Virtual sensors determined through machine learning

Yumi Iwashita and Adrian Stoica
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA, USA
Email: Yumi.Iwashita, Adrian.Stoica@jpl.nasa.gov

Kazuto Nakashima, Ryo Kurazume
Kyushu University
Fukuoka, Japan
Email: k_nakashima, kurazume@irvs.ait.kyushu-u.ac.jp

***Abstract*—We propose a method that increases the capability of a conventional sensor/instrument, transforming it into an enhanced virtual sensor/instrument. This paper focuses on a virtual thermal IR sensor based on a conventional visual (RGB) sensor. The estimation of thermal IR images can enhance the ability of terrain classification, which is crucial for autonomous navigation of rovers. The estimate in IR from visual band has inherent limitations, as these are different bands, yet correlations between visual/RGB and thermal IR images exist, as different terrains, which visually may appear different, also have different thermal inertia. We developed a deep learning-based algorithm that estimates thermal IR images from RGB images of terrains, providing the feasibility of the idea with average 1.21 error [degree].**

I. Introduction

The Mars Science Laboratory (MSL) rover [1] has several RGB cameras, but no thermal infrared (IR) cameras. RGB cameras respond to wavelength from about 390 to 700 $[nm]$ and thermal cameras respond to different wavelength, such as 7-14 $[\mu m]$ for long-wave IR. MSL has a Ground Temperature Sensor (GTS), but GTS can get only one pixel that covers 100 m^2 area. Sensing/imaging in different bands, to capture high resolution images, would allow imagery to be used both for science objectives [2] but also for high-level autonomy, e.g. in slip prediction [3].

We propose a method that increases the capability of a conventional sensor/instrument, transforming it into an enhanced virtual sensor/instrument. This paper focuses a method to estimate thermal IR images from RGB images. Theoretically it is not possible to deterministically obtain complete thermal information from RGB information. As Fig. 1 suggests, however, correlations between RGB and thermal IR images may exist, since terrains that are different (and look different in visual) also have different thermal inertia (here, thermal inertia is understood as being a physical property of a material that represents its resistance to changes in temperature). This would allow us to take an RGB visual camera input and derive from it the simulated (estimated) output of a virtual thermal IR camera.

We present a method to estimate thermal information from RGB information. Full estimation is impossible, as these are different bands, yet correlations between visual/RGB and thermal IR images exist, as terrains have different thermal inertia.

To see the feasibility of the proposed idea, we propose a deep learning-based algorithm to estimate thermal IR images from RGB images of terrains. The proposed method is based on U-Net [4], which is popularly used in a medical image segmentation [5] and also was used by the winner of a satellite image segmentation competition (Kaggle competition [6]).

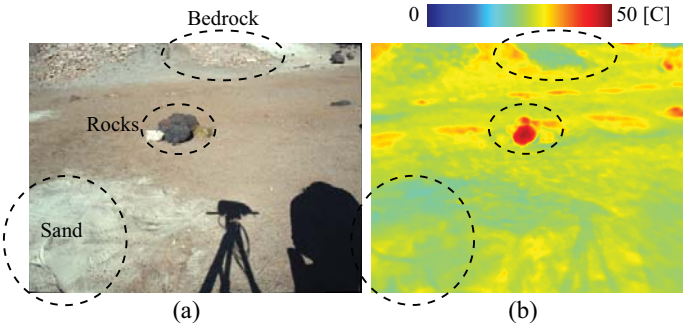

Fig. 1. (a) visual and (b) thermal images at 1 pm on Nov 17th. Each terrain type shows different surface temperature due to different thermal inertia. The IR image is originally grayscale image, but it is colorized for visualization purpose.

A. Previous work

To the best of our knowledge, there is no prior work estimating thermal IR images from RGB images. Instead, there are many papers for colorizing gray scale images [7] [8] [9]. In general these methods require to estimate chrominance, since the luminance is given in the grayscale images. Iizuka et al. proposed a deep convolutional neural networks (deep CNN) to directly estimate chrominance values in gray-scale images [11]. Larsson et al. [12] and Zhang et al. [13] initialized their networks with pre-trained networks. Limmer et al. proposed a CNN-based method to colorize near IR images, which requires to estimate chrominance and luminance [10].

