

RGBTD 데이터 가공법

조 원

목 차

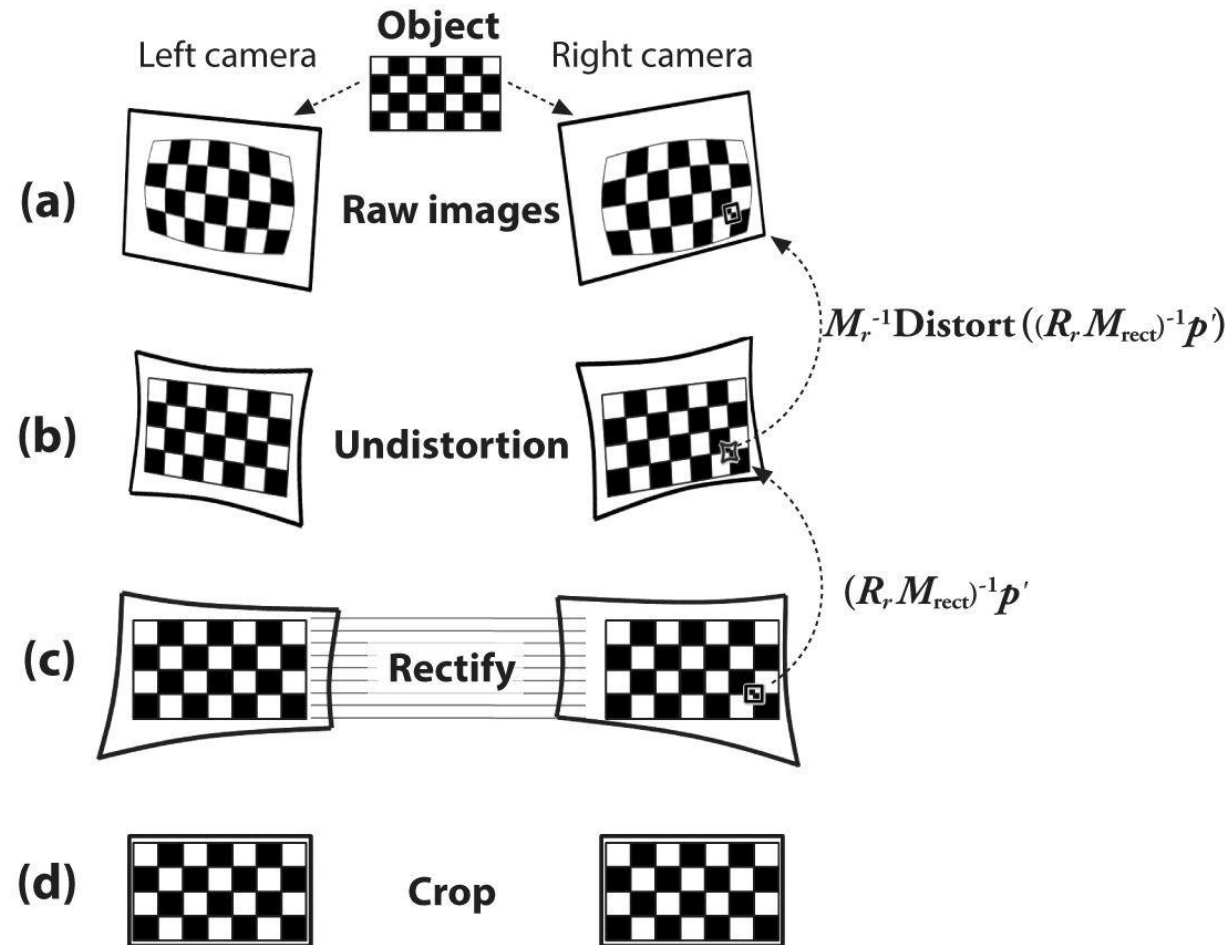
1. Stereo Rectification

2. RGB-T warping

3. Matlab code

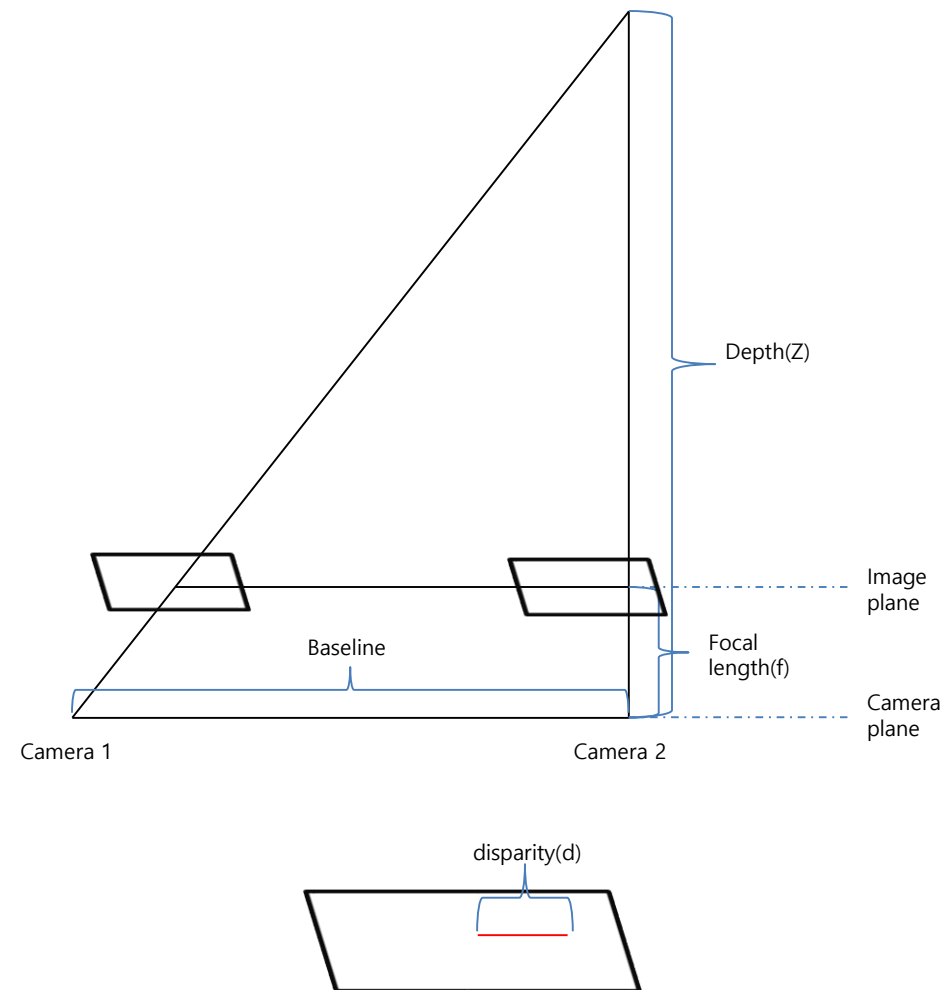
1. Stereo Rectification

Stereo Rectification



