Generating Visible Spectrum Images from Thermal Infrared

Amanda Berg, Jörgen Ahlberg and Michael Felsberg

Conference article

Cite this conference article as:

ISBN: 9781538661000 (online), 9781538661017 Print-on-demand

DOI: https://doi.org/10.1109/CVPRW.2018.00159

Copyright: Proceedings 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops CVPRW 2018

The self-archived postprint version of this conference article is available at Linköping University Institutional Repository (DiVA):
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-149429
Generating Visible Spectrum Images from Thermal Infrared

Amanda Berg¹,², Jörgen Ahlberg¹,², Michael Felsberg²
¹Termisk Systemteknik AB, Diskettgatan 11 B, 583 35 Linköping, Sweden
²Computer Vision Laboratory, Dept. EE, Linköping University, 581 83 Linköping, Sweden
{amanda.,jorgen.ahl}berg@termisk.se, {amanda.,jorgen.ahl,michael.fels}berg@liu.se

Abstract

Transformation of thermal infrared (TIR) images into visual, i.e. perceptually realistic color (RGB) images, is a challenging problem. TIR cameras have the ability to see in scenarios where vision is severely impaired, for example in total darkness or fog, and they are commonly used, e.g., for surveillance and automotive applications. However, interpretation of TIR images is difficult, especially for untrained operators. Enhancing the TIR image display by transforming it into a plausible, visual, perceptually realistic RGB image presumably facilitates interpretation. Existing grayscale to RGB, so called, colorization methods cannot be applied to TIR images directly since those methods only estimate the chrominance and not the luminance.

In the absence of conventional colorization methods, we propose two fully automatic TIR to visual color image transformation methods, a two-step and an integrated approach, based on Convolutional Neural Networks. The methods require neither pre- nor postprocessing, do not require any user input, and are robust to image pair misalignments. We show that the methods do indeed produce perceptually realistic results on publicly available data, which is assessed both qualitatively and quantitatively.

1. Introduction

This paper addresses the problem of transforming thermal infrared (TIR) images to visual, i.e., perceptually realistic RGB images. The process of adding color to black-and-white photography or visual grayscale images is commonly known as colorization. Colorization of grayscale visual images is an ambiguous, yet well-researched problem [3, 7, 14, 18, 25, 32, 39]. It is ambiguous in the sense that a grayscale intensity value can correspond to multiple color values. Despite this ambiguity, recent methods show impressive results, see e.g. the Colorful Image Colorization¹ [39] and Let there be color² [18] demos.

When colorizing visual grayscale images, the luminance is taken from the input image and only the chrominance has to be estimated. In contrast, colorizing TIR images requires estimation of both luminance and chrominance. Further, there is no direct relation between object appearance in TIR (the thermal signature) to its visual appearance (the perceived color). Hence, the problem at hand is more difficult than that of grayscale colorization and requires a process that generates RGB images from a semantic representation of the TIR.

Recent grayscale and NIR colorization methods base their success on Convolutional Neural Networks (CNNs) [3, 4, 7, 14, 18, 28, 32, 33, 34, 39] since they are able to model the semantic representation of an image. These techniques are dependent on large sets of training data. Finding training data for visual grayscale colorization is simple. In contrast, publicly available large datasets with corresponding TIR and visual RGB image pairs are rare, only one suitable instance could be found; the KAIST-MS traffic scene dataset [17]. For this dataset, a method for colorization of TIR images must be able to handle image pair misalignments. Since glass is opaque in TIR wavelengths, optics for thermal cameras are typically made of materials like germanium. Thus, it is difficult (but not impossible) to acquire thermal and visual images with the same optical axis. As a consequence, there are to our knowledge no available TIR and visual image pair datasets with perfect pixel to pixel correspondence.

