Recurrent Color Constancy

Yanlin Qian[1], Ke Chen[1,*], Jarno Nikkanen[2], Joni-Kristian Kämäräinen[1], Jiri Matas[1,3]
[1]Laboratory of Signal Processing, Tampere University of Technology
[2]Intel Finland
[3]Center for Machine Perception, Czech Technical University in Prague

Abstract

We introduce a novel formulation of temporal color constancy which considers multiple frames preceding the frame for which illumination is estimated. We propose an end-to-end trainable recurrent color constancy network – the RCC-Net – which exploits convolutional LSTMs and a simulated sequence to learn compositional representations in space and time. We use a standard single frame color constancy benchmark, the SFU Gray Ball Dataset, which can be adapted to a temporal setting. Extensive experiments show that the proposed method consistently outperforms single-frame state-of-the-art methods and their temporal variants.

1. Introduction

The human visual system perceives colors of objects independently of the incident illumination under varying conditions. The phenomenon is known as color constancy.

A pre-processing step compensating effects of changing illumination is necessary for consumer photos to look natural. It is also beneficial in a number of computer vision and graphical applications requiring intrinsic color information *e.g.* fine-grained classification, semantic segmentation and scene rendering. Consequently, illumination color compensation, known as the automatic white balancing, has become an essential component of the pipeline for processing color images [31].

The color constancy problem on still images has been investigated for decades [1, 7, 9, 13, 18, 39]. Assuming that the illumination is identical for all pixels in an image, the problem can be expressed as:

$$\hat{c} = f(I) \tag{1}$$

where function $f(\cdot)$ is the estimator of the groundtruth illumination vector c_{gt} for a single image I. With estimated illumination $\hat{c}$, chromatic adaption [14] is then adopted to correct color to obtain an illumination-independent image.

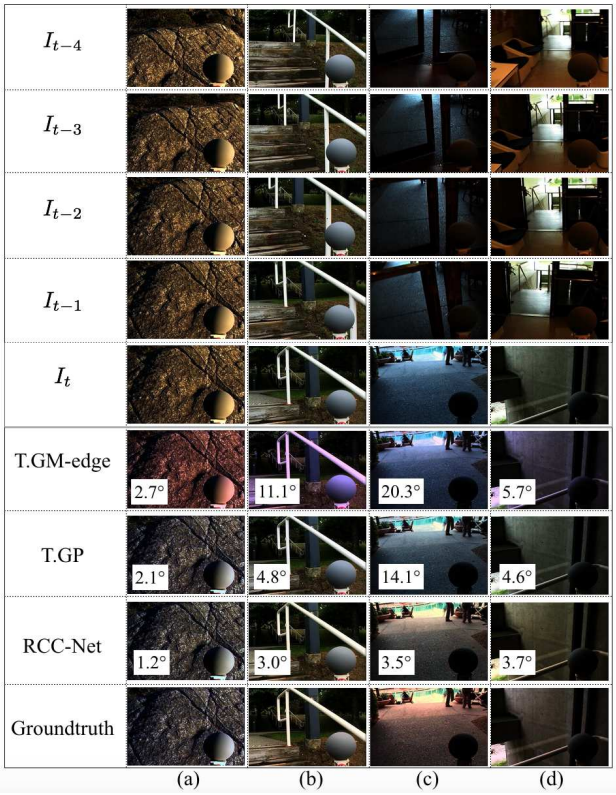

Figure 1. Color correction by temporal methods on image I_t using five-frame sequences with (c)(d) and without (a)(b) significant pictorial content change, with (d) and without (a)(b)(c) significant illumination color change. RCC-Net is the proposed method. T.GM-edge and T.GP refer to temporal extensions of the standard Gamut-based [19] and Gray Pixel [44] methods, respectively. Illumination color is visible on the ball in the bottom right corner. The angular error is shown in the bottom left corner of each color-corrected image. The images are from the SFU Gray Ball *linear dataset*, *i.e.* without gamma correction and thus appear to have unusual color composition.

With the rapid development of both cameras and storage devices, taking videos becomes more and more popular

*Corresponding author

in everyday life. For some latest devices like the popular iPhone, a short video can be taken and stored when shooting a photo. Yet the problem of estimating illumination chromaticity in videos has received minimum attention. This problem, which we call *temporal color constancy*, is formulated as:

$$\hat{c}_t = f\left(I_{t-(N-1)}, I_{t-(N-2)}...I_{t-1}, I_t\right) \quad , \qquad (2)$$

where $f(\cdot)$ uses, besides the shot frame I_t to be corrected, the $(N-1)$ preceding frames $I_{t-1}\dots I_{t-(N-1)}$ (rows $I_{t-1}\dots I_{t-4}$ in Figure 1).

Most of the existing algorithms [1, 7, 9, 13, 18, 39] were designed for compensating the effect of illumination in separate images rather than videos. The straightforward solution is to apply those single-frame color constancy methods to processing videos frame by frame. However, it is evident that temporal correlation of illumination changes in adjacent frames is not exploited in those algorithms, which can play a vital role in temporal color constancy.

