Papers on Color Grading

Paper 1
Example-Based Video Color Grading - 2013 (ACM Transactions on Graphics (TOG))

2 stage approach -

1. Compute per-frame transformations **T**t that match color palette of each input video frame to a corresponding representative model frame
2. Filter the per-frame transformations to make them temporally coherent. (to prevent temporal artifacts)
    a. This is achieved by a differential-geometry analysis of the series of color transformations **T** as a curve in a higher-dimensional transform space.
    b. Points at which this curve has a high curvature directly corresponds to instants where directly applying the transformations **T**t to the input video **I**t would produce artifacts such as temporal flickering.
    c. Instead, detect points on this curve that have a low curvature (**keyframes)** and interpolate the transformations at these points to build a set of temporally coherent color transformations **T'**.
    d. Apply the interpolated transformations **T'**t to the input video frames **I**t produces the final color graded output video frames **O**t.

Color Transfer Model -

Transfer the color distribution of a model video frame **M**t to an input video frame **I**t.
Transfer the luminance values using a smooth remapping curve and then matching the chrominance values using three affine transforms for -

1. Shadows
2. Mid-tones
3. Highlights

Cases where background and foreground segmentation is provided then this 3 transformation is applied to both background and foreground. So a total of 6 transformations per frame.
Finally apply an edge-aware smoothing filter that ensures clean edges with no halos.

Representative model frames

Summarize the model sequence with a few representative frames whose color distribution is representative of that of the entire video.
Perform this clustering using the K-medoids clustering technique. (eg- select 1 frame for 30 frames i.e. 1 frame for every second of the video sampled at 30fps)

Differential Color Transform Filtering

We have a color transform **T**t for each frame. (each of which has been computed in isolation and applying them directly to the video frames produces a temporally inconsistent result.)
Temporally filter out the color transforms applied to video frames. (ensuring that both the temporal coherence observed in the input sequence and the spatial details in the video frames are preserved.)
Introduce a smoothing filter that operates on color transforms. (based on curvature flow filter on surfaces in 3D)

1. Extend the notion of curvature to the color transforms previously computed
2. Find points of low curvature and select them as keyframes.
3. Interpolate the color transforms at the detected keyframes to compute a temporally smooth set of color transformations for the entire sequence.

(This replaces the high-curvature points that lie between the keyframes with transforms interpolated from the neighbouring keyframes)

Estimate the curvature
Approximate curvature flow

Paper 2
Intrinsic Video and Applications - ACM Transactions on Graphics (TOG) (2014)

Paper 3
Automatic Photo Adjustment Using Deep Neural Networks

Exemplar-based photo adjustment as a regression problem.

Pair images {I, J}, premise that a function F maps each pixel in I to J.

F(Θ, x) - using non-linear regression.
c = [L a b] V(c) = [L a b 1] if a 3x4 affine color transform is used.
F = ϕ(Θ, x).V(c)
They use 3x10 V(c) = [L2 a2 b2 La Lb ab L a b 1]

$$\arg\min_{\Phi\in\mathcal{H}} \sum_{i}^{n} \| \Phi(\Theta, x_i)V(c_i) - y_i \|^2,$$

ϕ here is a DNN with multiple hidden layers.

For a single neuron **v**

$$v_j^l = g\left(w_{j0}^l + \sum_{k>0} w_{jk}^l v_k^{l-1}\right)$$

Feature Descriptor -
x = (xp, xc, xg)
xp = pixel wise features
xc = contextual features
xg = global features

Pixel wise Features -
Learning spatially varying photos enhancement models.
xp = (c,p)
c = average color in the CIELab color space within the 3x3 neighbourhood.
p = (x,y) the normalized sample position within the image.

Global Features -

6 types of global features

1. Intensity Distribution
2. Scene Brightness
3. Equalization curves
4. Detail Weighted Equalization Curves
5. Highlight Clipping
6. Spatial Distribution

This gives a **207** dimensional vector.

Contextual Features -
Characterize the distribution of semantic categories - sky, building, car, person, tree, etc.
Run image segmentation. Get a semantic label map for the entire input image.
See sec 4.3 in paper.

Datasets -
MIT-Adobe 5K Dataset
Instagram Dataset

Paper 4 -
Neural Color Transfer Between Images (2018 ArXiv) (MSR)