Thermal infrared cameras, long-wave infrared (LWIR) cameras in particular, have seen an increase in popularity in recent years due to increased resolution and decreased cost. While previously being mostly of interest for military purposes, thermal cameras are entering new application areas [10]. Thermal cameras are now commonly used, e.g., in cars and in surveillance systems. The main advantages of thermal cameras are their ability to see in total darkness, their robustness to illumination changes and shadow effects, and less intrusion on privacy [2]. Due to the variety of inter-

¹http://richzhang.github.io/colorization/
²http://hi.cs.waseda.ac.jp:8082/
estig applications, but at the same time for humans difficult interpretation, transformation of TIR is highly relevant. Accurate transformation of TIR images significantly improves observer performance and reaction times in, e.g., tasks that involve scene segmentation and classification [35].

Contributions This paper proposes two methods, a two-step and an integrated approach, that transform TIR images into visual RGB images, a problem that is previously unaddressed (except for [29]). The proposed methods convert TIR images to plausible visual luminance and chrominance and are robust to image pair misalignments. The methods are evaluated, both qualitatively and quantitatively, on a publicly available dataset with convincing results.

2. Background

2.1. Infrared light and cameras

The infrared band of the electromagnetic spectrum is large and is usually divided into smaller parts depending on their properties. The near infrared (NIR) band is dominated by reflected radiation and is dependent on illumination. Essentially, it behaves just as visual light, except that we cannot see it. In contrast, the thermal infrared (TIR) band is dominated by emitted radiation. That is, the “light” received by the camera is mostly emitted from the observed objects and is related to the temperature of the object, not reflected from a light source (such as the sun). TIR is commonly subdivided into mid-wave (MWIR, 3-5 µm), long-wave (LWIR, 8-12 µm), and sometimes also far infrared (FIR). Objects at normal everyday temperatures emit mostly in the LWIR band (while a hot object like the sun emits most in the visual band), thus making the LWIR band the most suitable for night vision. In addition, cameras for LWIR based on microbolometer sensors have become more common and less expensive in recent years.

2.2. Related work

Early visual grayscale colorization methods have been heavily dependent on user input and interaction in the form of e.g. scribbles [27], texture classification [31], extracted features [16], as well as reference images for color transfer [15, 19, 37]. In recent years, deep Artificial Neural Networks, or more specifically, Convolutional Neural Networks (CNNs) [9, 26] have been successfully applied to a wide range of topics, e.g. image classification [23], image style transfer [21], and super resolution [6]. The success of deep learning inspired automatic CNN based visual grayscale image colorization methods [3, 4, 7, 14, 18, 32, 39] as well as NIR colorization methods [28, 33, 34].

Automatic colorization methods have a few common problems. First, colorization is, as previously mentioned, an ambiguous problem that is heavily dependent on semantics. Global priors, e.g. time of day, weather or location, can affect the color scheme in an image. Current colorization methods mitigate this problem either by using millions of training images [18, 25, 39], by limiting the method to a specific type of images (such as bedroom images [3], or faces & churches [7]), or by including global priors from the dataset itself (day/night, indoor/outdoor, etc.) [18]. The latter approach requires a dataset with such annotations.

Second, colors, when automatically learnt, often tend to be desaturated. Further, certain objects, e.g. cars, can have various colors while other, related objects, e.g. police cars, should have a specific color. The problem of diverse colorization is addressed in several works [3, 7, 14, 25, 32, 39]. Zhang et al. [39] and Larsson et al. [25] address the color diversity problem by treating it as a classification rather than a regression task. Larsson et al. [25] predict per-pixel histograms and Zhang et al. [39] quantize the chrominance space and use class-rebalancing at training time. The loss is, however, per-pixel based and does not enforce spatial coherence which occasionally results in speckle noise. Further, Cao et al. [3] employ a conditional GAN architecture and, unlike most other methods, do not use an autoencoder structure. Deshpande et al. [7] use a Variational Autoencoder, and Guadarrama et al. and Royer et al. [14, 32] propose methods based on probabilistic PixelCNNs.