Recently, several papers concerning color constancy on image sequences have appeared [29, 42, 45]. However, these algorithms make strong assumption, requiring identical illumination either for all video frames [29, 45] or a small set of frames [42]. This seems reasonable assumption in high frame rates as illumination changes are not expected to be abrupt. In practical use, in a video shot, illumination across video frames can be time-varying (*e.g.* column (d) in Figure 1) rather than constant. Illumination changes can have various reasons, *e.g.* the changes of illuminant chromaticity and varying viewing angles of the camera. In this sense, this paper aims at relaxing the assumption and coping with temporal color constancy under the varying illumination conditions in a varying-length sequence.

Inspired by the success of convolutional neural networks (CNN) in single-frame color constancy [6, 30] where CNN can learn powerful perceptual representations, we adapted it to the problem of color constancy in videos (T.CNN+MSVR in Section 5.1) via a simple temporal pooling. Another two single-frame methods (Gamut-based method [19] and Gray Pixel [44]) are also adapted in a natural way. The adaptation slightly improves their performance in videos, except on Gamut-based method.

Ideally, a temporal method should learn temporal dynamics and also allow varying-length sequence as input. Considering that long short-term memory (LSTM) unit embodies intrinsic mechanism for capturing inter-frame correlation [12], we build RCC-Net on the convolutional LSTM. We further present a task-specific data augmentation in both training and testing phase, namely simulated sequence (SS), based on a plausible assumption that a method estimating global illumination should give identical estimation in different spatial regions. SS consists of generated image patches along the spatial domain following a random simulated zoom-in trajectory, which carries global-to-local spatial information. SS is integrated into the RCC-Net by a second convolutional LSTM. By end-to-end optimization, the RCC-Net jointly learns temporal dynamics and deep visual representation, which we discover to be very beneficial for illumination estimation. Extensive experiments on two temporal variants (non-linear and linear RGB space) of the SFU Gray Ball Dataset verify that the proposed method achieves significantly better performance than several state-of-the-art single-frame methods and their temporal extension.

In a summary, the contributions of this paper are:

- We formulate generically the problem of temporal color constancy, relaxing the impractical assumptions about constant or piece-wise constant illumination for image sequences in the existing works[1].
- We present RCC-Net (a novel recurrent deep net) for temporal color constancy task. The RCC-Net is further equipped with a simulated sequence module, which boosts its performance by a large margin[2].
- We show that the popular SFU Gray Ball dataset [11] is suitable for the temporal setting, and introduce a Temporal SFU Gray Ball benchmark. We experimentally evaluate state-of-the-art methods for temporal color constancy on the new benchmark.

2. Related work

Single-Frame Color Constancy is a well-established problem [20] of estimating the color of image pixels under gray light only given pixel color values (I_t) under unknown illumination. Existing color constancy methods can be grouped into three categories: *static algorithms* [7, 8, 39, 44], *gamut-based algorithms* [19] and *learning-based algorithms* [3, 10, 15, 18, 26, 35, 40]. The static algorithms work on the assumption that zero-order [7, 8], first-order [39], and/or higher-order [39] statistics of some pixels in a image have gray average color. Gamut-mapping based methods assume that in the real world, only part of the color spectral distribution of objects is observable [19]. With the rise of convolutional neural networks [27], several CNN regression learning based approaches [6, 28, 30, 35] have recently achieved the state-of-the-art performance for color constancy. Recent work [3] formulated the problem as a localization task in a 2D log-chroma histogram space, yielding state-of-the-art performance. All of the publications deal with the color constancy problem in a single image.

Color Constancy for Image Sequences Only a few authors [4, 29, 45, 42] have investigated temporal color constancy. Yang *et al.* [45] first estimated illuminant chro-

[1]Note that *knowing the illumination is constant is a significant constraint (if the assumption is correct, and exploited, results improve).*

[2]code: https://github.com/yanlinqian/RCC-Net

maticity from pairs of point correspondences in consecutive images and then adopted majority voting for discretized values of the illuminant chromaticity. Similarly, Prinet *et al*. [29] kept the same dichromatic reflection model but replaced majority voting by a more robust probabilistic optimization. Both [29, 45] are early attempts to cope with automatic white balance in image sequences, but require high frame rate (*e.g*. 60 Hz frame rate in [29]) to keep the incident illumination identical in a short space-time domain. Moreover, both methods are limited to processing image pairs. Wang *et al*. [42] proposed a simple yet effective multi-frame illumination estimation method by clustering illumination of each frame into a number of video shots and then adopting the statistics (mean or median value) of illumination estimation within each shot as its global illumination. The strong assumption on constant illumination in one sequence is relaxed to piece-wise constant illumination. Recently, Barron *et al*. [4] handled temporal color constancy by constructing a Kalman filter-like smoothing model for image sequences, on the basis of reducing their single-frame method [3] CCC to localize a signature on a log-chroma torus space. The method is aimed at smoothing erratic predictions from neighboring frames, but it cannot capture temporal dynamics in a sequence of frames.

Convolutional LSTM – Long Short Term Memory networks are a special kind of Recurrent Neural Network (RNN), first introduced by Hochreiter and Schmidhuber [24] to learn long-term dependencies in sequences. Convolutional LSTM is LSTM equipped with CNN, considered as typically "deep in space" and "deep in time" respectively, which can be seen as two modalities of deep learning. CNNs have achieved massive success in visual recognition tasks [6, 30], while LSTMs sparkle in long sequence processing [33, 37]. Because of the decent properties (rich visual description, long-term temporal memory and end-to-end training) of the convolutional LSTM, it is heavily investigated for many other computer vision tasks involving sequences (e.g. activity recognition [12], image captioning [25], human re-identification in videos [43] and video description [12]) and brought significant improvement. Our work is the first attempt to introduce convolutional LSTM to the field of color constancy.