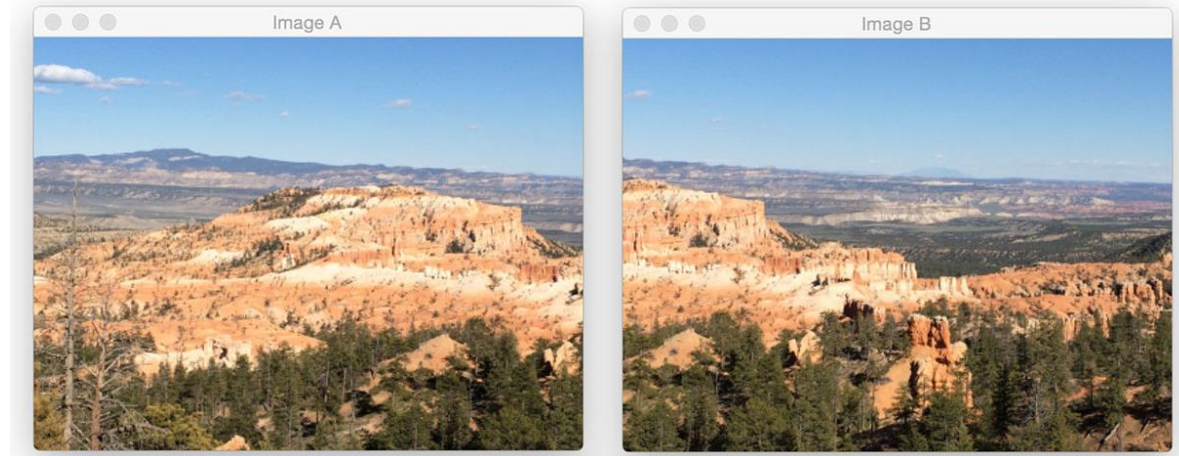
Stereo Rectification

- 거리 정보를 알기 위해서는 두 영상간 대응되는 점이 얼마나 차이나는 지 확인해야한다
- 카메라에 가까울 수록 두 영상간 차이가 크고 멀수록 차이가 작다
- 두 영상간 차이를 disparity(d)라고 정의했을 때 다음처럼 나타낼 수 있다. $d:f = B:Z \rightarrow Z = fB/d$



