

Supplementary Material for RGB-Multispectral Matching: Dataset, Learning Methodology, Evaluation

Fabio Tosi* Pierluigi Zama Ramirez* Matteo Poggi*
 Samuele Salti Stefano Mattoccia Luigi Di Stefano
 CVLAB, Department of Computer Science and Engineering (DISI)
 University of Bologna, Italy
 {fabio.tosi5, pierluigi.zama, m.poggi}@unibo.it

In this document, we provide additional details concerning CVPR submission “RGB-Multispectral Registration: Dataset, Learning Methodology, Evaluation”.

1. Calibration of our trinocular rig

In this section, we detail the calibration procedure needed to gather images and either i) annotate them with accurate ground-truth depth labels or ii) distill accurate proxy labels.

We define the three cameras of our setup as RGB , MS , and RGB^2 , with RGB and RGB^2 being the two RGB cameras with identical resolution ($W_{RGB} \times H_{RGB}$) and MS the multi-spectral lower resolution device ($W_{MS} \times H_{MS}$). We define raw images acquired by the RGB , MS , and RGB^2 cameras as rgb , ms , and rgb^2 respectively.

First of all, we calibrate each camera independently, using the standard calibration procedure from OpenCV based on the detection of corners on a chessboard pattern. We found out empirically that we could detect corners with standard algorithms also in our MS images, simply by processing a grayscale image defined as the average across channels. This allows for straightforward calibration of the MS camera.

Then, following again the standard procedure from OpenCV, we calibrate the $RGB - RGB^2$ stereo system and rectify images. In this way we obtain rectified images rgb_2, rgb_2^2 for the $RGB - RGB^2$ stereo system.

Finally, we calibrate the *unbalanced* stereo system $RGB - MS$ following the procedure discussed below.

Unbalanced Stereo Rectification. To rectify images acquired by the $RGB - MS$ unbalanced stereo system, we follow the *unbalanced rectification* scheme sketched in [1], yielding images that are rectified when brought to the same resolution by means of up-sampling or down-sampling operations solely.

We denote the camera with the smaller $HFOV$ as j while the other one as i .

$$\begin{cases} i = RGB, j = MS & \text{if } HFOV_{MS} < HFOV_{RGB} \\ i = MS, j = RGB & \text{if } HFOV_{RGB} < HFOV_{MS} \end{cases} \quad (1)$$

By modifying the intrinsic parameters of i , we simulate a crop and scale change so as to match the $HFOV$, Aspect Ratio (AR) and size of j . Then, we compute the rectification transformation based on these new parameters.

Hence, we compute the new width and height of i , \hat{W}_i and \hat{H}_i , which we use to crop the image with the larger $HFOV$ so as to match the smaller $HFOV$ one while preserving the aspect ratio

$$\hat{W}_i = 2 \tan \frac{HFOV_j}{2} f_i \quad (2)$$

$$\hat{H}_i = \frac{H_j}{W_j} \hat{W}_i \quad (3)$$

Then, we change the intrinsic parameters of i to simulate the crop and resize, and can thus match the resolution of j as follows:

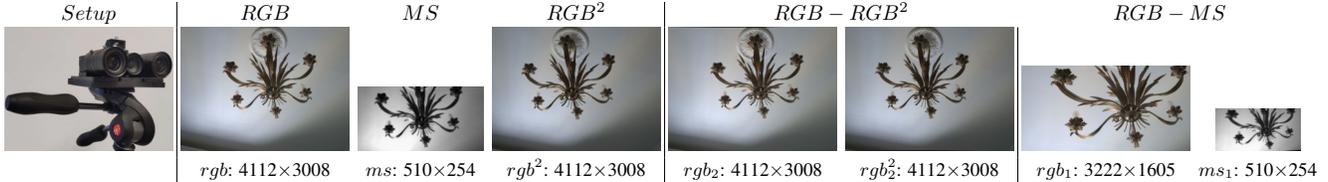


Figure 1. **Rectification example.** From left to right: our trinocular rig, rgb , rgb^2 , and ms raw images acquired by our rig. rgb_2 , rgb_2^2 rectified balanced stereo pair from of the $RGB - RGB^2$ stereo system. rgb_1, ms_1 unbalanced rectified stereo pair from of the $RGB - MS$ stereo system.

$$\hat{A}_i = \begin{bmatrix} f_x^i \cdot \frac{W_j}{\hat{W}_i} & 0 & (u_0^i - \frac{W_i - \hat{W}_i}{2}) \cdot \frac{W_j}{\hat{W}_i} \\ 0 & f_y^i \cdot \frac{H_j}{\hat{H}_i} & (v_0^i - \frac{H_i - \hat{H}_i}{2}) \cdot \frac{H_j}{\hat{H}_i} \\ 0 & 0 & 1 \end{bmatrix}$$

Then, we estimate the rectification transformation as we would have two cameras of height H_j and width W_j , finding the new intrinsics and rotations to map the initial image plane into the rectified one. As we have estimated the intrinsic matrices at the resolution of j , we rescale the intrinsic matrix of i with a vertical and horizontal scale factors equal to $\frac{\hat{H}_i}{H_j}$ and $\frac{\hat{W}_i}{W_j}$, respectively, in order to adjust the focal length and piercing point of the camera.