Third, a question arises: What is accurate colorization? How does one provide a measure of the accuracy of a colorization? Some papers use image distance error measures like Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR), or Structural Similarity (SSIM) [7, 25, 33] while others provide user studies [3, 18] or Visual Turing tests [14, 39].

The above-mentioned colorization methods estimate the chrominance from the luminance, and are thus not directly applicable to colorization of infrared images, where also the luminance has to be estimated. A few recent publications treat colorization of NIR images [28, 33, 34]. NIR images are dominated by reflected radiation (such as sunlight) reflected on the objects in the scene. The difference between NIR and visible red light is just a small shift in wavelength; NIR is thus quite similar to visual light, especially to the red channel of an RGB image. TIR images, on the other hand, are dominated by emitted radiation, which is correlated to the temperature, not the color, of the observed objects. The color of objects can thus be retrieved from a TIR image only by some higher level semantic information. Limmer et al. [28] propose a method for transferring the RGB color spectrum to NIR images using deep multi-scale CNNs. In order to preserve details, the high frequency features are transferred in a post-processing step. Suárez et al. [34] utilize a DCGAN architecture to colorize 64×64 patches of architectural NIR images. They improve the method in [33] by separating estimation of each channel into a three channel DCGAN architecture. No merging of patches is done and
the method is evaluated per patch only.

Infrared color fusion or mapping, which are well studied research areas [1, 13, 35, 41, 40], combine multispectral images by mixing pixel intensities or require a manually selected reference image or object segmentations [12], in contrast to the proposed methods.

Cross-domain image to image translation methods have made much progress recently [8, 20, 29, 38, 42]. The unsupervised method proposed by Liu et al. [29] is evaluated on thermal images in an early version of the paper. The translated TIR to RGB images have cartooning effects, probably due to regularization or a piecewise constant prior. The main difference to our approach is that we achieve smooth RGB results just by adjusting the objective function in a suitable way. We intended to directly compare to the method by Liu et al., but we did not have access to their TIR to RGB network model, neither on request.

In the following, we describe two novel methods that transform full TIR images into visual, i.e. perceptually realistic, RGB images, without requiring post-processing or user input.

3. Method

The two proposed methods are based on one proposed architecture. The architecture is inspired by the generator architecture in [20], an overview can be seen in Fig. 1. It has an autoencoder structure with an encoder and a decoder part. Details and key features are further discussed below.

3.1. Network architecture

Since there is no direct relation between the object appearance in TIR to its visual appearance, the proposed architecture is required to generate RGB images from a semantic representation of the TIR images. Motivated by the success of previous CNN based colorization methods, we assume that these underlying representations common to TIR and RGB images can be modelled using deep learning, and more specifically, CNNs with autoencoder structure. This approach has been verified in [29] where a Coupled GAN with a Variational Autoencoder (VAE) is used as generator.

An overview of the proposed architecture is presented in Fig. 1. The network has an encoder-decoder structure based on the generator architecture in [20]. The encoder contains four blocks, each block consists of a Convolution, BatchNormalization, and LeakyReLU layer. In addition, the second and third green block also contain an Upsampling layer. The Concatenation layers are called skip connections. Skip connections are used in [20] in order to preserve structure. A skip connection simply concatenates all channels at layer $i$ with those at layer $n - i$ (where $n$ is the total number of layers).

Both proposed methods are based on the described architecture. The first method is a two-step approach where the proposed architecture estimates the luminance and an existing grayscale to RGB method is used to estimate the chrominance from the luminance. The second method is an integrated approach that estimates both the luminance and the chrominance using the proposed architecture.
3.2. Representation

As input, the proposed architecture accepts one channel 8-bit TIR images (due to the 8-bit limitation of the dataset) of any width and height. The output is either a one or a three channel visual image of the same width and height as the input image. Regarding internal color representation, any representation could be used. CIELAB\(^3\) was chosen since it is designed to mimic human perception of colors. Hence, distances between colors in the CIELAB space corresponds to perceptual differences. The CIELAB color space is also used in [4, 18, 39]. Iizuka et al. [18] performed an evaluation of three color spaces (RGB, YUV, and CIELAB) and concluded that CIELAB gave the most perceptionally reasonable results.