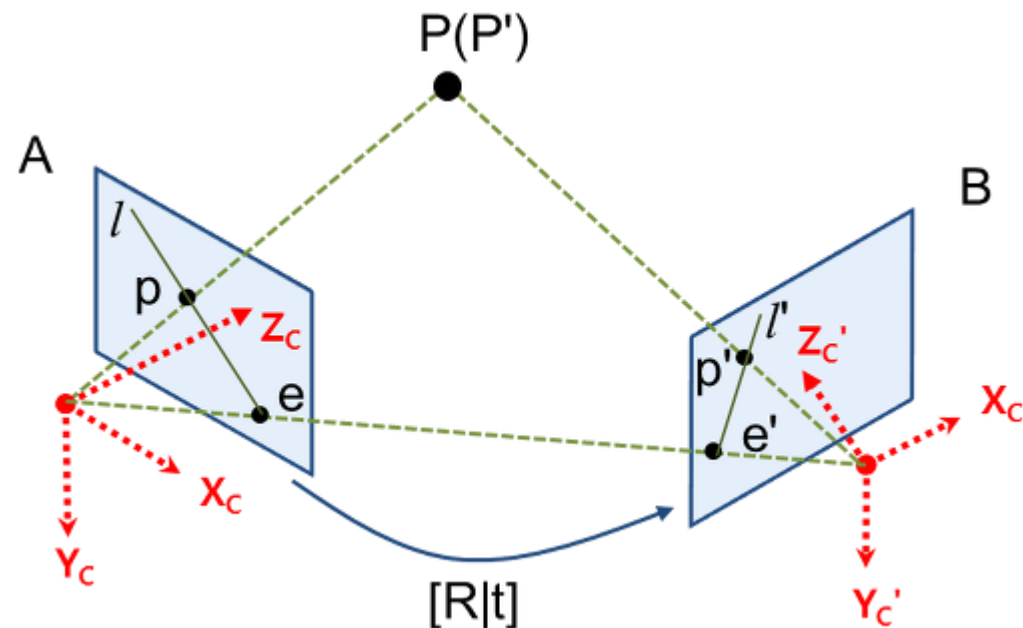
Stereo Rectification

- Disparity를 구하기 위해서는 3D 공간 상에서 두 영상에 투영된 점을 찾아야한다 -> 즉 두 영상에서 동일한 점을 찾는 문제
- 두 영상에서 $M \times N$ 의 모든 픽셀을 비교하면서 찾으려면 너무 긴 computational cost가 든다