3. Temporal SFU Gray Ball Datasets

In this section, we explain how the existing and widely adopted single-frame color constancy benchmark, the SFU Gray Ball [11], was used for temporal color constancy evaluation. The Gray Ball was generated from 15 video clips with significantly different content. From each video, $81 - 1312$ frames were selected and provided with ground truth. The videos are sampled at roughly 3 frames per second. The images in the original dataset are stored in a non-linear device specific RGB color space. We refer to the original images as the non-linear SFU Gray Ball dataset. Following the procedure in [17], we modified this set by applying gamma-correction (gamma $= 2.2$). The resulting images are assumed to be approximately linearized and thus we call the transformed image dataset the linear SFU Gray Ball dataset.

Another temporal color constancy dataset, containing 11 videos, is introduced by Prinet *et al.* [29]. However, the videos are short, consisting of 13 frames. The camera motion is small and the illumination is constant. The small size of this dataset does not permit training a deep net.

In the experiments, all evaluated methods were run in a *causal way*, i.e. the prediction of illumination color of the frame was based solely on its content and on past frames. Only such methods are suitable for on-line applications such as real-time camera white balancing. Non-causal processing utilizes past and future knowledge to help estimate illumination. In this work we only consider causal processing.

Our benchmark protocol is straightforward. For learning-based methods we use 15-fold cross-validation by leave-one-sequence-out. Sequence border effects were handled by repetition of the first frame in a video. Since predictions for all frames of the videos were made, the proposed method can be compared with single-frame methods. For temporal color constancy experiments, the number of frames processed as a sequence used for predictions is important parameter, which we denote subsequence length N. In our experiments (see Table 3) we tested different values of N. Five-frame example sequences from the SFU Gray Ball Dataset with different characteristics are shown in Figure 1. Consistent with the pre-processing of the ordinary Gray Ball Dataset, the pixels of the gray sphere, which is in a known fixed location,1 are excluded by cropping.

4. Recurrent Color Constancy Networks

In this section, we present the RCC-Net – an end-to-end trainable recurrent color constancy network (Figure 2). The proposed model has two parallel convolutional LSTM subnetworks, one for processing the original frame sequence and the other for processing a simulated spatial sequence. The simulated sequence consists of randomly generated frames from the shot frame (Section 4.2). Simulated sequences can be produced online and therefore the two subnetworks can run in parallel and their outputs are concatenated in a merging layer implemented as a single shallow network.

There are three essential components in the RCC-Net:

1. Convolutional LSTM for temporal sequence (Section 4.1, the top branch in Figure 2),
2. Simulated sequence network for training(Section 4.2, the bottom branch in Figure 2),

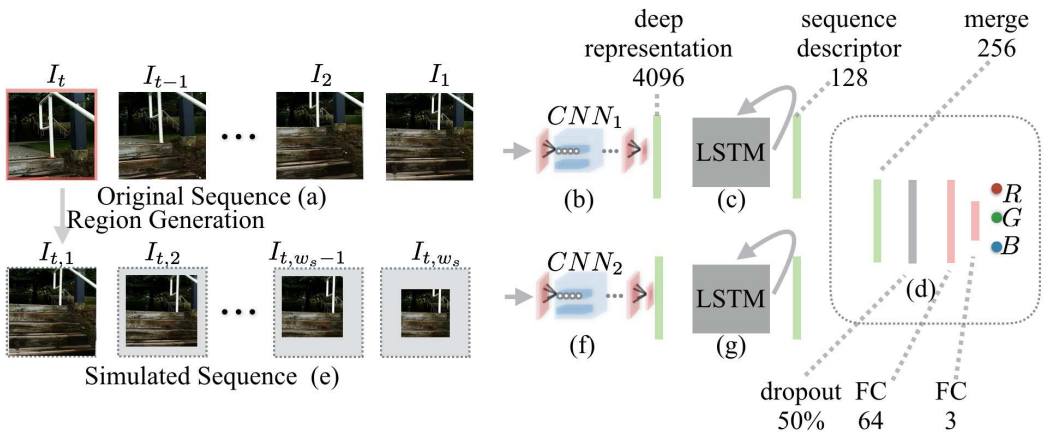

Figure 2. The RCC-Net architecture. The RCC-Net learns and infers in an end-to-end manner and outputs a global illumination vector for the shot frame (highlighted in red rectangle) given an image sequence. The RCC-Net has two independent convolutional LSTM sub-networks, one (b, c) for processing the input temporal sequence (a) and the other (f, g) for processing a simulated spatial sequence (e). The outputs of the two sub-nets are combined in the shallow network (d).

3. End-to-end optimization of the whole net (Section 4.3).

Given an unseen test sequence, we obtain an estimate of global illumination of the shot frame using the RCC-Net. Afterwards we correct the color of the final shot frame with the estimated illumination. We apply the standard von Kries model [41] to correct the R, G and B channels by independent scaling.

