Thermal SuperPoint SLAM

Spencer Carmichael, Yizhou Chen, Matthew Saraceno, Xiaoxi Zhang

***Abstract*—Thermal cameras offer a number of advantages over RGB cameras for navigation, but they also present challenges that prevent direct functionality with existing visual Simultaneous Localization And Mapping (SLAM) algorithms. Recent work has shown that deep learning based features may be robust to the characteristics of thermal images. To test if such features can support SLAM, this project integrates a SuperPoint network trained on thermal images and a corresponding thermal SuperPoint visual vocabulary into ORB-SLAM2. The final integration does not succeed on the thermal sequences tested, however, we show positive intermediate results indicating the idea may have potential. We have made our code publicly available.**

I. INTRODUCTION

The use of thermal cameras for navigation offers a number of advantages over RGB cameras. Thermal cameras sensitive to the Long Wave Infrared (LWIR) spectrum are passive sensors that can operate in the absence of visible light and are robust to changes in illumination and visual obscurants such as fog, dust, and smoke [1] [2]. Thermal cameras also present a number of challenges however: they are lower contrast, experience greater motion blur, and often require large ($\sim$ 500 ms) image interruptions to combat noise [1] [2]. These challenges prevent handcrafted feature detectors and existing Visual Simultaneous Localization And Mapping (SLAM) algorithms from performing well on thermal images. There are two main variants of thermal cameras, cooled and uncooled, and the aforementioned issues are much less severe in cooled thermal cameras. However, cooled thermal cameras are prohibitively expensive for most applications and, to the authors' knowledge, are unseen in navigation literature.

Recently, a robust thermal-specific feature detection network called ThermalPoint was introduced as an alternative to handcrafted feature detectors [2]. ThermalPoint is a modified version of SuperPoint [3], which is a self-supervised framework for training feature detection and description networks. Inspired by the results of ThermalPoint, this project aims to create a monocular indirect thermal SLAM algorithm based on SuperPoint. Specificially, we train a thermal SuperPoint network and a corresponding thermal SuperPoint visual vocabulary and integrate them with ORB-SLAM2 [4]. Furthermore, we investigate the performance of our algorithm on both uncooled and cooled thermal cameras.

II. RELATED WORK

A. SuperPoint

SuperPoint is a self-supervised framework for training feature detection and description networks [3]. The architecture is composed of a shared encoder and two separate decoders: an interest point decoder and a descriptor decoder. The encoder includes three 2×2 non-overlapping max pooling layers that downsample the image into cells representing 8×8 pixel patches of the original image. The interest point decoder produces a pixel-wise interest point probability heatmap that is post-processed into keypoints via thresholding and non-maximum suppression (NMS). The descriptor decoder produces a descriptor for each pixel of the downsampled image and upsamples the result via bi-cubic interpolation. Each descriptor contains 256 32-bit floating point numbers and is normalized to 1. Computing the loss during training requires groundtruth keypoints. Instead of requiring real groundtruth, which would be infeasible to obtain, pseudo-groundtruth is generated. The full process is depicted in Figure 1. First, an interest point detection network called MagicPoint is trained on millions of synthetic images of simple geometric shapes with known keypoints. Then pseudo-groundtruth is generated on the training and validation sets using MagicPoint and a process called Homographic Adaptation. In Homographic Adaptation random homography transformations are applied to an image, each transformed image is passed through MagicPoint, and the final pseudo-groundtruth is found by projecting the outputs back into the original image and aggregating the results. Finally the full SuperPoint network is trained on the pseudo-groundtruth keypoints.

B. ThermalPoint

ThermalPoint is a modification of SuperPoint designed for thermal images [2]. ThermalPoint uses a different interest point detection loss to improve repeatability and uses bi-linear instead of bi-cubic interpolation when computing the descriptors in order to save computation time. In addition, fixed pattern noise that is characteristic of thermal cameras is added to the training dataset prior to training in an attempt to make it more robust to this type of noise. Also, SIFT features are used to augment the MagicPoint-generated pseudo-groundtruth. In the ThermalPoint paper, the descriptor network is discarded after training and feature correspondences are instead obtained through IMU-aided feature tracking. This technique is motivated by real-time limitations. The paper compares ThermalPoint against SuperPoint and multiple handcrafted features in regard to their performance on thermal images. ThermalPoint and SuperPoint are similarly robust to large photometric changes, demonstrate similar repeatability at detecting keypoints across randomly applied homographies, and significantly outperform the handcrafted features in general. The one area in which ThermalPoint greatly outperforms SuperPoint is robustness to fixed pattern noise: ThermalPoint is capable of rejecting areas with fixed pattern noise while SuperPoint finds keypoints within these areas.