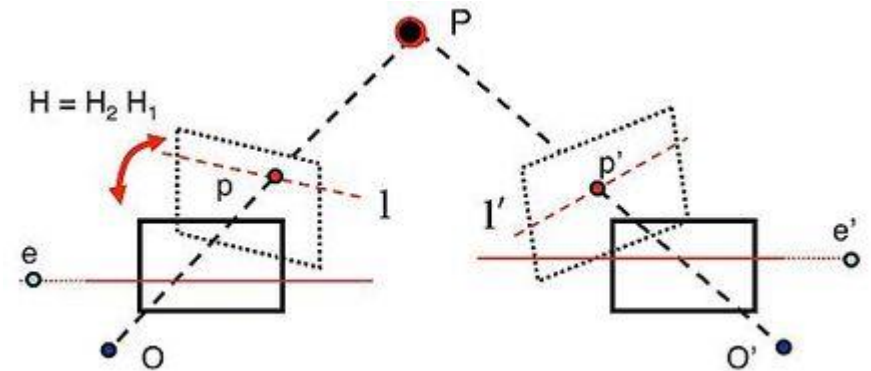
Stereo Rectification

- 점 P 를 각각 영상에 투영한 점을 p, p' 이라고 하면 거리정보를 알기 이전에 영상 A에서 점 p 의 3D공간 점인 P 는 선분 pP 에 어디든 위치할 수 있어 복원 불가능
이므로 영상 B의 점 p' 또한 유일하게 결정할 수 없다
- 두 카메라 원점을 이은 선분이 두 영상과 만나는 점인 e, e' 을 epipole 이라고 부르고 투영점과 epipole을 잇는 직선 l, l' 을 epipolar line이라고 부른다
- 점 P 는 거리정보가 없을 때 유일하게 결정할 수 없으나 카메라 A의 원점과 영상 A의 점 p 를 잇는 직선 상에 존재하고 이를 영상 B에 투영시키면 epipolar line인 l' 은 유일하게 결정할 수 있다
- 이를 통해 직선 l 위의 점 p 에 대응하는 점인 p' 의 l' 을 구할 수 있게 되고 영상 전체에서 l' 으로 탐색 범위를 줄일 수 있다



Stereo Rectification

- 대응점 탐색을 위해 구한 두 epipolar line을 수평으로 맞게 변환한다
- 변환시에는 마지막에 두 영상을 같은 각도 반대 방향으로 rotation한다
- 이 때문에 카메라가 수평으로 놓아져 있지 않으면 우측 사진처럼 기울어져서 rectification이 진행될 수도 있다



2. RGB-T warping

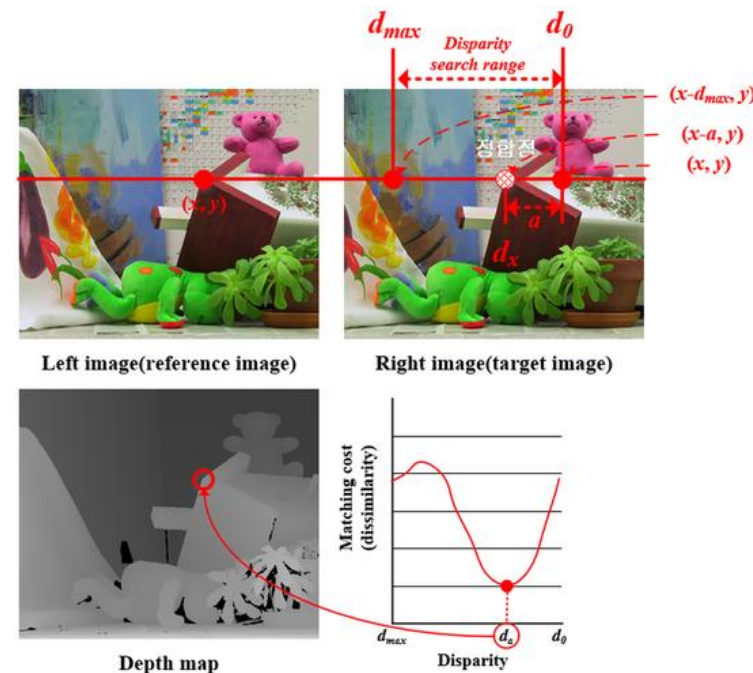
RGB-T warping

- Rectify된 두 영상을 가지고 disparity를 구한다
- 이 disparity는 왼쪽 영상에 맞게 나온다
- disparity는 깊이정보를 구하는 데에도 사용될 수 있지만 두 영상간 차이이기에 rectify된 두 영상을 겹치는 데에도 사용할 수 있다