4.1. Convolutional LSTM for Color Constancy

Given an image sequence (Figure 2(a)), a convolutional LSTM is employed to produce a temporal sequence descriptor for global illumination of the shot frame. Aiming at end-to-end learning, an integration of fully-connected layers (Figure 2(d)) is used to map the high-dimensional LSTM descriptor to a 3-dimensional chromatic vector $\hat{c} = (\hat{c}_R, \hat{c}_G, \hat{c}_B)$. Specifically, our shallow network consists of two fully-connected layers and one Dropout layer controlling the training data over-fitting. The convolutional LSTM can be divided into CNN and LSTM, which we will describe separately in the following paragraphs.

CNN for Feature Extraction. Inspired by its success, we adopt the 19-layer VGGNet [36] (Figure 2(b)) with layers removed after the fully-connected fc_6 to directly output deep representation of each frame. Following [30], with the limited training data, we do not fine-tune the network. The 4096-dimensional representations after the non-linearity of the fully-connected fc_6 are then used as the sequential LSTM input.

LSTM for Sequence Processing. Let us assume a sequence of CNN representations of input frames as input and a vector describing the last (shot) frame color as output. The key challenge is the design of the model to recursively process a sequence and produce a single vector. Here LSTM unit (Figure 2(c)) is adopted to learn such "*many-to-one*" mapping. More particularly, during training, our LSTM model takes a sequence of 4096-dimensional deep representations (CNN_1,CNN_2,...,CNN_t). LSTM computes a sequence of hidden states (h_1,h_2,...,h_t), but produces only a 128-dimensional output (y_t), by iterating the following equations:

$$h_t = f(CNN_t, h_{t-1}) \tag{3}$$

$$y_t = sigmoid(w_{oh} h_t + b_o), \tag{4}$$

where $f(\cdot)$, w_{oh} and b_o are learnable functions and parameters to be optimized in LSTM. Function $f(\cdot)$ includes input gate, forget gate, output gate and the computation of memories [24]. Equation (4) only runs in the last iteration (the shot frame at the time step t), outputting a sequence descriptor y_t for the illumination regression step. The number of hidden layer neurons in our LSTM is 128.

The training proceeds in the following order: initializing all weights by *Glorot uniform* [21] and all cells by *orthogonal vectors* [23], h_t is computed recurrently on the basis of CNN_t and h_{t-1} until the end of the sequence. Note that LSTM is a intermediate unit in our framework which is trained entirely end-to-end and no auxiliary loss function is employed in the LSTM.

4.2. Simulated Zoom-in-like Sequence (SS)

Our hypothesis for adopting simulated shot frame sequences is the following: intra-frame global illumination is consistent for all regions in the frame. This is clearly not always true due to spatial distribution the illumination. However, for the global-illumination setting, we empirically prove that the approach boosts the estimation performance (see Table 1). The method requires a process for spatial region generation. We tested several region-generating

Figure 3. Simulated zoom-in-like sequences (red → yellow → green → blue) generated from the three images. These sequences simulate camera zoom-in actions in the view finding stage that end with a "frame" shot by user.

methods (*e.g.* spatial pyramid [22], random patches, *etc.*). The results are given in Table 2.

Inspired by people who often move a camera, zoom in, and finally focus on a certain object, we introduce a region-generating strategy called the *Simulated zoom-in-like Sequence* (SS). SS is composed of multiple synthesized frames generated by a random zoom-in path consisting of sub-windows with geometric transformations applied (see the examples in Figure 3). As opposed to other region-generating methods, our SS produces smoothly-changing frames, which are not visually isolated. We ensure that in each pair of consecutive frames in SS, the latter frame is fully contained in the previous one.

The generation of SS is defined by a number of free parameters. These parameters are resolution-independent and refer to the ratios with respect to the original image size (for the first generated image) or last generated image (for generated image after the first one). All random parameters are generated from a uniform distribution:

- **zoom_range:** The frame-wise zoom-in scale parameter 0.8.
- **xshift_range:** Random horizontal translation shift $[-0.1, 0.1]$.
- **yshift_range:** Random vertical translation shift $[-0.1, 0.1]$.
- **rotation_range:** Random in-plane rotation $[-5°, 5°]$.

SS is generated only from the shot frame, then fed to an AlexNet-based CNN (Figure 2(f)) [6] followed by a LSTM for SS (Figure 2(g)). The reason we switched to the AlexNet-based CNN here instead of VGGNet is that we achieve competitive performance with less computation. One explanation of this can be that SS consists of low-resolution frames and color constancy is a low-level vision task. It is easy to integrate the convolutional LSTM for SS to our RCC-Net, owing to the merging layer added to the front of the shallow regression network, *i.e.* Figure 2(d). From the data perspective, we consider SS as a specific data augmentation in both training and testing phase, providing more regulating data for our RCC-Net.

4.3. End-to-End Optimization

As shown in Figure 2, RCC-Net training takes image sequences $(I_1^k,\ldots,I_t^k)_{k=1}^{\mathrm{N}}$ and illumination ground truth colors $c_{gt,t}^k$ as inputs. As the objective function, general multi-output CNN regression and related CNN-based color constancy algorithms use the Euclidean loss [6, 35]. However, instead of the Euclidean loss we employ the *angular error* in Equation (5) as the objective function since common performance metrics are based on it. RMSprop [38] optimization strategy with mini-batches of 128 sequence-illumination pairs is employed and we complete 50 epochs to train our deep model.