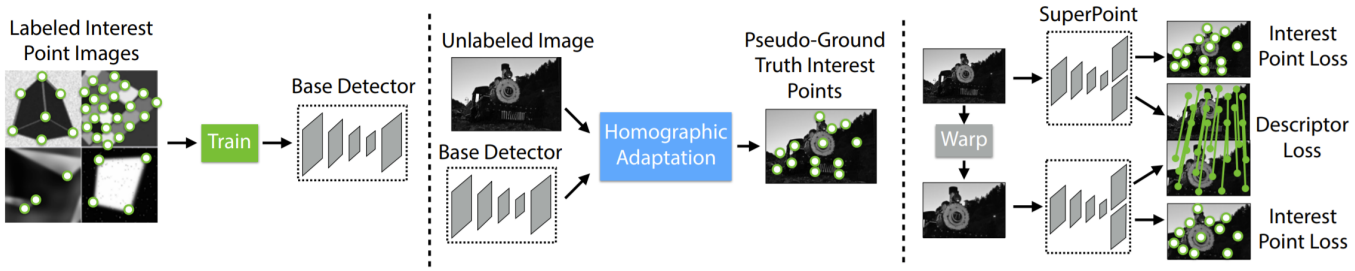

Fig. 1. The stages of training a SuperPoint network. (Left) training MagicPoint with synthetic images (Middle) generating pseudo-groundtruth via Homographic Adaptation (Right) training SuperPoint with the pseudo-groundtruth. Diagram from [3].

C. DeepFEPE

Deep learning-based Feature Extraction and Pose Estimation (DeepFEPE) is a two-frame pose estimation pipeline including learning-based feature extraction and pose estimation modules [5]. SuperPoint is used as the feature extraction module with some small modifications: a sparse descriptor loss is used to speed up training and a 2D softmax function is applied as an additional interest point post-processing step to yield subpixel keypoints.

D. DBoW2

DBoW2 is a library for generating a visual vocabulary [6]. A visual vocabulary is a discretization of descriptor space into visual words that allows for efficient similarity scoring between images represented as bags-of-words (BoWs). The DBoW2 vocabulary is generated using a set of descriptors by recursively applying k-means clustering up to L times to generate a tree where k is the branch factor and L is the depth level. After the process is complete the W leaves of the tree represent the visual words. Each word is defined by the descriptor centroid of the corresponding cluster and a frequency based weight. The weight decreases the impact of words that were common in the training set under the assumption they are less discriminative. To represent an image as a BoW a set of descriptors is found in the image and then each of the descriptors traverses the tree down to a visual word by selecting the closest node at each level as determined by the distance metric used for that descriptor type. The resulting BoW is a vector counting the number of descriptors to land on each word. The L1-score is computed to measure the similarity between two BoW vectors v_1 and v_2:

$$s(v_1, v_2) = 1 - \frac{1}{2}\left|\frac{v_1}{|v_1|} - \frac{v_2}{|v_2|}\right| \tag{1}$$

E. ORB-SLAM2

ORB-SLAM is an indirect monocular SLAM algorithm that incorporates many individual modules and uses ORB features throughout [7]. A depiction of the full system is shown in Figure 2. Notably the system includes automatic initialization, local bundle adjustment, relocalization, and loop closure. The automatic initialization is performed by estimating the relative pose between two frames by computing both a homography and fundamental matrix and heuristically selecting one. If