Regardless of the color space used, the pixel values are normalized to the range \([0, 1]\) since the Sigmoid function is used in the output layer.

3.3. Objective function

The imperfect pixel to pixel correspondence of the TIR and RGB image pairs in the dataset (see Section 4.1) discourages the use of simple pixel to pixel loss functions, like e.g., \(L_1\) (and \(L_2\)). In addition, it is well known that the \(L_1\) and \(L_2\) loss produces blurry results on image generation tasks [24]. To mitigate this problem, the loss is separated into a luminance and a chrominance loss.

The human visual system has lower acuity for color differences than for luminance [30]. This is, e.g., exploited by perception-based compression methods like JPEG (which uses fewer bits for chrominance than luminance), and also by the grayscale colorization method designed by Guadarrama et al. [14]. Therefore, we propose to separate the loss for the luminance (\(L_L\)) and the chromatic (\(L_{ab}\)) channels and to use the \(L_1\) loss between the ground truth image \(y_{t}^{ab}\) and the estimated image \(y_{e}^{ab}\) on the chromatic channels only, as

\[
\mathcal{L}_{ab}(y_{t}^{ab}, y_{e}^{ab}) = |y_{t}^{ab} - y_{e}^{ab}|	ag{1}
\]

For the luminance channel (L), where humans have higher acuity, we employ an image quality measurement, Structural Similarity (SSIM) [36], between the ground truth image \(y_{t}^{L}\) and the estimated image \(y_{e}^{L}\):

\[
\text{SSIM}(y_{t}^{L}, y_{e}^{L}) = \frac{1}{M} \sum_{j=1}^{M} \frac{2\mu_{jt}\mu_{je} + c_1}{\mu_{jt}^2 + \mu_{je}^2 + c_2} \frac{2\sigma_{jte} + c_2}{\sigma_{jte}^2 + \sigma_{jte}^2 + c_2}
\]

where the local statistics, the mean \(\mu_{jt}\) and \(\mu_{je}\), the variance \(\sigma_{jte}^2\), and the covariance \(\sigma_{jte}\), are calculated within \(M\) pixel neighbourhoods using a sliding window\(^4\). The size of the sliding window is chosen to incorporate the imperfect pixel to pixel correspondence (see Section 4.3).

The constants, \(c_1 = (k_1L_{\text{max}})^2\) and \(c_2 = (k_2L_{\text{max}})^2\) stabilize the division with a weak denominator. \(L_{\text{max}}\) is the dynamic range \((L_{\text{max}} = 1\) in this case\) and \(k_1 = 0.01\) and \(k_2 = 0.03\). Color (3-channel) SSIM applied to CIELAB leads to strong clouding effects, as colors shift continuously in uniform regions. From SSIM the Structural Dissimilarity (DSSIM) is derived, which is suitable as a loss function:

\[
\mathcal{L}_L(y_{t}^{L}, y_{e}^{L}) = \text{DSSIM}(y_{t}^{L}, y_{e}^{L}) = 1 - \frac{\text{SSIM}(y_{t}^{L}, y_{e}^{L})}{2}
\]

The total loss \(\mathcal{L}\) is the sum of the two:

\[
\mathcal{L} = \mathcal{L}_L + \mathcal{L}_{ab}
\]

and the objective is to minimize the loss. In Fig. 2, examples of using the \(L_1\) and DSSIM loss functions on all channels can be seen together with examples of using DSSIM regularized with \(L_1\) as well as the proposed division of the loss function.