5. Experiments

5.1. Adaption of Single-frame Methods

For fair comparison, the single frame state-of-the-art color constancy algorithms can be easily "upgraded" for sequence processing. A straight-forward solution is to apply a statistical approach and estimate the illumination color as a mean or median value of the per-frame estimates in the sequence of frames [32, 42].

From the set of single-frame methods compared in [10, 30, 44], three well-performing methods are selected and specifically modified to the temporal setting. One method achieving relatively competitive performance from static, gamut-based and learning-based algorithms respectively was selected. Specifically, we choose Gray Pixel [44], GM-edge [19] and CNN+MSVR [30].

Temporal Gray Pixel (T.GP) – We employ the Gray Pixel algorithm [44] to generate gray pixels, i.e. illumination color estimates, for all frames. The illumination in the shot frame is estimated by averaging the gray pixels over a variable length subsequence preceding the frame.

Temporal GM-edge (T.GM-edge) – This method incrementally builds the color space convex hull (gamut) along the sequence, and the final gamut is used for illumination estimation (mapping of the gamut).

Temporal CNN+MSVR (T.CNN+MSVR) – For learning based CNN+MSVR, we simply extend CNN+MSVR by obtaining the channel-wise mean vectors of the illumination vectors for all frames in the sequence.

5.2. Parameter Settings

The following settings were used in the experiments:

- Data preprocessing of image sequences: subtracting channel-wise mean from each channel.
- The sub-sequence length N controlling the length of input sequence – $N = \{1, 2, 5, 10\}$.
- The simulated sequence length $w_s = 5$.
- The optimizer is set to *RMSprop* [38] for end-to-end training of the complete model.

Table 1. Color constancy on the *non-linear* and *linear* SFU Gray Ball Datasets. *90%* refers to the 90%-percentile of the obtained angular errors. All values are in degrees as defined in (5). The source of the results: w - color constancy benchmarking website [17]), p - the cited paper, and r - from our rerun of authors' implementation, i - our implementation.

		SFU Gray Ball (non-linear)					SFU Gray Ball (linear)			
		Med	*Mean*	*90%*	*Max*		*Med*	*Mean*	*90%*	*Max*
Single Frame Methods										
Gray World (GW) [7]	(w)	7.0	7.9	–	48.1	(w)	11.0	13.0	–	63.0
General GW (gGW) [1]	(w)	5.3	6.1	–	41.2	(w)	9.7	11.6	–	58.1
Gray Pixel (edge) [44]	(p)	4.6	6.1	–	–		–	–	–	–
GM-edge [19]	(w)	5.8	6.8	–	40.3	(w)	10.9	12.8	–	58.3
1^{st}GE [39][1]	(w)	4.7	5.9	–	41.2	(w)	8.8	10.6	–	58.4
SVR [15][2]	(p)	–	–	–	15.9	(w)	11.2	13.1	–	59.6
Automatic-CC [5][3]	(p)	3.2	4.8	–	–		–	–	–	–
NIS [18]	(w)	3.9	5.2	–	44.5	(w)	7.7	9.9	–	56.1
Exemplar-based [26]	(w)	3.4	4.4	–	45.6	(w)	6.5	8.0	–	53.6
Top-down [40]	(i)	–	–	–	–	(w)	8.3	10.2	–	63.0
Regression Tree [10]	(r)	4.8	6.1	13.1	30.6	(r)	8.5	10.6	**22.2**	56.3
Existing Temporal or Multi-frame Methods										
Image Sequence [29] [4,5]	(p)	4.6	5.4	–	–		–	–	–	–
Wang [42]	(p)	4.1	5.4	–	26.8		–	–	–	–
Extended methods (Section 5.1)										
T.GP	(i)	4.7	6.0	16.7	25.8	(i)	9.7	12.4	32.4	49.7
T.GM-edge	(i)	8.2	9.3	21.8	37.8	(i)	12.8	14.5	33.7	57.3
T.CNN+MSVR	(i)	4.0	4.8	12.7	26.0	(i)	7.2	10.0	33.7	48.1
Our Convolutional LSTM based method										
RCC-Net (no SS)	(i)	3.2	4.5	13.6	**23.2**	(i)	6.3	7.7	23.7	45.9
RCC-Net	(i)	**2.9**	**4.0**	**12.2**	25.2	(i)	**5.1**	**7.2**	22.5	**45.7**

[1] For GE, the original paper reports a different result – median error of 4.1, which results from the evaluation experiments performed only on a subset of 150 images.
[2] For SVR, only 2-fold cross-validation was made and only average RMS and maximum error were reported.
[3] For Automatic-CC, their evaluation are performed only on a subset of 1135 images.
[4] The setting of linear RGB only applies to the Prinet dataset, while other experiments are evaluated on the *non-linear* Grey Ball dataset [29].

- The image size for VGG (w×h): *224×224*.
- The image size for AlexNet-based SS: *32×32*.

Free parameters of the temporal extensions in Section 5.1 are tuned by 15-fold cross-validation.