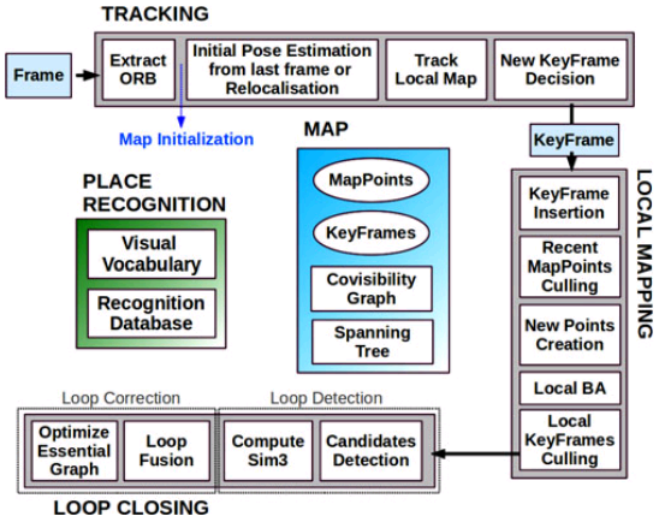

Fig. 2. Depiction of the modules that comprise ORB-SLAM and their interrelationships. Diagram from [7].

the scene is planar or nearly so it can be explained by a homography, while the fundamental matrix can model both planar and non-planar scenes. Despite the generality of the fundamental matrix, the homography is favored during initialization because the fundamental matrix estimation is not well constrained when there is low parallax between the two frames or when the scene is planar. An ORB DBoW2 vocabulary is used in ORB-SLAM for relocalization, loop closure, and in some instances, to speed up feature matching. The main contribution of ORB-SLAM2 is the addition of support for stereo and RGB-D cameras [4]. From the monocular standpoint the main benefit is an improved codebase.

F. SuperPoint SLAM

SuperPoint SLAM is an existing integration of SuperPoint features with ORB-SLAM2 that was used in a study comparing the effectiveness of using ORB, SuperPoint, and GCNv2 within RGB monocular ORB-SLAM2 [8] [9] [10]. The methods were evaluated on the KITTI dataset and the results of SuperPoint SLAM were mixed. In terms of the Root Mean Square Error (RMSE) of the trajectory, SuperPoint SLAM was similar to or better than standard ORB-SLAM2

on some KITTI sequences, but it completely failed to track on others.

III. METHODOLOGY

Most thermal cameras output 16-bit images. We make the assumption that these 16-bit images are not compatible with the synthetic images typically used to train MagicPoint. In this case, the resulting MagicPoint network will not be able to identify any pseudo-groundtruth keypoints in the 16-bit images. Rather than modify the synthetic imagery generation we choose to convert the 16-bit images to 8-bit images both for training and inference.

The values of the raw 16-bit images represent temperatures spanning a wide possible range. The values present in typical scenes only occupy a very small portion of this range and there is a low amount of contrast even within the occupied range. If the 16-bit image is directly mapped to an 8-bit image nearly every pixel is mapped to same value. Therefore, some method is needed to distribute the values in the 16-bit image over a wider range prior to mapping the values to an 8-bit image. Figure 3 compares multiple methods. The image on the far left shows the result of directly mapping the 16-bit image to an 8-bit image and the image looks entirely black. The next image shown is the result of applying min-max normalization which rescales the original pixel value distribution to the full range before converting it to 8-bit. The scene is visible in the normalized image, but the histogram is divided into two tight clusters and the resulting image is very low contrast. This lack of contrast could also pose an issue for generating pseudo-groundtruth and this motivates the use of histogram equalization techniques.

Histogram equalization increases contrast by transforming the image such that its histogram is approximately uniform. As seen in the figure, when histogram equalization is applied globally it tends to cause local saturation. The solution is adaptive histogram equalization (AHE) which divides the image into tiles, applies histogram equalization within each tile individually, and then uses bi-linear interpolation to smooth the edges between tiles. Contrast limited adaptive histogram equalization (CLAHE) additionally reduces noise amplification by clipping and redistributing histogram bins above a certain contrast threshold. The result of applying CLAHE is also shown in the figure. We use CLAHE over the other methods as it produces contrast across the entire image. We use an 8×8 tile grid size to align the tiles with those in the SuperPoint network and we qualitatively found a clip limit of 100 to work best.