\(^3\)CIE L*a*b* D65 to be precise.

\(^4\)We use the implementation by https://github.com/farizrahman4u/keras-contrib/blob/master/keras_contrib/losses/dssim.py
4. Experimental results and evaluation

4.1. Dataset

We use the dataset from the KAIST Multispectral Pedestrian Detection Benchmark [17]. The dataset contains around 95,000 color-thermal (RGB-TIR), 8-bit, day and night image pairs (640×512) captured from a vehicle. The day training set consists of 33,395 RGB-TIR image pairs. Since the images are captured from a vehicle, a number of subsequent images are assumed to be similar depending on the vehicles speed. Therefore, only one quarter, 8,336 pairs (randomly chosen), were included in the training phase. The day validation set consists of 29,168 RGB-TIR image pairs. The complete validation set was used for evaluation.

The dataset has been recorded using a FLIR A35 microbolometer LWIR camera with a resolution of 320×256 pixels. During image alignment, the TIR images were upsampled to 640×512 [17] (using an unknown method). Therefore, the RGB-TIR pairs were downsampling back (using nearest neighbour) to 320×256 prior to training.

While studying the RGB-TIR image pairs, it became clear that they do not have perfect pixel to pixel correspondence. The pixel error in the vertical direction is estimated to be up to 4 pixels, and up to 16 pixels in the horizontal direction, increasing towards the lower left corner, see Fig. 3. Further, inspecting e.g. example 5 in Fig. 4, there seems to be a radial distortion not accounted for in the TIR images. Further, the error appears to increase when the car is moving, indicating a slight offset in camera synchronization.

4.2. Methods

Three different TIR to visual image transformation approaches have been evaluated. The latter two, TIR2L and TIR2Lab are the proposed methods:

- **Naïve** Estimating the chrominance using the TIR image as luminance using an existing grayscale colorization method.
- **TIR2L** Estimating the luminance using the proposed network and then the chrominance using an existing grayscale colorization method.
- **TIR2Lab** Estimating both the luminance and the chrominance using the proposed network.

The method proposed by Zhang et al. [39] was chosen as the reference colorization method. The reasons were that it does not require global priors (as e.g. the method by Iizuka et al. [18]) and neither does it produce multiple versions of the same grayscale image (like e.g. Guadarrama et al. [14] and Royer et al. [32]). The reference colorization method was trained from scratch on the given dataset.

4.3. Training

The proposed architecture is implemented in Keras [5] with Tensorflow back end. For network training, we use the ADAM optimizer [22]. Weights are initialized with Xavier normal initialization [11]. The proposed architecture was trained for 100 epochs using 8,336 samples with a batch size of 8 samples. The parameters of the ADAM optimizer were set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and learning rate 0.001. The LeakyReLU layers in the encoder had $\alpha = 0.2$ and the Dropout layer had a dropout rate of 0.5.

The size of the sliding window in the DSSIM loss function was set to $4 \times 16$ pixels in order to incorporate the misalignment of the dataset described in Section 4.1. Training of the proposed architecture was done on a NVIDIA GTX1080 GPU. A single training epoch with batch size 8, 1,042 iterations, and 8,336 samples took about 10 minutes. In total, the network was trained for 100 epochs (about 18 hours) at image resolution 320×256 pixels.

As implementation of the method by Zhang et al. [39] the Keras implementation by de Boissiere\(^5\) has been chosen. The experiments have been performed on a NVIDIA K40 GPU. A single training epoch with batch size 8, 1,042 iterations, and 8,336 samples took about 1 hour and 36 minutes. In total, the network was trained for 38 epochs (about 2 days and 18 hours) at image resolution 320×256 pixels. At 38 epochs, the loss had converged and the training was considered to be finished.