5.3. Performance Metric

Following the prior works [2, 11, 16, 34], we adopt the angular error ε between the estimated illumination vector $\hat{c}$ and the groundtruth c_{gt} as the performance measure:

$$\varepsilon_{\hat{c},c_{gt}} = \arccos\left(\frac{\hat{c}\cdot c_{gt}}{\parallel \hat{c} \parallel \parallel c_{gt} \parallel}\right), \tag{5}$$

where $\cdot$ denotes the inner product between the two vectors and $\parallel \quad \parallel$ is the Euclidean norm.

5.4. Results

Table 1 compares the proposed methods with the state-of-the-art single-frame methods and their adapted variants (Section 5.1) in terms of the median, mean, maximum and $90th$-percentile of the obtained angular errors. We report the experimental results on both non-linear and linear temporal SFU Gray Ball benchmarks. With the exception of the maximum errors, the (*RCC-Net*) obtains the best performance on the non-linear dataset with leading by at least 15% on median, 8% on mean angular error. On linear one, this performance improvement is more evident – over 22% on median and 10% on mean error. We note that the maximum errors results from a few of incorrect "ground truth" labels in the SFU Gray Ball Dataset (please refer to our supplementary material in the code page).

The behavior of the methods for five sequences with large groundtruth changes is illustrated in Figure 4. The plots show that the proposed RCC-Net (red line) has relatively low error, demonstrating the approach performs better in varying-illumination conditions (*e.g.* relative change of the spatial arrangement of the object viewed, changing lighting conditions) in comparison to other temporal methods. Experimental results under challenging conditions are also presented in Figure 1, *i.e.* varying-illumination and/or varying-content (Figure 1(c) and (d)) .

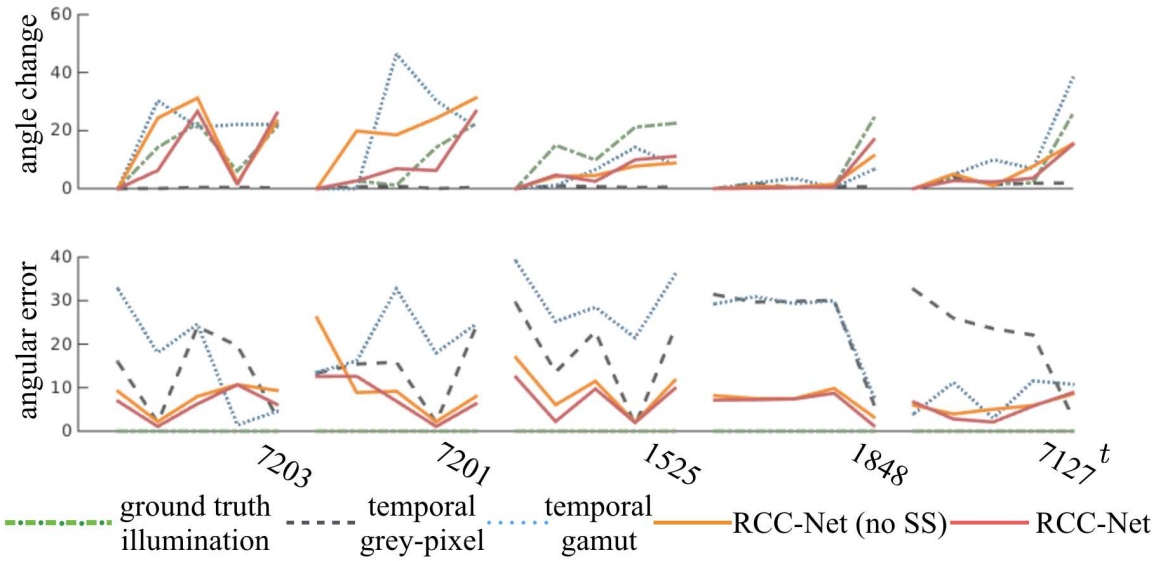

Figure 4. Five five-frame sub-sequences from linear Temporal SFU Gray Ball with significant illumination change. Rapid illumination angle change mainly occurs in the last frame (see the ground truth graphs in the top row) which no method handles well. Top: the *angle change* between two consecutive frames $\sphericalangle(c_{t-1}, c_t)$, t is the index marking the position of the frame in the original video. RCC-Net (no SS) is not equipped with sequence simulation.

We also evaluate robustness of the proposed RCC-Net against the cross-dataset setting: on the small Prinet Dataset released in [29], we can evaluate RCC-Net pretrained on the linear Gray Ball benchmark. Using leave-one-out cross-validation for video sequences recorded under normal light conditions (allows fine-tuning with 6 sequences out of 7), the RCC-Net achieved 5.0 degree mean error which is better than 5.4 degree reported in [29].

Moreover, we consider the effect of image resolution on illumination estimation to conducted an experiment on the non-linear data with 50% resolution. RCC-Net obtained 4.2 mean error and 3.0 median error, compared to 4.0 mean error and 2.9 median error on full-resolution images. This indicates that a lower resolution can slightly affect the performance of color constancy in a negative way.

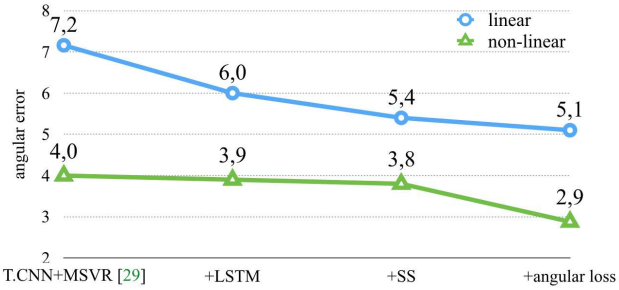

Figure 5. Median angular errors of the RCC-Net architecture with/without the proposed processing modules on the non-linear Temporal Gray Ball (green line) and the linear version (blue line). *+LSTM* is 5-frame-long RCC-Net without the simulated sequence *SS* and with the *MSE* loss function.