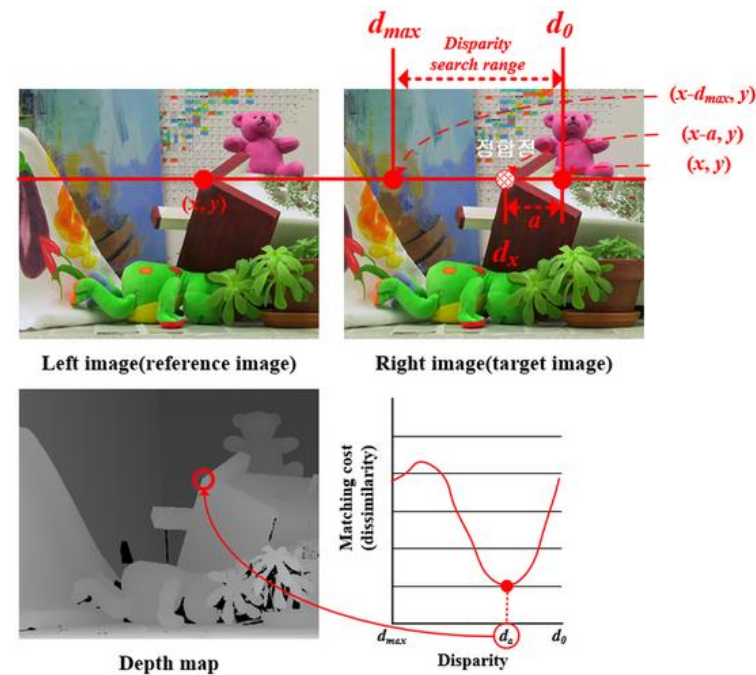
RGB-T warping

- Disparity를 구할 때는 기존에 rectify 시킨 영상으로 일정 범위만 탐색하는 local matching 방식을 이용하는데 이는 일정 범위 내에서 대응되는 점에 대해 비유사도가 가장 낮은 점을 찾는 방식이다
- 이로 인해 disparity의 정확성은 영상의 feature에 따라 결정될 수 밖에 없다
- 같은 도메인의 영상인 RGB-RGB, Thermal-Thermal의 경우는 영상에 feature가 어느정도 존재할 시 disparity를 구할 수 있다 (Thermal이 일반적으로 RGB보다 feature가 적기에 부정확하다)



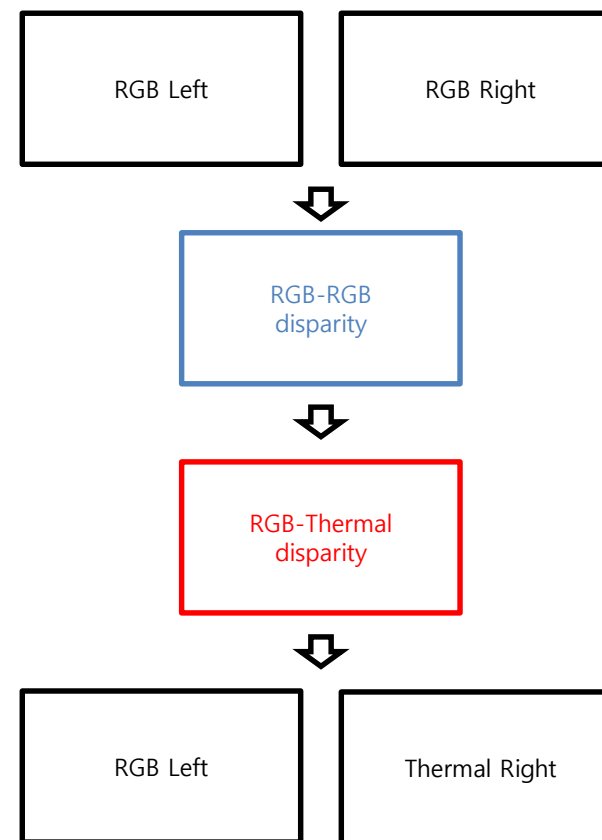
RGB-T warping

- 그러나 RGB-Thermal의 경우는 다른 도메인이기 때문에 표현되는 픽셀의 feature가 다를 수 밖에 없고 local matching 방식을 사용 하는 것은 disparity의 정확도를 매우 낮추게 된다



