https://doi.org/10.2352/ISSN.2470-1173.2018.05.PMII-442

An Automatic Tuning Method for Camera Denoising and Sharpening based on a Perception Model

***Weijuan Xi*[1], *Huan Zeng*[2], *Jonathan B. Phillips*[2]**
[1] ***Electronic Imaging Systems Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN-47906, U.S.A., xi9@purdue.edu***
[2] ***Google Inc., Mountain View, CA-94043, U.S.A., hzeng, jbphillips@google.com***

Abstract

Camera denoising and sharpening parameters are device related rigid parameters which are programmed in phone camera device. The current tuning method depends solely on manual modulation and visual evaluation of image quality, which is time consuming and difficult to optimally achieve. To this end, we will introduce an automatic tuning method for mobile cameras in this paper, which can tune the WNR parameters automatically and produce high quality images within a feasible processing time. The method contains two parts, a perception model and an optimization algorithm. For the first part, we developed a perception model to evaluate the image quality for mobile cameras through modified CPIQ metrics. For the second part, in order to overcome a high-dimension non-convex optimization problem, we developed a searching strategy to find the optimal solution by conducting quantization and iteratively minimizing the error metric of the perception model.

Introduction

With the development of mobile phones, the mobile camera has become a very important part of our daily life and higher image quality is required in more and more aspects of life. Camera tuning is a very important part of camera image processing and can influence the image quality dramatically. However, camera tuning is currently based completely on manual tuning and the image quality can only be checked visually. As a result, camera tuning takes tremendous amount of time, usually several weeks, and in most cases only approximately optimal parameters can be find instead of the global optimal parameters. Additionally, in order to get high quality images from mobile cameras, there are more than ten (depending on camera model) image processing blocks which can be unequally influenced by tuning parameters. This makes it difficult to achieve optimal tuning parameters and takes an unpredictable amount of tuning time. To our best knowledge, there are several patents that discuss manual tuning methods, but no articles discuss automatic tuning. Therefore, an automatic tuning method, which can tune camera in a short time, is very important and urgently needed.

In this paper, we will focus on the Wavelet Noise Reduction (WNR) block in the camera image processing pipeline and introduce our automatic tuning method. Based on the experiment of manual tuning, we divide this problem into three different parts. First, we developed a new block diagram for an automatic tuning method based on the principle of image quality and the experience of manual camera tuning. Second, we generate a perception model through modifying the metrics in Camera Phone Image Quality (CPIQ) standard in order to evaluate the image quality. Third, the high-dimensional space formed by tuning parameters is a non-convex space since the tuning parameters are not independent and have high cross-correlation with each other. Therefore, there are many local minimums in the high-dimension space and difficult to find the optimal parameters. In this situation, the traditional optimization methods fail to get optimal parameters in a short time. Finding optimal parameters in a short time is a great challenge for automatic tuning. In the the tuning process, the appropriate metrics for WNR are very important and affect the tuning result dramatically. For a reasonable perception model, it should reflect the image quality quantitatively with a small error metric. A smaller error metric should correspond to better image quality and vice versa.

Preliminary: Dead Leaves Target

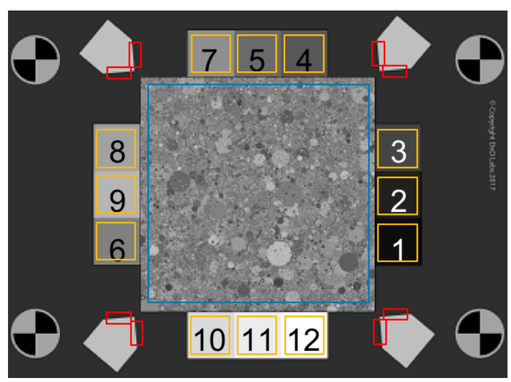


Figure 1. *Dead leaves target. There are eight slanted edges in the eight red boxes which are the regions of interest (ROIs) for the slanted-edge SFR measurements. The twelve square gray patches in the twelve yellow boxes are the ROIs for visual noise measurement and also be used to conduct the tone reproduction curve linearization. The dead leaves chart in the middle blue box is the ROI for the texture acutance measurement.*

Cao, Guiehard, and Hornung [1] described the utilization of the ”dead leaves” target for measurement of system spatial frequency response in the context of texture detail. This target consists of a series of overlapping circles with a uniform distribution of gray levels, with radii(r) distributed approximately as $1/r^3$ [2]. The power law distribution suggests near scale-invariance of the target, so that in principle actions such as scaling, cropping, or rotation should not change the power spectral density.