Generally deep learning methods require huge dataset to train its parameters. In [10] almost 38,495 image pairs are used. Thus in the case that users do not have enough data to train the network from scratch, pre-trained parameters with public dataset such as ImageNet [14] are used. However, since generally public dataset includes images with huge inter-class variations, such as cars, humans, balls, etc, the pre-trained parameters do not efficiently describe features of terrain types,

where inter-class variations are much smaller than the public dataset. On the other hand, the advantage of U-Net, which we use in this paper, is that it can train the network with much smaller datasets.

II. Methodology

In this section we first explain U-Net, and then the methodology used to synthesize IR images from RGB images.

A. U-Net

Overall the architecture of the U-Net consists of a contracting path (left) and an expansive path (right). Each path has repeated units. The unit on the contracting path (contracting unit), as shown with light blue rectangles, consists of two 3×3 convolutions, each followed by a rectified linear unit (ReLU) and 2×2 max-pooling. There are two different units on the expansive path. The first one (expansive unit 1) as shown with red rectangles consists of two 3×3 convolutions, each followed by ReLU, 2×2 deconvolution, and concatenation of outputs from both deconvolution layer and convolution layer from the contracting path. Another one (expansive unit 2) is the last unit of the expansive path as shown with an orange rectangle has two 3×3 convolutions, each followed by a rectified linear unit (ReLU). Here, in the expansive unit 1, bilinear up-sample is applied to output of the convolution layer from the contracting path. This concatenation layer is one of key ideas in U-Net, it enables the user to train the network with a small amount of data. Input images into U-Net have 3 channels (RGB), and at the final layer a 1×1 convolution is applied to map 64 channel information at each pixel to the number of classes.

The loss function $\mathcal{L}_{CE}$ of all architectures (CE: Cross Entropy) is defined as a pixel-wise soft-max over the final map, followed by the cross-entropy loss function, as define as follows.

$$\mathcal{L}_{CE} = -\frac{1}{|S|} \sum_{i \in S} \sum_{j=1}^{N} y_{ij} \log p_{ij}, \tag{1}$$

where N, $|S|$, y_{ij}, p_{ij} are the number of terrain classes (e.g. sand, soil, rocks, etc), the total number of pixels over images S, groud-truth distribution at each pixel, and outputted probability distribution at each pixel, respectively. The loss function is minimized by a stochastic gradient descent method.

B. Virtual thermal camera

To synthesize thermal IR images from RGB images, we replace the output annotation in Fig. 2 with thermal IR image. We use a Mean Squared Error $\mathcal{L}_{MSE}$ as a loss function, which is defined as

$$\mathcal{L}_{MSE} = \frac{1}{|S| \times C} \sum_{i \in S} \sum_{j=1}^{C} (a_{ij} - b_{ij})^2, \tag{2}$$

where C is the number of channels, and a_{ij} is b_{ij} are thermal value at each pixel (i, j) of a ground-truth thermal image a and an output thermal image b.

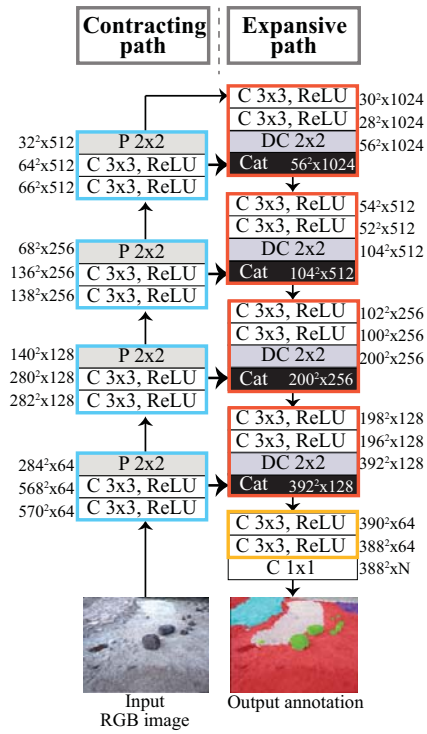

Fig. 2. U-Net architecture. "Cat", "C", "ReLU", "P", and "DC" mean "concatenate", "convolution", "rectified linear unit", "pooling", and "deconvolution", respectively. Relatively thick arrows between "Cat" and "C" include "bilinear up-sample". "N" at the final layer show the number of classes. Light blue rectangles show units of the contracting path (contracting units). Red and orange rectangles show two different units of expansive paths (expansive unit 1 and 2).