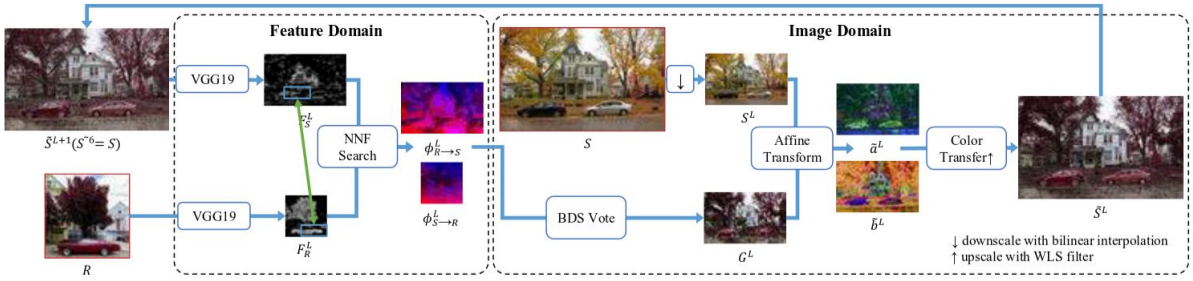
Algorithm for color transfer between images that have perceptually similar semantic structures.
Can be extended from “one-to_one” to “one-to-many” color transfers.
Uses a Markov Random Field optimization to obtain the piecewise smooth NNF.

1. Neural color transfer method,
    a. Matching in deep feature domain
    b. Local color transfer in image domain
2. Pixel-granular linear function, avoiding local structural distortions and global incoherency by enforcing both local and global constraints.
3. Extend ‘one-to-one’ to ‘one-to-many’ - avoids content mismatching between images.

Local color transfer, Deep color transfer, Multi-reference color transfer, colorization.

Method -
Use features from a pre-trained CNN.
Build correspondences between the source images S and the reference image R. Select features from different layers in the CNN.
Apply local color transform in the image domain.

Nearest Neighbor Field Search
Build bi-directional correspondences between S and R.
Perform NNF in the deep feature domain.



Pass S^{L+1} and R into the VGG19 network and get corresponding feature maps in reluL_1 layer - F_s^L and F_R^L respectively.
Each feature map is a 3D tensor with widthxheightxchannel and it’s spatial resolution is $1/2^{L-1}$

Mapping function $\phi^L_{S->R}$ from Fs to Fr is given by

$$\phi^L_{S\to R}(p) = \arg\min_q \sum_{x\in N(p), y\in N(q)} (||\overline{F}^L_S(x) - \overline{F}^L_R(y)||^2$$

Use relative values for feature map patches.
So we compute mappings from S->R and R->S in a similar fashion.

BIdirectional correspondences allow us to use BDS voting - Bidirectional Similarity voting.
Construct guidance image G_L

Local Color Transfer Algorithm
Downscale S to S^L to match the dimensions of G^L

$$\mathcal{T}_p^L(S^L(p)) = a^L(p) \times S^L(p) + b^L(p).$$

We formulate the problem of estimating $\mathcal{T}^L$ by minimizing the following objective function consisting of three terms:

$$E(\mathcal{T}^L) = \sum_p E_d(p) + \lambda_l \sum_p E_l(p) + \lambda_{nl} \sum_p E_{nl}(p), \quad (3)$$

$$E_d(p) = (1 - \bar{e}^L(p)) \|\mathcal{T}_p^L(S^L(p)) - G^L(p)\|^2, \quad (4)$$

$$E_{nl}(p) = \sum_{q \in K(p)} \omega(p,q) \|\mathcal{T}_p^L(S^L(p)) - \mathcal{T}_q^L(S^L(q))\|^2, \quad (6)$$

Main Idea- Extract local correspondences in the feature map space.

Paper 5 -
Exemplar Based Image and Video Stylization Using Fully Convolutional Semantic Features
(IEEE Transactions on Image Processing 2017)

Color and tone stylization in images and videos to enhance unique themes with artistic color and tone adjustments.

A Novel Deep Learning architecture for stylistic image enhancement.

1. Fully convolutional networks extracts global features and contextual features. This has 2 parts -
    a. Semantics aware feature extracted from deep layers of a fully convolutional network.
    b. Second part consists of a set of color histograms over a small spatial grid.

Method

$\Phi(\Theta, x_i)$ maps the pixel at p_i before adjustment, c_i to it's corresponding color y_i after adjustment.

Model parameters are learnt by the following objective function minimization.

$$\arg\min_{\theta} \sum_{i} ||\phi(\theta, x_i)V(c_i) - y_i||^2 \qquad (1)$$

After Superpixilization -

Thus, the above objective function is revised as follows.

$$\arg\min_{\theta} \sum_{v} \sum_{j \in s_v} ||\phi(\theta, x_v)V(c_j) - y_j||^2. \qquad (2)$$

Ideas -

Approach 1 -

1. Get a mapping between moods/emotions and video snippets. (from movies)
    a. This can be done using sentiment analysis on text - subtitles and scripts.
    b. Then we associate a video clip with it's sentiment or mood.
2. Enter text to get a set of retrieved candidates.
3. Apply color grading of top k retrieved candidates to the input image.

Approach 2 -

Not very clear. Need more insight and discussion.

1. Learn a form of joint embedding where the textual features also capture the tone/mood from training.
2. Then we apply some efficient color grading.

Discussion -

What is the data we have?

1. Text
2. Images
3. Videos

What more do we need?

Methods/pipelines which may be used?