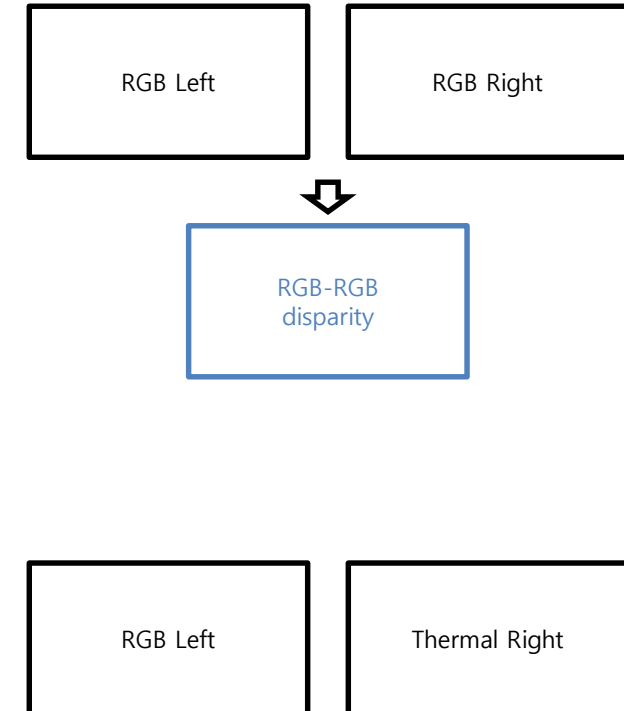
RGB-T warping

- 이에 대한 해결책으로 어느정도 신뢰도가 있는 RGB-RGB disparity를 RGB-T의 disparity로 근사시키는 방식을 생각할 수 있다
- RGB-RGB disparity를 그대로 RGB-T 쌍에 가져다 쓸 수 없는 이유는 두 영상을 찍은 카메라의 위치가 서로 달라 epipolar line이 달라져 rectification이 서로 다르기 때문이다