Fig. 3. Camera settings.

III. Experiments

In this section, we first present a dataset that includes visible and thermal images, followed by experimental results with the dataset.

A. Dataset of visible and thermal images

To collect images, we used a RGB camera (FLIR Grasshopper 5M) and a thermal camera (FLIR AX65) as shown in Fig. 3. We collected images at 5 pm; there are 52 images, obtained by changing the position of the cameras. Figure 4

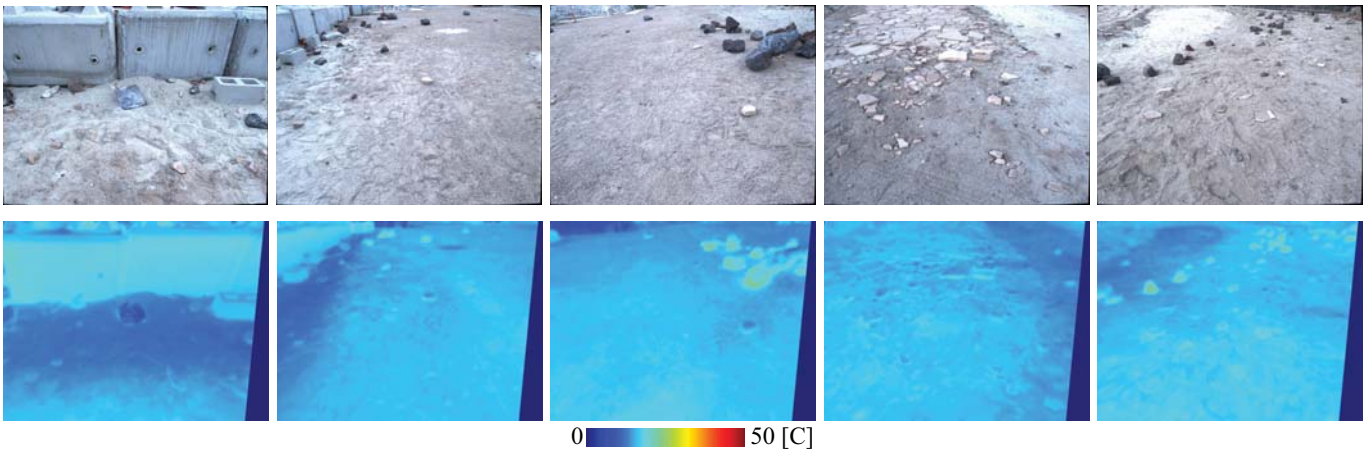

Fig. 4. Examples of RGB (top) and IR (bottom) image pairs.

shows examples of captured image pairs. Areas with sand tend to show lower temperature, due to sand's lower thermal inertia. On the other, areas with rocks show higher temperature, as rocks have higher thermal inertia. The visible and thermal images are taken by different cameras, so a registration process between cameras is necessary. After we removed distortion with estimated camera inner parameters, we applied an affine transformation with estimated homography matrix as shown in Fig. 5. Here, the camera parameters and the homography matrix were estimated with a calibration board with reflectors. Since the thermal images are aligned into the visible images, a right area in each thermal image does not have thermal information (black) (Fig. 5 (b)).

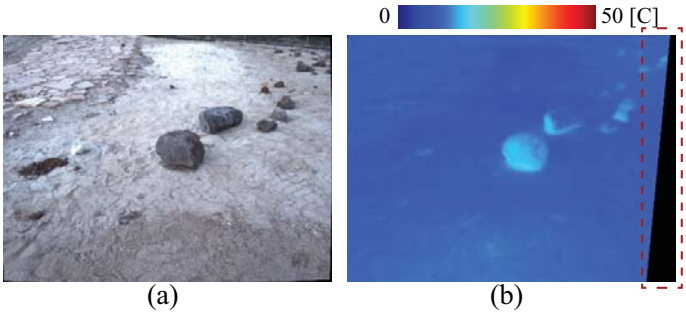

Fig. 5. (a) An example of undistorted RGB images, (b) a undistorted and transformed IR image corresponding (a). Right area of (b) has no data (black).