Unlike the traditional knife-edge target, the spatial fruquency response (SFR) derived from the dead leaves target will be penalized for systems that employ aggressive noise reduction. Besides, the dead leaves SFR correlates well with sharpness/texture blur preference, and thus the target can potentially be used as a surrogate for more expensive subjective image quality evaluations.

Figure 1 shows the image chart that we used in this work. There are eight slanted edges in the eight red boxes which are the regions of interest (ROIs) for the slanted-edge SFR measurements. The twelve square gray patches in the twelve yellow boxes are the ROIs for visual noise measurement and also be used to conduct the tone reproduction curve linearization. Besides, the dead leaves chart in the middle blue box is the ROI for the texture acutance measurement.

Image Quality Standard

The CPIQ standard is a specific standard for camera phone image quality, it includes seven metrics: slanted-edge spatial frequency response (eSFR), lateral chromatic displacement, chroma level, color uniformity, local geometric distortion, visual noise (VN) and texture blur/acutance (TA). Similarly, the International Organization for Standardization (ISO) 15739 standard also contains several metrics to evaluate image quality. Because the parameters of WNR block affect the noise and sharpness of images, in this paper, we generate a perception model for the WNR block based on eSFR, VN and TA, since all three metrics are related to image noise and sharpness. In this section, we only introduce the process to generate VN and TA metrics in CPIQ standard.

A. Visual Noise Metric

The ISO [3] VN standard is the starting point of CPIQ VN standard as it is a preexisting standard, and the frequency-based spatial filtering allows multiple frequency filters to be easily cascaded. The steps to generate VN metric in ISO standard is shown in table 1.

Table 1: Steps to generate VN metric in ISO standard

ISO	STEP
B.2.1	RGB to XYZ(E)
B.2.2	XYZ(E) into opponent space AC1C2
B.2.3	Discrete Fourier Transform
B.2.4	Apply the contrast sensitivity function (CSF)
B.2.5	Inverse Fourier Transform
B.2.6	Opponent space AC1C2 into XYZ(E)
B.2.7	XYZ(E) to XYZ(D65)
B.2.8	XYZ(D65) to $L^*U^*V^*$
B.2.9	Standard deviation for each gray patch
B.2.10	Weighted sum representing the visual noise

Table 2 shows the detailed steps to generate the VN metric in the CPIQ standard. Compared with traditional noise metrics, such as signal-to-noise ratio, the VN metric of the CPIQ standard considers the frequency of noise which enables a better representation on the noise intensity in the human visual system. For the CPIQ VN metric, we should transfer the color image from RGB color space to AC_1C_2 color space, then conduct the Discrete Fourier Transform (DFT) to get the frequency signal. After that, three filters in frequency domain are cascaded to process the image: 1). the CSF (Contrast Sensitivity Function) filter, 2). the display MTF (Modulation Transfer Function) filter, and 3). the high pass filter. The CSFs are calculated as

Table 2: Steps to generate VN metric in CPIQ standard

CPIQ	step
D.1-D.3	RGB to XYZ(E)
D.4	XYZ(E) into opponent space AC1C2
D.5	Discrete Fourier Transform
D.6	Apply the contrast sensitivity function (CSF)
D.7	Apply Display MTF
D.8	High Pass Filter (HPF)
D.9	Inverse Fourier Transform
D.10-D.11	Opponent space AC1C2 into XYZ(E)
D.12	XYZ(D65) to $L^*a^*b^*$(CIE Lab)
D.13	Objective noise (CPIQ)

$$CSF_{lum} = \frac{a_1 \times f^{c_1} \times exp(-b_1 \times f)}{K}, \tag{1}$$

$$CSF_{chrom} = \frac{a_1 \times exp(-b_1 \times f^{c_1}) + a_2 \times exp(-b_2 \times f^{c_2}) - S}{K}, \tag{2}$$

where, CSF_{lum} is the CSF for luminance channel, CSF_{chrom} are the chromininance CSF for C_1 and C_2 channels. The parameter values in these formulas are shown in Table 3.