Each of the contrast enhancement methods just discussed will apply an inconsistent transformation to the same area of the scene as the objects in the frame move and the pixel value statistics change. This is especially true of CLAHE as the transformation is applied in a small grid that can change significantly frame to frame. The result is artificial brightness changes in the image. This could pose a problem for feature detection algorithms, however, random brightness changes are applied to images during the training of the SuperPoint network and so we assume that the network will be robust to this issue.

There is no publicly available code for training a Thermal-Point network. Given that the differences between Thermal-Point and SuperPoint are relatively minor, both algorithmically and terms of performance, we assume SuperPoint is a representative substitute. After preprocessing the 16-bit images into 8-bit images we proceed with training the thermal SuperPoint network as described in Section II-A using the DeepFEPE SuperPoint implementation [5].

From a theoretical standpoint the subsequent integration of the trained thermal SuperPoint network with ORB-SLAM2 is straightforward. While ORB is in the name and is used throughout the system, every module of ORB-SLAM2 is fundamentally feature agnostic. To perform the integration a thermal SuperPoint DBoW2 vocabulary must be generated and swapped in for the original ORB DBoW2 vocabulary. Then the ORB-SLAM2 front-end must be adjusted to compute Super-Point features and the rest of the system needs to be slightly modified to account for the difference in the descriptors. In particular, the ORB descriptor is binary while the SuperPoint descriptor is not, therefore the descriptor distance metric must be changed from the Hamming distance to the L2 distance.

IV. EXPERIMENTS

A. Implementation & Datasets

We have publicly released the code for the implementation described in this section[1]. See our repository for details that could not be covered here. All steps were performed on a single machine with a NVIDIA 1080 Ti GPU, Intel Core i7-7700K CPU, and 32 GB of RAM.

Training the Thermal SuperPoint Network: We used the publicly available DeepFEPE SuperPoint implementation to train our thermal SuperPoint network [5]. The repository contains a pretrained MagicPoint network. We trained on the FLIR ADAS thermal dataset containing images taken from a car with a FLIR Tau2 uncooled thermal camera [11]. We used the recommended training and validation sets containing 8,862 and 1,366 images respectively. We resized the images to 256×320 and, as described in Section III, CLAHE was applied to the images prior to training using a climit of 100 and a tile grid size of 8×8. The pretrained MagicPoint network was used to generate pseudo-groundtruth for the training and validation sets and the thermal SuperPoint network was trained over 46,000 iterations. Both pseudo-groundtruth generation and SuperPoint training took around five hours to complete. See the configuration files located in our repository for the full training details[1].

Training the Thermal SuperPoint Vocabulary: We added support for SuperPoint features to the DBoW2 repository and generated a vocabulary using the 8,862 images in the FLIR ADAS thermal dataset [6] [11]. As in the original DBoW2 paper we used 300 descriptors per image for total of $\sim 2.7M$ words in total [6]. We added an iteration limit of 100 to

[1]https://github.com/specarmi/Thermal_SuperPoint_SLAM

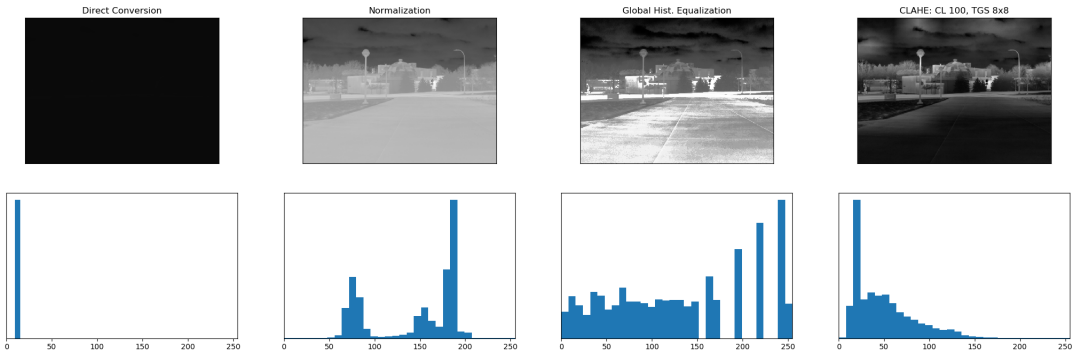

Fig. 3. Multiple methods tested for converting 16-bit thermal images to 8-bit. (Top) the 8-bit image resulting from the method (Bottom) a histogram of the values in the 8-bit image.

each application of k-means. Previously, DBoW2 would only progress to the next node if all the descriptors remained in the same clusters for two iterations. In our experience, this would frequently not occur and instead the percentage of descriptors switching clusters each iteration would oscillate. The vocabulary was created with a branching factor $k = 10$, depth level $L = 5$, TF-IDF weighting type, and L1 scoring type. The generation process took around three hours to complete.