5.5. Ablation Study

In this section, we report how selection of the method parameters (strategies for image patch generation, subsequence length N and loss function) affect the performance of our RCC-Net. We experimentally evaluate the proposed method and report on both non-linear and linear Temporal SFU Gray Ball. The ablation study results are collected to Figure 5 which shows the progressive error reduction achieved by each module of our RCC-Net. Switching from CNN to Convolutional LSTM (+LSTM) predates the extended conventional deep model T.CNN+MSVR effectively on linear dataset, with a 17% lower median angular error. One explanation is that the preceding frames contributes to the illumination estimation of the shot frame. Error reduction can also be observed on the non-linear dataset, but not significant. It is also interesting to point out that SS and angular loss function benefit our method in the direction of adding spatial illumination consistency and alighted optimization.

Effect on Region Generation – In this experiment, we investigate three strategies for generating spatial regions as a sequence. The random patch is implemented by sampling $w_s = 5$ quarter-sized regions randomly while the spatial pyramid is constructed in two layers, *i.e.* top layer from the root level (full frame) and four non-overlapping sub-windows jointly covering the whole image. Table 2 shows that all strategies improve the performance by a noticeable margin on both datasets. The SS is comparable with or even

Table 2. Comparison of data augmentation procedures for the RCC-Net: RP – random patch, SP – spatial pyramid [22], SS – simulated sequence. Other settings: $N = 5$, angular loss function.

	SFU Gray Ball non-linear/linear			
	Med	Mean	90%	Max
No	3.2 / 6.3	4.5 / 7.7	13.6 / 23.7	**23.2** / 45.9
RP	3.0 / 4.9	4.3 / 7.2	13.8 / 23.6	24.3 / 49.8
SP	2.9 / **4.7**	4.3 / **7.1**	12.9 / 22.8	24.1 / 47.5
SS	**2.9** / 5.1	**4.1** / 7.2	**12.2** / **22.5**	25.2 / **45.7**

better than spatial pyramid, but significantly outperforms random patches. This observation can be explained by the fact that superior robustness can be achieved by discovering latent correlation across the overlapping spatial regions in the sequences.

Table 3. Performance of the RCC-Net with varying sub-sequence length N. Other settings: SS, angular loss function.

	SFU Gray Ball non-linear/linear			
N	Med	Mean	90%	Max
1	3.3 / 5.5	4.4 / 7.7	12.8 / 22.7	26.9 / 47.4
2	3.2 / 5.4	4.3 / 7.7	12.4 / **22.2**	26.8 / 47.6
5	**2.9** / **5.1**	**4.0** / **7.2**	**12.2** / 22.5	25.2 / **45.7**
10	2.9 / 5.2	4.0 / 7.5	**12.2** / 23.0	**23.4** / 47.5

Effect on Subsequence Length N – An important parameter of our method is the subsequence length N, *i.e.* how many frames processed as a sequence. Such a setting certainly depends on the viewfinder frame rate in digital cameras. From the results shown in Table 3, we found that $N = 5$ provides good accuracy and no significant improvement can be achieved with longer sequences. Our results agree with the observation that training on shorter video clips is a useful data augmentation strategy [12].

Table 4. Performance of the RCC-Net with the standard MSE and the proposed angular loss function ε with and without SS augmentation. The number of frames was set to $N = 5$.

	SFU Gray Ball non-linear/linear			
	Med	Mean	90%	Max
	without SS			
MSE	3.9 / 6.0	5.1 / 8.0	14.3 / 22.9	25.5 / 45.9
ε	3.2 / 6.3	4.5 / 7.7	13.6 / 23.7	**23.2** / 45.9
	with SS			
MSE	3.8 / 5.4	4.5 / 7.7	14.1 / 24.0	30.9 / 46.2
ε	**2.9** / **5.1**	**4.0** / **7.2**	**12.2** / **22.5**	25.2 / **45.7**

Effect on Loss Function – We test the effect of loss functions: the angular loss function vs. the MSE loss function. The results in Table 4 verify that the angular loss function is superior to the *MSE* cost function, especially when SS is given (25% improvement in median error for the non-linear dataset). This observation is consistent with the philosophy of deep learning – optimization on the objective directly boosts the performance.

6. Conclusion

In this paper, we formulate the temporal color constancy problem and propose the RCC-Net, a novel recurrent deep net, which consists of a convolutional LSTM, a novel simulated sequence component and a shallow network for merging. An ablation study confirms that all components of the RCC-Net improve performance.

On the non-linear and linear versions of the Temporal SFU Gray Ball Dataset, the RCC-Net achieves state-of-the-art performance – 2.9 and 5.1 median angular error respectively, outperforming the single-frame methods and their temporal variants by 14∼22%. The RCC-Net is very fast in inference on a GPU, *e.g.* illumination for a frame in a five-frame sequence is estimated in less than 50 ms on a Nvidia K40C GPU.