B. Results

We randomly separated the RGB and IR pair images as follows: 50% for training, 25% for evaluation to determine parameters, and the rest 25% for test. We categorize the ground area into 6 terrain types: soil, sand, rocks, bedrocks, rocky terrain, and ballast. The data size of each terrain type is not balanced, so we introduce weights to the MSE loss. Here, the assumption is that each terrain has unique temperature on IR images and we ignore other factors that change temperature, such as shades, due to the fact that the amount of shaded area is small in the dataset. A weight of each terrain type is defined as the square root of the ratio of number of pixels in the training dataset. We set weight at each pixel based on its terrain type, which we manually determined in advance.

In the first experiment, we simply use the MSE loss without weights. Figures 6 (a) and (b) shows examples of captured RGB images and its corresponding thermal IR images in test dataset. Figure 6 (c) shows estimated thermal IR images from RGB images. Mean and standard deviation of temperature difference are 2.3 degree and 3.8 degree (Table I (a)). From these results, lower and higher temperature areas are not estimated well.

In the next experiment we used the MSE loss with weights. Figure 6 (d) shows estimated thermal IR images from RGB images. Mean and standard deviation of temperature difference are 1.8 degree and 3.7 degree (Table I (b)). Compared with the results without weights (Fig. 6 (c)), Fig. 6 (d) shows better performance and we can see a feasibility of estimating thermal IR images from RGB images.

TABLE I
MEAN SQUARE ERROR (MSE) AND STANDARD DEVIATION (STDDEV) OF ESTIMATED THERMAL IR IMAGES BY (A) U-NET WITHOUT WEIGHTS AND (B) U-NET WITH WEIGHTS.

	MSE	stddev
(a) U-Net without weights	2.3	3.8
(b) U-Net with weights	**1.8**	**3.7**

IV. CONCLUSION

In this paper we proposed the idea of estimating thermal IR images from RGB images, and we showed its feasibility based on U-Net. There are many parameters to determine temperature of terrain surface, such as thermal inertia, direction to the Sun, geological condition, etc. Thus the future work will include these parameters in the model.