Integrating the Network and Vocabulary with ORB-SLAM2: We built off of the repository for the existing integration between SuperPoint and ORB-SLAM2: SuperPoint SLAM [8]. The original SuperPoint SLAM supports the online computation of SuperPoint features using the pretrained network provided by the original SuperPoint authors [3]. The DeepFEPE SuperPoint implementation we used employs different layers in the network and enhanced subpixel post-processing and is therefore not directly compatible with the SuperPoint SLAM algorithm. Properly adding support would require significant development time. Due to time constraints, we instead work around this issue by precomputing the SuperPoint features and later importing them. Note that to simplify the implementation we also only compute features at full scale. This is unlike the original SuperPoint SLAM and ORB-SLAM2 that compute features at multiple spatial scales. Those algorithms also use multiple methods to ensure a good distribution of keypoints over the image, while we only use NMS.

B. Evaluation

See some of our results in motion in our video[2].

Feature Matching: As an intermediate evaluation of our thermal SuperPoint network we compared its feature tracking performance against ORB, SIFT, and a pretrained RGB SuperPoint network that was included in the DeepFEPE SuperPoint repository and was trained on $\sim 80k$ images in the MS-COCO dataset [5] [12]. The ORB and SIFT implementations are from

[2]https://youtu.be/TwUVYOlQn44

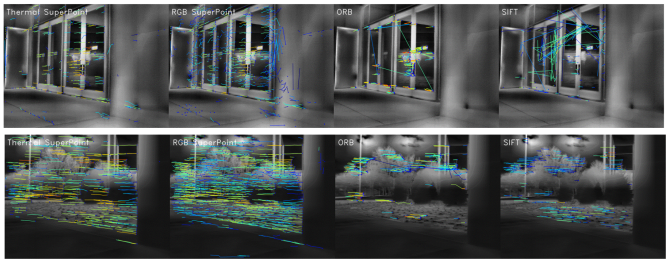

Fig. 4. Feature tracking on thermal images. From left to right: Thermal SuperPoint, RGB SuperPoint, ORB, and SIFT. (Top) a difficult scene in which ORB and SIFT find many mismatches (Bottom) and easier scene where there are fewer mismatches but the SuperPoint methods find many more good matches.

Fig. 5. Progressively dissimilar frames of a thermal video.

OpenCV and were used with default settings. Figure 4 shows the tracking performance on thermal images from the UM Ford Center for Autonomous Vehicles (FCAV) collected using a FLIR A655sc uncooled thermal camera [13]. The features are tracked using brute force feature matching with cross-check. The color indicates the average descriptor distance over the track and it increases from red to blue. The figure shows that both SuperPoint networks greatly outperform the handcrafted methods: they find more matches and fewer mismatches. Interestingly the RGB SuperPoint network appears to outperform our thermal SuperPoint network by finding more matches. This may be because we trained on a relatively small and homogeneous dataset while the RGB SuperPoint network was trained on the large MS-COCO dataset.

Image Similarity Scoring: We evaluated our thermal SuperPoint vocabulary by comparing it against the ORB-SLAM2 ORB vocabulary and the SuperPoint SLAM RGB SuperPoint

TABLE I
VOCABULARY L1 IMAGE SIMILARITY SCORING RESULTS. SCORES RANGE FROM 0 TO 1; A HIGHER SCORE MEANS MORE SIMILAR.