4.4. Quantitative evaluation

Four image distance error measurements for quantitative evaluation have been used: $L_1$, Root Mean Squared Error (RMSE), Peak Signal To Noise Ratio (PSNR), and Structural Similarity (SSIM). Error measurements are calculated in the RGB color space in range $[0, 1]$, because this is the expected format for SSIM. Results for the Naïve, TIR2L, and

\(^5\)https://github.com/tdeboissiere/DeepLearningImplementations/tree/master/Colorful
Table 1: Image distance validation results, mean and standard deviation for 29 168 image pairs.

<table>
<thead>
<tr>
<th>Method</th>
<th>L1</th>
<th>RMSE</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.24</td>
<td>0.84</td>
<td>9.34</td>
<td>0.43</td>
</tr>
<tr>
<td>±0.04</td>
<td>±0.10</td>
<td>±1.23</td>
<td>±0.08</td>
<td></td>
</tr>
<tr>
<td>Naïve</td>
<td>0.24</td>
<td>0.82</td>
<td>9.55</td>
<td>0.45</td>
</tr>
<tr>
<td>±0.04</td>
<td>±0.09</td>
<td>±1.21</td>
<td>±0.08</td>
<td></td>
</tr>
<tr>
<td>TIR2L</td>
<td>0.14</td>
<td>0.46</td>
<td>14.7</td>
<td>0.64</td>
</tr>
<tr>
<td>±0.04</td>
<td>±0.09</td>
<td>±2.18</td>
<td>±0.08</td>
<td></td>
</tr>
<tr>
<td>TIR2Lab</td>
<td>0.13</td>
<td>0.46</td>
<td>14.7</td>
<td>0.64</td>
</tr>
<tr>
<td>±0.04</td>
<td>±0.09</td>
<td>±2.20</td>
<td>±0.08</td>
<td></td>
</tr>
</tbody>
</table>

TIR2Lab methods can be seen in Table 1. In addition, error measurements are calculated between the input TIR image and the true RGB image in order to provide a baseline. All evaluations were performed using the day validation set.

The Naïve method gives a slight, but not significant, improvement compared to the baseline for all image distance measurements in the evaluation. Comparing TIR2Lab to the baseline, it is clear that it gives a significant improvement in terms of image distance.

TIR2L and TIR2Lab have similar results, indicating that the chrominance can be estimated by the proposed network equally well as with [39] for this dataset. However, estimating both luminance and chrominance simultaneously using TIR2Lab is less computationally demanding than first estimating the luminance and then the chrominance from the luminance in a two-step approach as in TIR2L. A forward-pass through TIR2Lab with batch size 8 takes around 1.33 seconds while a forward-pass through TIR2L takes around 2.83 seconds on a NVIDIA GTX1080 GPU.

### 4.5. Qualitative evaluation

In Fig. 4, six transformation examples for the Naïve, TIR2L, and TIR2Lab methods are provided.

The Naïve method gave a slight improvement compared to the baseline. Since the luminance is taken directly from the TIR image, there is no degradation in terms of structure compared to the original TIR image. The method proposed by Zhang et al. [39] does, however, fail to correctly estimate the chrominance which is clearly visible in Fig. 4b.

The colorized images by TIR2L, Fig. 4c, and TIR2Lab, Fig. 4d, have similar appearance. TIR2L appears to have a stronger tendency to colorize the sky in a more pink color which is more similar to ground truth in some cases (ex. 1) and less in others (ex. 5, 6). The dataset was recorded during sunset and the images in the training set have skies with a varying degree of pink. TIR2L also colorizes some road markings in a more vivid orange than TIR2Lab (ex. 3) but sometimes it is the other way around (ex. 1).

Based on our observations, we conclude that the results are similar in terms of subjective assessment. TIR2L transforms the TIR image to a more plausible RGB image in some cases, and vice versa.