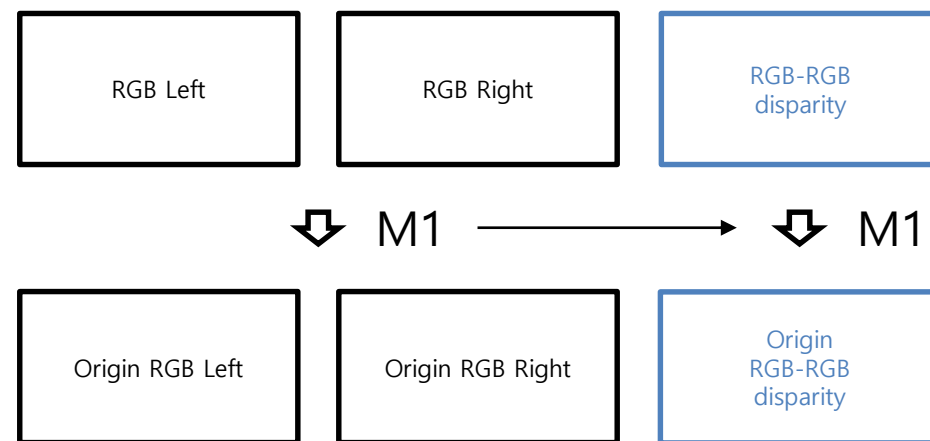
RGB-T warping

- 첫번째 단계는 RGB-RGB로 rectify한 두 영상과 disparity를 구하고 RGB-T로 rectify한 두 영상을 구한다



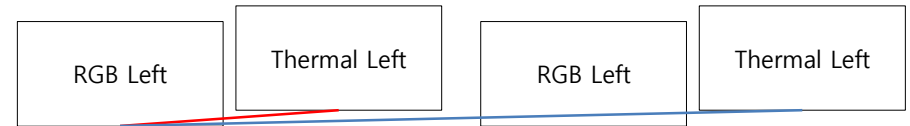
RGB-T warping

- 구한 RGB-RGB disparity를 RGB-T disparity로 변환시키는 과정이 필요한데 이를 위해서는 RGB-RGB로 rectify 시켰던 영상을 기존 초기 영상으로 되돌리는 작업이 필요하다
- 이 관계를 M1이라고 하고 이를 구한 뒤 disparity에도 적용하여 기존 초기 영상의 disparity로 바꿔준다
- 이는 초기 왼쪽 RGB의 disparity가 된다



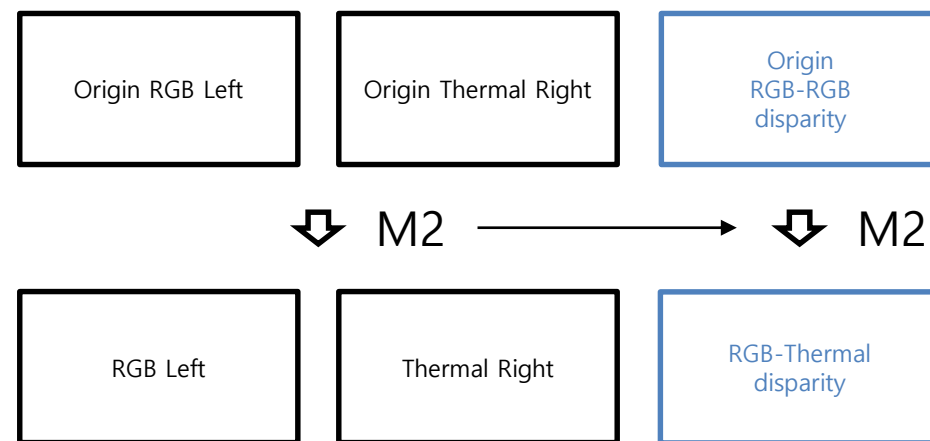
RGB-T warping

- 초기 왼쪽 RGB의 disparity를 구하였으니 왼쪽 RGB와 Thermal 간 rectification을 진행한다
- 우측 그림처럼 실제로 RGB와 Thermal은 높이 차이가 있게 배치되어 있기 때문에 rectification 시 결과가 회전되는 정도를 줄이기 위해 왼쪽 RGB 카메라와 사이의 각이 가장 작은 오른쪽 Thermal 카메라를 선택했다



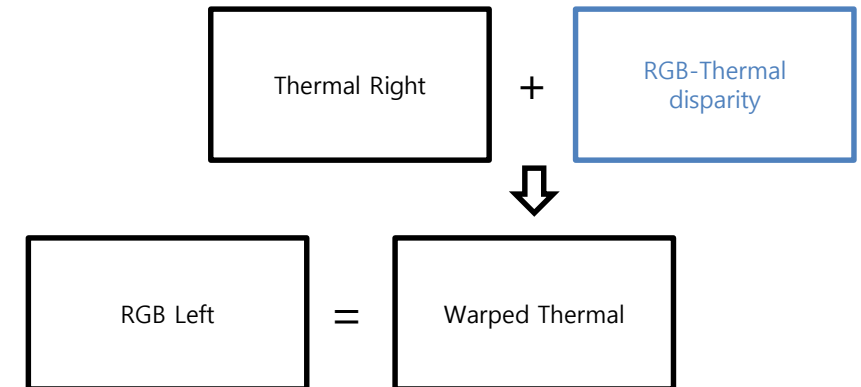
RGB-T warping

- 왼쪽 RGB와 오른쪽 Thermal간 rectification을 진행한 후 이 관계를 M2라고 했을 때 M2의 관계로 초기 RGB disparity를 변환 시킨다
- 변환시킨 후 중간중간 픽셀의 구멍 뚫림 현상이 나타날 수 있는데 이는 여러 보간 방식으로 해결할 수 있다
- 현 Matlab 코드에서는 이동평균으로 보간 하였다