Finally, we can rectify the unbalanced pair obtaining two rectified images rgb_1, ms_1 for the $RGB - MS$ stereo system.

Fig. 1 shows an example of images before and after the rectification procedures described so far.

2. Disparity Warping

When using RGB^2 , both during the proxy labels creation or the ground-truth acquisition, we need to warp the left ground-truth $Disp_2$ aligned with rgb_2 of the $RGB - RGB^2$ stereo system in order to obtain the ground-truth $Disp_1$ aligned with the rgb_1 of the $RGB - MS$ stereo system.

We know that the rectification transformation is only a change of intrinsic parameters and a rotation, thus an homography. Therefore, we can compute the mapping between pixels of rgb_2 of the $RGB - RGB^2$ stereo system, with coordinate (u, v) , and pixels of the left image rgb_1 of the $RGB - MS$ stereo system, with coordinate (u', v') as:

$$\begin{pmatrix} u' \\ v' \\ 1' \end{pmatrix} = A_1 R_1 R_2^{-1} A_2^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (4)$$

where A_1, R_1 and A_2, R_2 are the intrinsic and rectification rotation matrix of the camera RGB in the $RGB - MS$ and $RGB - RGB^2$ stereo systems, respectively.

Once this mapping is known, we can perform a backward warping to obtain $Disp_1$ from $Disp_2$. However, we need to modify the disparity values according to the 3D rotation and baseline change before warping. Thus, given the disparity map $Disp_2$, we first transform it into the corresponding depth map D_2

$$D_2 = \frac{f_2 b_2}{Disp_2} \quad (5)$$

where f_2 is the focal length of the rectified rgb_2 and b_2 is the baseline of the stereo system $RGB - RGB^2$. Then, we back-project each pixel of rgb_2 into the 3D space using D_2 and we rotate it, obtaining the pixel in the rgb_1 reference frame:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R_1 R_2^{-1} D_2 A_2^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (6)$$

In this way we can create a depth map $D_{2 \rightarrow 1}$ for which any pixel (u, v) contains the depth value of the corresponding pixel aligned in the rgb_1 reference frame, z' . At this point we perform the backward warping of the depth:

$$D_1 = \phi(D_{2 \rightarrow 1}) \quad (7)$$

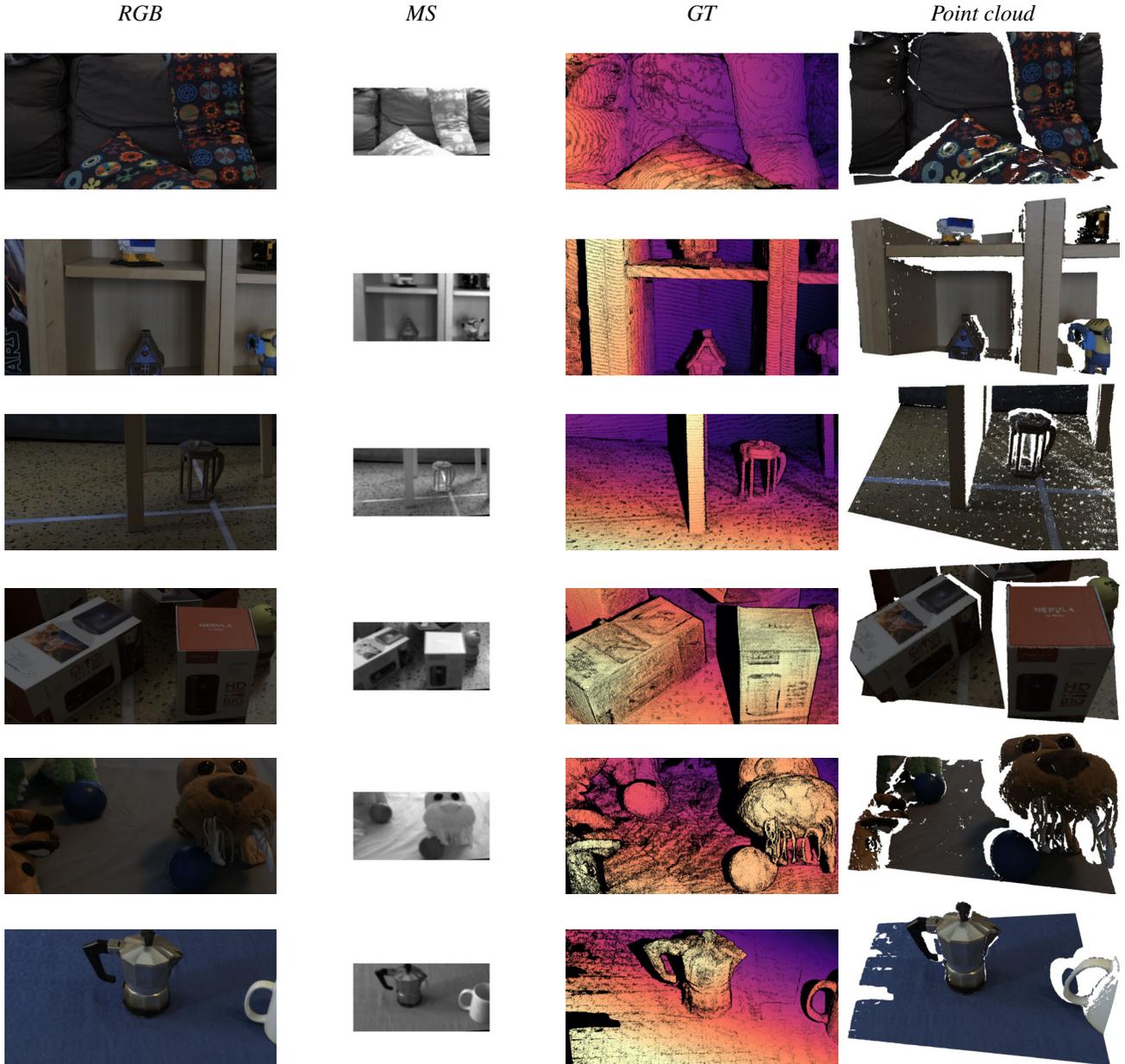


Figure 2. **Ground Truth Examples.** From left to right: RGB image, MS image (mean across channels), ground-truth disparity map obtained with our acquisition pipeline, point cloud visualization.

where ϕ is the backward warping operation that use the mapping defined at Eq. 4 and D_1 is the depth map aligned with rgb_1 . Finally we transform it to the ground disparity map of rgb_1 as:

$$Disp_1 = \frac{f_1 b_1}{D_1} \quad (8)$$

where f_1 and b_1 are the focal length of rgb_1 and the baseline of the $RGB - MS$ stereo system.