Fig. 5 provides a few transformation examples of particular interest. Note that objects that have adopted the background temperature and objects for which different colors is not equivalent to different thermal properties will not be plausible colorized unless they are hallucinated or there is some semantic information related to the different colors. In ex. 5a there is a faint line visible to the left of the blue line in the true RGB image. This line is barely visible in the TIR image and the proposed methods colorizes it only partially. TIR2L colorize the blue line white while TIR2Lab chooses an orange color. The same goes for the case in ex. 5d.

In ex. 5b, both TIR2L and TIR2Lab fail to colorize the road markings since they have the same apparent temperature as the road paving. A similar scenario is the one in ex. 5c where TIR2Lab fails to colorize the crossing as both white and orange (there are both kinds in the dataset). TIR2L on the other hand adds orange markings between the white markings.

Further, in ex. 5d, the brake lights of the vehicles are not colorized correctly. Turning on the brake light does not change the apparent temperature of the lamp cover. However, for the specific application of night vision in traffic scenes, it is possible to fuse colorized images with the true RGB image, where the brake lights (when it is dark) will be clearly visible.

There are two scenarios for which both TIR2L and TIR2Lab fail in most cases. Both are shown in ex. 5e. We believe that this is because urban environments and cars close to the camera are not as frequently occurring in the dataset as more rural environments and cars at longer distances.

### 4.6. Night to day

In addition to the above mentioned experiments, LWIR night images from the KAIST-MS dataset have been colorized using TIR2Lab (trained on day images). Two examples of colorized day RGB images together with true RGB night images can be seen in Fig. 6. At night, the surroundings will adopt a more homogeneous temperature, thus the contrast in the LWIR images will be lower than during the day. This can potentially be (partly) compensated for by adjusting the dynamic range of the 16-to-8 bit conversion, but was unfortunately not done in the dataset. The low contrast makes it difficult for the network to correctly recognize the different objects in the scene and the output RGB image, colorized with a day color scheme, looks blurred.
5. Conclusion

We have addressed the problem of transforming TIR images into visual, perceptually realistic color images, an unaddressed problem despite its relevance for practical applications. In our work, we have proposed a general, deep learning based approach to generate luminance and chrominance information from TIR images. We further suggest two methods based on different variants of the proposed approach to determine chromatic information. The methods are robust to image pair misalignments and have been evaluated both qualitatively and quantitatively on a publicly available dataset. The evaluation was, however, limited to traffic scene images due to the lack of large, publicly available TIR-RGB image pair datasets.

In comparison to grayscale to RGB colorization, which only estimates the chrominance, a TIR to RGB transformation method has to estimate both luminance and chrominance. There is no direct relation between object appearance in TIR to its visual appearance. The proposed approach was, therefore, based on Convolutional Neural Net-
Figure 5: Failure cases for the proposed methods, first row shows input TIR image, second row the output from TIR2L, third row the output from TIR2Lab, and fourth row the true RGB images.

Figure 6: Examples of a) two night TIR images, colorized using b) TIR2Lab trained on day images and their corresponding c) true RGB images.

works, due to their ability to model semantic representations. The first proposed method estimates both plausible luminance and chrominance using the proposed approach. The second proposed method estimates luminance using the proposed approach and chrominance using an existing grayscale to color transformation method, the method proposed by Zhang et al. [39] in this case⁶.

Further work is twofolded. First, in order to provide a more extensive evaluation of future TIR to RGB transformation methods, new large TIR-RGB image pair datasets that target other application areas are needed. Second, future work for the method includes tuning of weights for the losses $L_L$ and $L_{ab}$. Failure cases for the proposed methods include a slight loss of structure compared to the input TIR image and cloudy colorization of uniformly colored areas, e.g., pavements. We do, however, conclude that separating the objective function into luminance and chrominance loss is favourable.

Acknowledgements

The research was funded by the Swedish Research Council through the project Learning Systems for Remote Thermography (2013-5703) and the project Energy Minimization for Computational Cameras (2014-6227).

⁶Code is available at https://gitlab.ida.liu.se/amabe60/PBVS2018
References