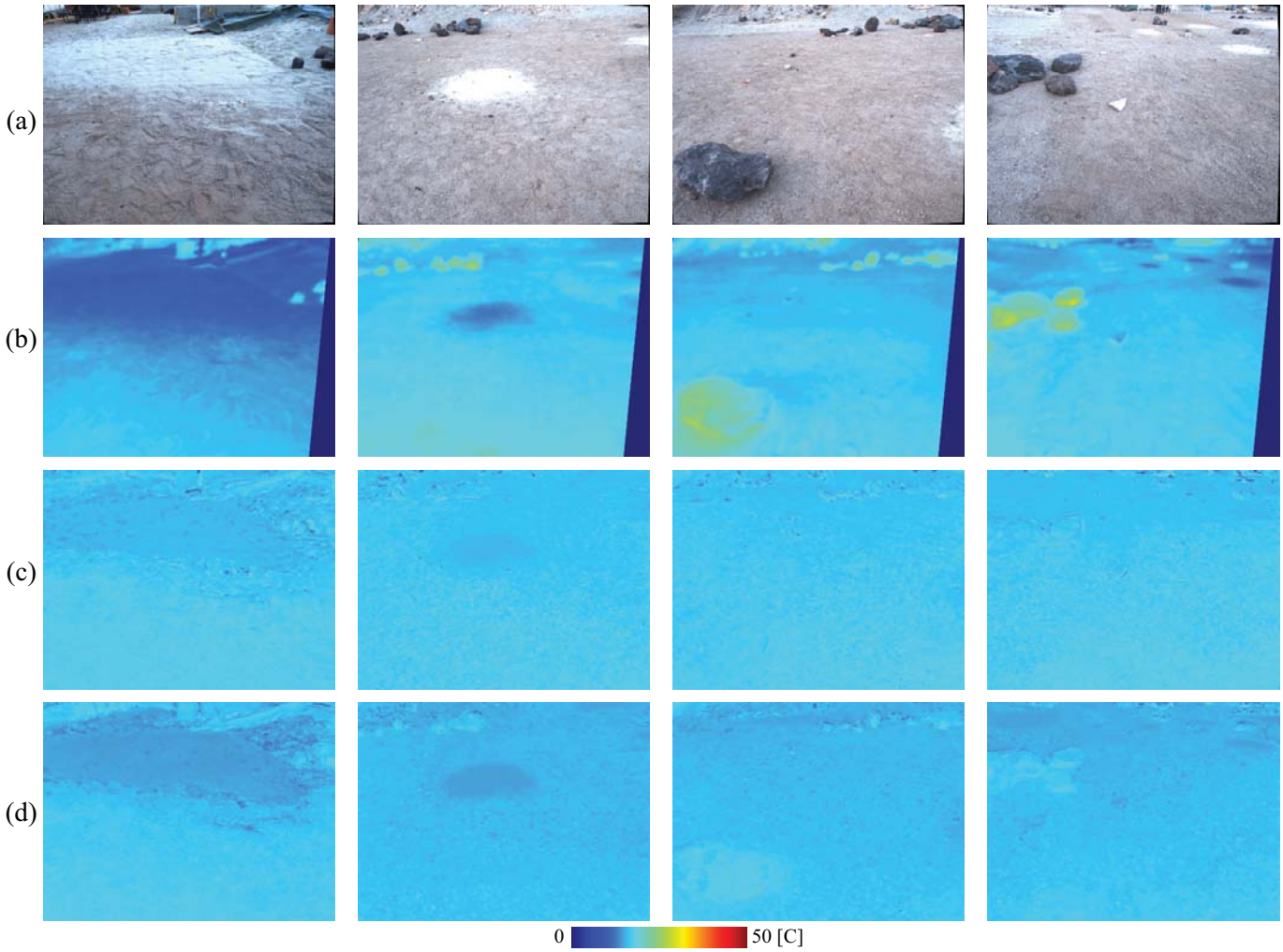

Fig. 6. (a) Examples of RGB images in test dataset, (b) ground truth thermal IR images corresponding to (a), (c) estimated thermal IR images WITHOUT weights, and (d) estimated thermal IR images WITH weights. Thermal IR images are colorized for visualization purpose.

ACKNOWLEDGMENT

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Mars Science Laboratory (MSL), https://mars.nasa.gov/msl/
[2] P.R. Christensen, G. Mehall, S.H. Silverman, S. Anwar, G. Cannon, N. Gorelick, R. Kheen, T. Tourville, D. Bates, S. Ferry, T. Fortuna, J. Jeffryes, W. O'Donnell, R. Peralta, T. Wolverton, D. Blaney, R. Denise, J. Rademacher, R.V. Morris, and S. Squyres, The Miniature Thermal Emission Spectrometer for the Mars Exploration Rovers, J. Geophysical Research 108, no. E12 (2003): 8064. 20 .
[3] C. Cunningham, I. Nesnas, and W. Whittaker, Improving Slip Prediction on Mars Using Thermal Inertia Measurements, RSS2017.
[4] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015.
[5] O. Cicek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, MICCAI 2016.
[6] Dstl Satellite Imagery Feature Detection, https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection
[7] Z. Cheng, Q. Yang, and B. Sheng, Deep colorization, Proc. International Conference on Computer Vision ,2015.
[8] G. Patterson and J. Hays, Sun attribute database: Discovering, annotating, and recognizing scene attributes, Proc. Conference on Computer Vision and Pattern Recognition, 2012.
[9] E. Tola, V. Lepetit, and P. Fua, A fast local descriptor for dense matching, Proc. Conference on Computer Vision and Pattern Recognition, 2008.
[10] M. Limmer, and H. Lensch Infrared Colorization Using Deep Convolutional Neural Networks, IEEE Int. Conf. on Machine Learning and Applications (ICMLA), 2016.
[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa, Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, Proc. ACM SIGGRAPH, vol. 35, no. 4, 2016.
[12] G. Larsson, M. Maire, and G. Shakhnarovich, Learning representations for automatic colorization, arXiv:1603.06668, 2016.
[13] R. Zhang, P. Isola, and A. A. Efros, Colorful image colorization, arXiv:1603.08511, 2016.
[14] J. Den, W. Dong, R. Socher, and F. Li, ImageNet: a Large-Scale Hierarchical Image Database, CVPR 2009.