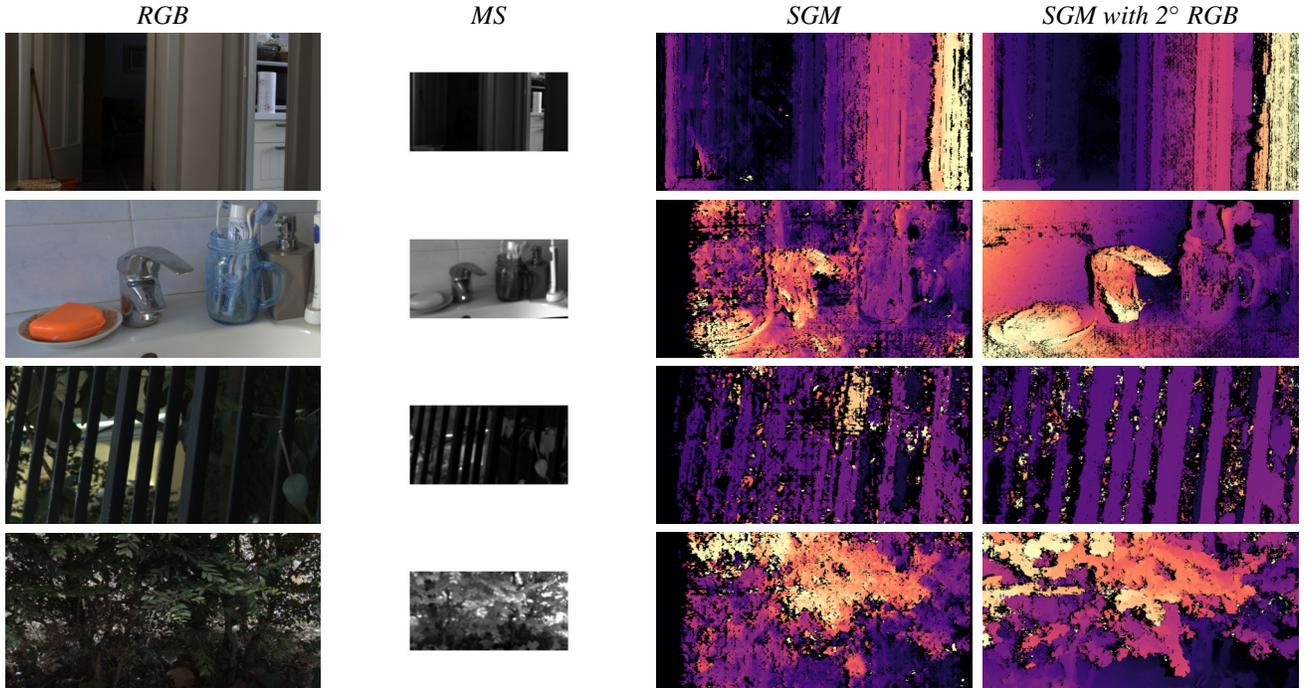


Figure 3. **Unlabeled Examples.** From left to right: RGB image, MS image (mean across channels), SGM computed between grayscale representation of RGB and MS images, SGM computed on two RGB cameras and then warped.

3. Additional Qualitative Results

In this section, we present additional qualitative results. In Fig. 2 we show examples of RGB-MS pairs with available disparity ground-truths. In Fig. 3 we present examples of the images acquired in uncontrolled environments used to train our deep architecture by our proxy supervision approach. In particular, in the last two columns we show SGM proxy labels computed on a RGB image converted into grayscale and a single channel MS image attained by computing the mean across all channels at each pixel, as well as the SGM proxy labels obtained by two RGB images and then warped. Finally, in Fig. 4 we report additional qualitative results obtained by our network using PSM as backbone and trained also with auxiliary synthetic data (i.e., the best configuration of Tab. 1 and Tab 2. of the main paper).

4. Qualitative Registration Results.

To provide hints on the registration quality, in Fig. 5 we report qualitatives obtained by using either the disparity predicted by our network or the ground-truth to warp the low resolution MS image into the high-resolution RGB image. We notice how the two warped images appear very similar, which vouches for the effectiveness of our cross-modal registration architecture.

5. Additional Implementation Details

We provide here additional details regarding our deep cross-spectral network architecture as well as on the adopted continuous output representation.

5.1. Stereo Backbones

As for the stereo backbones, we follow the original implementation of two popular architectures: PSM [2] and GWC [3]. For both of them, we follow the idea proposed in [4] to exploit a different combination of feature maps computed at different spatial resolutions in order to capture both local and global context.

PSM [2]: For the PSM stereo backbone, we use the original deep feature extractor, consisting of Spatial Pyramid Pooling (SPP) modules as Φ_θ while the stereo matching cost probabilities are extracted from stacked hourglass 3D convolutions on the cross-spectral cost volume computed by Ψ_θ . More specifically, we perform bilinear interpolation on features of size

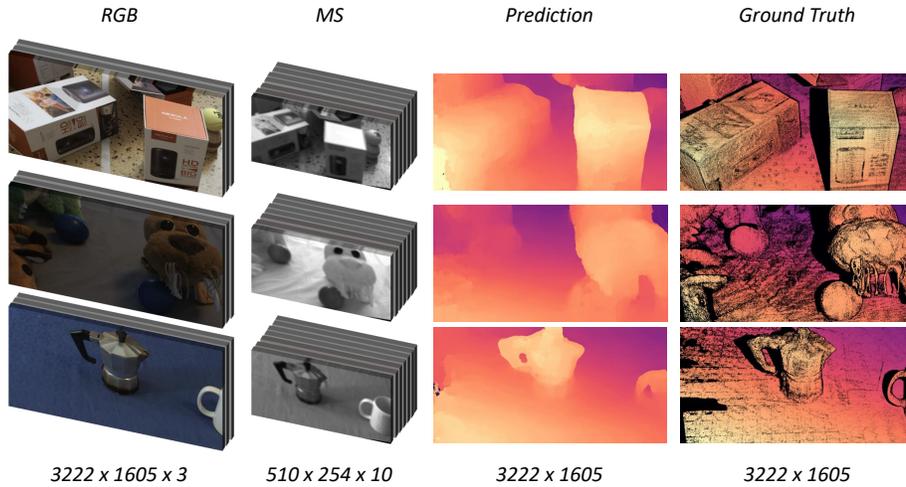


Figure 4. **Additional Qualitative Results of our Network.** From left to right: input RGB image, input multi-channels MS image, disparity predicted by our network, ground-truth.