Frames	Thermal SP Vocab & Thermal SP Features	RGB SP Vocab & Thermal SP Features	RGB SP Vocab & RGB SP Features	Thermal SP Vocab & RGB SP Features	ORB Vocab & ORB Features
1 vs. 1	1	1	1	1	1
1 vs. 2	0.555669	0.25812	0.358499	0.582337	0.143056
1 vs. 15	0.269333	0.0598247	0.191989	0.4141	0.0114565
1 vs. 40	0.229502	0.0300138	0.180794	0.427281	0.00612358
1 vs. 4244	0.13876	0.0176324	0.180387	0.434895	0.0016234

TABLE II
VOCABULARY STATISTICS.

Vocabulary	Branching factor k	Depth Level L	Number of Words
Thermal SuperPoint	10	5	17.5k
RGB SuperPoint	10	5	99.8k
ORB	10	6	1M

vocabulary that were both trained on $\sim 10k$ images from the Bovisa 2008-09-01 dataset [4] [8] [14] [15]. We tested them by using them to score the similarity of progressively dissimilar images. All combinations of SuperPoint features and SuperPoint vocabularies were tested. The images, shown in Figure 5, are from a thermal video in the FLIR ADAS thermal dataset (outside of the training set) [11]. The scores, as shown in Table I, indicate that in almost every case the similarity is ranked correctly, but the ORB vocabulary is more discriminative. This may once again be due to our limited training dataset. The vocabulary statistics listed in Table II further indicate that our dataset may be insufficient. The other two vocabularies were trained on a similar number of images and descriptors but have a much larger number of words. During the vocabulary generation process the recursive k-means process will stop before it has been applied L times only if there is only one word in the node. The low number of words in our vocabulary therefore implies that many of the words only contain one descriptor in the training set. This combined with the fact that we had to add an iteration limit to k-means seems to imply our thermal vocabulary tree is imbalanced. Whether this issue is caused by our dataset alone or thermal imagery more generally is unclear.

SLAM Performance: As a baseline we tested the performance of standard ORB-SLAM2 on KITTI sequence 03 [16]. The result is shown in Figure 6. Subsequently, to validate our integration of SuperPoint with ORB-SLAM2 we ran SuperPoint SLAM with the pretrained RGB SuperPoint network on KITTI 03. The result, shown in Figure 7, demonstrates that our integration is functional. We also ran standard ORB-SLAM2 on a sequence from FCAV taken with a FLIR A655sc uncooled thermal camera and another FCAV sequence taken with a FLIR X8500sc cooled thermal camera; it was not able to initialize on either. Finally we tested our thermal SuperPoint SLAM on the same two thermal camera sequences. Thermal SuperPoint SLAM failed to initialize on the uncooled camera, but was able to initialize briefly on the cooled camera before

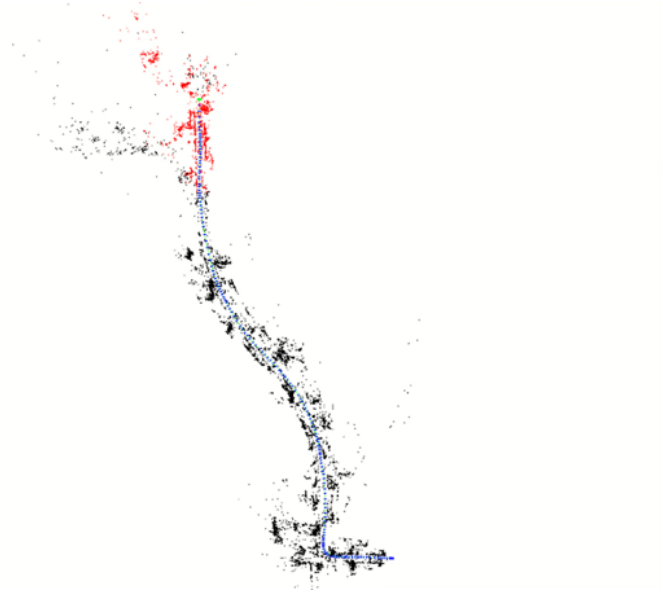

Fig. 6. Estimated trajectory from standard ORB-SLAM2 run on KITTI sequence 03.