RGB-T warping

- 구해진 RGB-Thermal disparity를 적용해 Thermal 의 픽셀을 RGB로 수평 이동하면 서로 겹치는 pair 영상을 만들 수 있다



3. Matlab code

Matlab code

- 환경은 Matlab R2019a이며 vision toolbox를 필요로 합니다
- 사용한 함수가 버전을 타고 이 함수 내부에는 내장 함수가 있어 버전을 맞춰주셔야 합니다

Matlab code

- doWarpPairImage.m 는 script 이며 파라미터 설정을 할 수 있습니다
- 실행 = F5
- isSave는 가공한 영상 저장 여부이며 저장을 원할 시 true로 해주시면 됩니다
- isShow는 매 영상마다 가공된 것을 화면에 띄우고 싶으실 때 true로 해주시면 됩니다
- startSet과 finalSet은 시작 set과 끝 set의 번호입니다
- step은 가공시 몇 프레임씩 건너 뛰며 가공할 것인지 정하는 것입니다
모두 가공하시려면 1로 설정하면 됩니다
- savePath는 저장 경로, saveImageName은 가공후 저장될 영상 이름,
imageType은 압축 형식입니다
- path_rrParams는 RGB-RGB를 Matlab APP에서 스테레오 캘리브레이션하고 나온 값의 경로이며 path_rtParams는 RGB-Thermal의 경우의
경로입니다
- Base_path_leftRGB는 왼쪽 RGB 영상의 경로이며 base_path_rightRGB
는 오른쪽 RGB 영상의 경로, base_path_rightTher은 오른쪽 Thermal
영상의 경로 입니다

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Params %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
isSave = false;
isShow = true;

startSet = 1;
finalSet = 8;
step = 10;
reSize = [512, 640];

savePath = "D:/LAB/sejongDatasetV3/Rect_test/";
saveImageName = "I%07d";

path_rrParams = "./parameters/Params(MatAPP)/RGB/Stereo/stereoParams.mat";
path_rtParams = "./parameters/Params(MatAPP)/RGB-T/R_T/stereoParams.mat";

base_path_leftRGB = "D:/LAB/sejongDatasetV3/Set%02d/Set%02d/decode/RGB_L/";
base_path_rightRGB = "D:/LAB/sejongDatasetV3/Set%02d/Set%02d/decode/RGB_R/";
base_path_rightTher = "D:/LAB/sejongDatasetV3/Set%02d/Set%02d/decode/Thr_R/";

imageType = "png"; % "tif" , "jpg" ...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Matlab code

- warpingImagePair.m 은 이전 설명드린 RGB-T warping이 구현되어 있는 코드입니다
- 내부에 disparityRange라는 변수가 [0 96]으로 설정되어 있는데 이는 disparity를 구할 때 대응점을 찾는 픽셀 수평 범위이며 96이 최대입니다 이 범위가 넓을 수록 가까이 있는 물체가 많을 때 대응점 탐색에 유리하지만 그만큼을 disparity 좌측에서 잘라야 하기에 정보의 손실이 있습니다
- 해당 코드를 돌리시면 disparityRange의 크기를 96으로 가장 크게 설정해 두었지만 데이터 셋 중 두 영상의 차이가 이보다 크게 날 정도로 가까이 있는 물체에 대해서는 오류가 있을 수 있습니다 또한 RGB-
RGB disparity를 RGB-Thermal disparity로 근사시켜 사용하는 방법
기에 Thermal영상에서의 그만큼의 오차가 있을 수 있습니다

Matlab code

- parseInputsCalibrated.m과 StereoParametersImpl.m Matlab에 구현되어 있는 Rectification 함수를 수정하여 만든 함수들이며 이 두 함수를 통해 warpImagePair.m 함수 내부에서 rectification과 rectification 관계를 얻을 수 있습니다