Config.cfg";    // the file name of the configuration file
int *gVerify_Items, main_w, main_h, sub_w, sub_h;
unsigned short *raw_main_near, *raw_main_far, *raw_sub_near, *raw_sub_far;

// initial
main_w = 4208;                        // the with of main camrea
main_h = 3120;                        // the height of main camera
sub_w = 2560;                         // the width of sub camera
sub_h = 1920;                         // the height of sub camera
gVerify_Items = new int[180];
raw_main_near = new unsigned short[main_w*main_h];    // main camera at 20 cm
raw_main_far = new unsigned short[main_w*main_h];    // main camera at 90 cm
raw_sub_near = new unsigned short[sub_w*sub_h];       // sub camera at 20 cm
raw_sub_far = new unsigned short[sub_w*sub_h];        // sub camera at 90 cm

// load two captured data (not supply this function)
LoadData(raw_main_near, raw_main_far, raw_sub_near, raw_sub_far);

// verify
int status = IQC(gVerify_Items, raw_main_near, raw_main_far, main_w, main_h, raw_sub_near,
raw_sub_far, sub_w, sub_h, ConfigName);
```