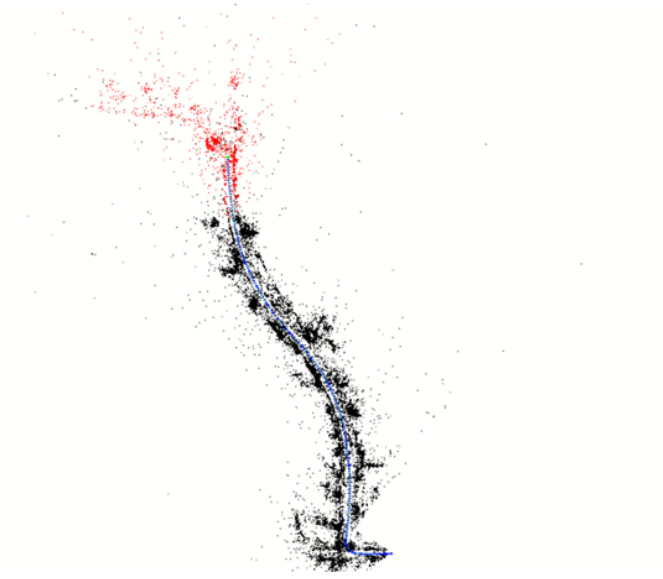

Fig. 7. Estimated trajectory from RGB SuperPoint SLAM run on KITTI sequence 03.

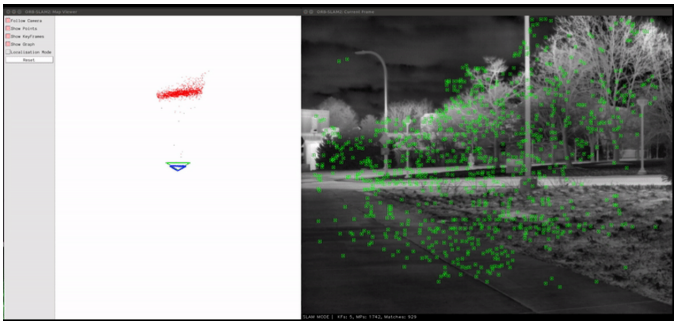

Fig. 8. Initial point cloud from thermal SuperPoint SLAM initialization on the FLIR X8500sc sequence.

losing tracking a short time later into the sequence. The initial point cloud from that run is shown in Figure 8. Note that the points do not align with the scene and seem to lie on a vertical plane. This seems to imply that the automatic initialization method falsely selected a homography over a fundamental matrix. Whether the fundamental matrix would have provided a good initialization here is unclear.

V. CONCLUSIONS & FUTURE WORK

We have shown that our thermal SuperPoint network and the pretrained RGB SuperPoint network appear better than ORB and SIFT at tracking features in thermal image sequences. We have also shown that our thermal SuperPoint vocabulary can correctly rank the similarity of images, however the ORB vocabulary is more discriminative. The integration of SuperPoint and ORBSLAM2 we put together is functional but still failed on thermal images. The performance may be able to be improved with keypoint generation over multiple spatial scales, better parameter tuning, a larger and more diverse training dataset or an improved contrast enhancement method. Additionally, it may be possible to leverage the larger amount of information contained in the raw 16-bit images by training and running directly on them.

REFERENCES

[1] M. S. Ramanagopal, Z. Zhang, R. Vasudevan, and M. Johnson-Roberson, "Pixel-Wise Motion Deblurring of Thermal Videos," *arXiv:2006.04973 [cs]*, Jun. 2020.

[2] S. Zhao, P. Wang, H. Zhang, Z. Fang, and S. Scherer, "TP-TIO: A Robust Thermal-Inertial Odometry with Deep ThermalPoint," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 4505–4512.

[3] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 337–33 712.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] Y. Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4950–4957.

[6] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] C. Deng, K. Qiu, R. Xiong, and C. Zhou, "Comparative study of deep learning based features in slam," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2019, pp. 250–254.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[10] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "Gcnv2: Efficient correspondence prediction for real-time slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3512, 2019.

[11] "Flir thermal dataset for algorithm training," Jul. 2018. [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[13] "UM Ford Center for Autonomous Vehicles (FCAV)," Apr. 2021. [Online]. Available: https://fcav.engin.umich.edu/

[14] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based slam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 846–853.

[15] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos, "Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets," in *In proceedings of IROS*, vol. 6, 2006, p. 93.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231–1237, Sep. 2013.