Acknowledgements

This work was funded by the Academy of Finland Grants No. 267581 and 26980 and the Technology Agency of the Czech Republic project TE01020415 (V3C – Visual Computing Competence Center). The authors wish to acknowledge Intel Finland for generous research resources and CSC - IT Center for Science Finland for computational resources.

References

[1] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–984, 2002. 1, 2, 6

[2] K. Barnard, L. Martin, B. Funt, and A. Coath. A data set for color research. *Color Research & Application*, 27(3):147–151, 2002. 6

[3] J. T. Barron. Convolutional color constancy. In *ICCV*, 2015. 2, 3

[4] J. T. Barron and Y.-T. Tsai. Fast fourier color constancy. In *CVPR*, 2017. 2, 3

[5] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recognition*, 43(3):695–705, 2010. 6

[6] S. Bianco, C. Cusano, and R. Schettini. Color constancy using cnns. In *CVPR workshop*, 2015. 2, 3, 5

[7] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26, 1980. 1, 2, 6

[8] V. C. Cardei, B. Funt, and K. Barnard. White point estimation for uncalibrated images. In *Color Imaging Conference (CIC)*, 1999. 2

[9] A. Chakrabarti, K. Hirakawa, and T. Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1509–1519, 2012. 1, 2

[10] D. Cheng, B. Price, S. Cohen, and M. S. Brown. Effective learning-based illuminant estimation using simple features. In *CVPR*, 2015. 2, 5, 6

[11] F. Ciurea and B. Funt. A large image database for color constancy research. In *Color Imaging Conference (CIC)*, 2003. 2, 3, 6

[12] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2, 3, 8

[13] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221, 2001. 1, 2

[14] G. D. Finlayson, S. D. Hordley, and P. Morovic. Colour constancy using the chromagenic constraint. In *CVPR*. IEEE, 2005. 1

[15] B. Funt and W. Xiong. Estimating illumination chromaticity via support vector regression. In *Color Imaging Conference (CIC)*, 2004. 2, 6

[16] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. 6

[17] A. Gijsenij. Color constancy research website: `http://colorconstancy.com`. 3, 6

[18] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):687–698, 2011. 1, 2, 6

[19] A. Gijsenij, T. Gevers, and J. Van De Weijer. Generalized gamut mapping using image derivative structures for color constancy. *International Journal of Computer Vision*, 86(2-3):127–139, 2010. 1, 2, 5, 6

[20] A. Gijsenij, T. Gevers, and J. Van De Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011. 2

[21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 4

[22] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 5, 8

[23] M. Henaff, A. Szlam, and Y. LeCun. Orthogonal rnns and long-memory tasks. *arXiv preprint arXiv:1602.06662*, 2016. 4

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3, 4

[25] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *CVPR*, 2016. 3

[26] H. R. V. Joze and M. S. Drew. Exemplar-based color constancy and multiple illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):860–873, 2014. 2, 6

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[28] Z. Lou, T. Gevers, N. Hu, M. P. Lucassen, et al. Color constancy by deep learning. In *BMVC*, 2015. 2

[29] V. Prinet, D. Lischinski, and M. Werman. Illuminant chromaticity from image sequences. In *ICCV*, 2013. 2, 3, 6, 7

[30] Y. Qian, K. Chen, J. Kämäräinen, J. Nikkanen, and J. Matas. Deep structured-output regression learning for computational color constancy. In *ICPR*, 2016. 2, 3, 4, 5

[31] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005. 1

[32] J.-P. Renno, D. Makris, T. Ellis, and G. A. Jones. Application and evaluation of colour constancy in visual surveillance. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005. 5

[33] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. 3

[34] L. Shi and B. Funt. Re-processed version of the gehler color constancy dataset of 568 images. *accessed from* `http://www.cs.sfu.ca/~colour/data/`. 6

[35] W. Shi, C. Change Loy, and X. Tang. Deep specialized network for illuminant estimation. In *ECCV*, 2016. 2, 5

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 4

[37] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 3

[38] T. Tieleman and G. E. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In *Coursera Lecture slides:* `https://www.coursera.org/learn/neural-networks`, 2012. 5

[39] J. Van De Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007. 1, 2, 6

[40] J. Van De Weijer, C. Schmid, and J. Verbeek. Using high-level visual information for color constancy. In *ICCV*, 2007. 2, 6

[41] J. von Kries. Influence of adaptation on the effects produced by luminous stimuli. *Source of Color Science*, pages 109–119, 1970. 4

[42] N. Wang, B. Funt, C. Lang, and D. Xu. Video-based illumination estimation. In *Color Imaging Conference (CIC) Workshop*, 2011. 2, 3, 5, 6

[43] L. Wu, C. Shen, and A. V. D. Hengel. Deep recurrent convolutional networks for video-based person re-identification: An end-to-end approach. *arXiv preprint arXiv:1606.01609*, 2016. 3

[44] K.-F. Yang, S.-B. Gao, and Y.-J. Li. Efficient illuminant estimation for color constancy using grey pixels. In *CVPR*, 2015. 1, 2, 5, 6

[45] Q. Yang, S. Wang, N. Ahuja, and R. Yang. A uniform framework for estimating illumination chromaticity, correspondence, and specular reflection. *IEEE Transactions on Image Processing*, 20(1):53–63, 2011. 2, 3