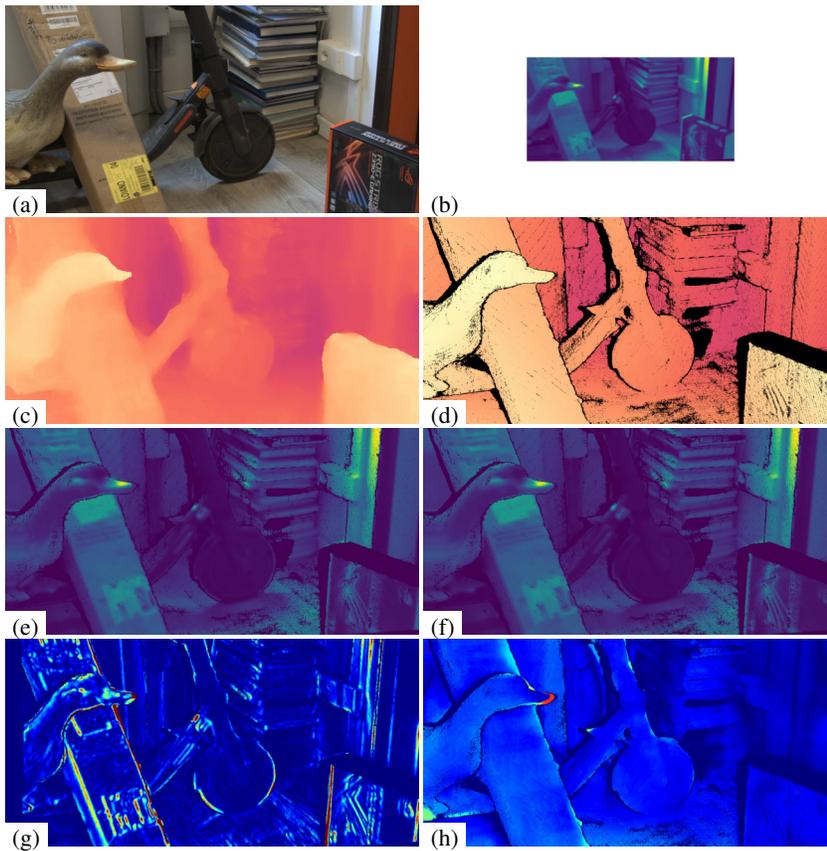


Figure 5. **Registration Results.** RGB image (a); MS image displayed as a channel average with color-map (b); predicted (c) and ground-truth (d) disparity maps; MS images warped based on the predicted (e) and ground-truth (f) disparities; absolute difference between registered images (g) and disparities (h).

$576 \times 960 \times 32$ computed by Φ_{θ} and on matching cost probabilities of dimension $92 \times 160 \times 32$ computed by Ψ_{θ} . It is worth noticing that, for the cost volume computation, we modify the original implementation of PSM such that the two feature

extractors for L and R do not share weights and accept 3 channels as input for the RGB branch whilst 10 channels for the multi-spectral branch.

GWC [3]: The GWC stereo backbone shares the same feature extractors as in PSM, while Ψ_θ incorporates group-wise correlation to build-up the cross-spectral cost volume. Specifically, we perform bilinear interpolation on convolutional features of size $576 \times 960 \times 332$ from Φ_θ and matching cost probabilities of dimension $92 \times 160 \times 32$ from Ψ_θ .

5.2. Continuous Output Representation

Our output representation follows the same implementation as proposed in [1], where two multi-layer perceptrons are in charge of estimating a categorical distribution over disparity values and a subpixel offset starting from interpolated features computed by the deep stereo backbone. More specifically, the number of neurons is $(D_\psi + F_\phi, 512, 256, 128, d_{max})$ for MLP_C while $(D_\psi + F_\phi + 1, 128, 64, 1)$ for MLP_R , where $D_\psi = 32$ and $F_\phi = 32$ in case of the PSM stereo backbone while $D_\psi = 32$ and $F_\phi = 332$ for GWC. In our experiments, we always use $d_{max} = 768$. Notice both multi-layer perceptrons rely on Sine activation functions, except the last layer where Softmax and Tanh activations are used for MLP_C and MLP_R , respectively.

References

- [1] Filippo Aleotti, Fabio Tosi, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Neural disparity refinement for arbitrary resolution stereo. In *International Conference on 3D Vision*, 2021. 3DV. 1, 6
- [2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018. 4
- [3] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019. 4, 6
- [4] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4