Table 3: Parameter values in Contrast Sensitivity Function (CSF)

param	luminance CSF_{lum}	Red-Green Chrominance CSF	Blue-Yellow Chrominance CSF
a_1	75	109.1413	7.0328
b_1	0.2	0.0004	0
c_1	0.8	3.4244	4.2582
a_2		93.5971	40.691
b_2		0.0037	0.1039
c_2		2.1677	1.6487
K	75	202.7384	40.691
S		0	7.0328

The display MTF depends on the viewing condition, and in this paper we choose the computer monitor viewing condition: 100% at 100 ppi, on the basis of the CPIQ standard, the display MTF can be calculated as

$$M_{disp}(v) = \| \frac{sin(\pi k_{disp} v)}{\pi k_{disp} v} \|, \tag{3}$$

where, $k_{disp} = 0.0243$ is a parameter depending on the viewing condition. After another color space transformation from AC_1C_2

to $L^*a^*b^*$ color space, the objective VN in the CPIQ standard is calculated in $L^*a^*b^*$ color space as

$$\begin{aligned}\Omega =&log10[1+23.0\cdot\sigma^2(L^*)+4.24\cdot\sigma^2(a^*)-5.47\cdot\sigma^2(b^*)\\&+4.77\cdot\sigma^2(L^*a^*)],\end{aligned} \tag{4}$$

B. Texture Acutance

Table 4: Desired reflectance of 12 gray patches

patch number	12	11	10	9	8	7
patch reflectance	1	0.927	0.854	0.708	0.635	0.562
patch number	6	5	4	3	2	1
patch reflectance	0.5	0.416	0.343	0.27	0.124	0.051

In the CPIQ standard, there are four steps to generate the texture acutance metric.
Step 1: Generate the tone reproduction curve (TRC) based on 12 gray patches in the 12 yellow boxes shown in Fig. 1, then, conduct the linearization with the TRC. The desired patch reflectance is shown in Table 4.
Step 2: compute the luminance. In this step, we converge the color dead leaves target into gray image and the gray level can be calculated as

$$L = 0.21260\times R+0.71523\times G+0.07220\times B, \tag{5}$$

where, R,G,B are the values of red, green and blue channels respectively.
Step 3: Get the texture MTF for TA. First, the 1-D DFT is calculated by the formula

$$U(m,n) = \| \sum_{x=-\frac{N}{2}+1}^{\frac{N}{2}} \sum_{y=-\frac{N}{2}+1}^{\frac{N}{2}} I(x,y)e^{\frac{2i\pi(mx+ny)}{N}} \|^2, \tag{6}$$

where, $I(x,y)$ is the pixel value of the gray dead leaves target at (x,y). Then, the arithmetic mean of the 2D-FFT over all possible directions is computed which results in a one dimensional profile U_{1D}.

Now, we already get the frequency response of dead leaves target by the step 1 - 3. In order to ensure the accuracy of the texture MTF, compensation for noise must be factored in. according to the CPIQ standard, regarding the 1-D MTF of No.6 uniform gray patch as the NPS (Noise Power Spectrum), the frequency response of the dead leaves target can be expressed as

$$U'_{1D}(r) = U_{1D}(r) - p^2H(\frac{r}{p}), \tag{7}$$

Finally, through normalizing the U'_{1D} by the ideal chart power spectrum, the texture MTF for texture acutance can be obtained by

$$MTF_{1D}(r) = \sqrt{\frac{U_{1D}(r)}{T(r)}}, \tag{8}$$

where, $T(r) = \frac{c(N)}{r^\eta}$ is the power spectrum density of the theoretical chart; Both η and $c(N)$ are constants, which only depend on the target design and the crop size of the dead leaves target. Specifically, let the crop size of dead leaves target be N, then we have $c(N) = \frac{var}{\sum_{-\frac{N}{2}+1,\dots,\frac{N}{2}}}$.
Step 4: Texture MTF curve fitting. This is a very important step for TA since the texture MTF that was acquired from step 3 is very noisy. In order to resolve this, the MTF is fitted with a cubic Hermit spline in CPIQ standard. This spline has six control points and is defined by the 18 values shown in below:

- The X-axis values of each control point, written as $[x_0,x_1,x_2,x_3,x_4,x_5]$, is set to $[0,0.1,0.2,0.3,0.4,0.5]$
- The Y-axis values (MTF value) of each control point are written as $[y_0,y_1,y_2,y_3,y_4,y_5]$ with $y_0 = 1$
- The first derivative of the spline at each control point are written as $[d_0,d_1,d_2,d_3,d_4,d_5]$ with $d_0 = 0$ and $d_5 = 0$

Since the spline is a continuous function, with a continuous first derivative, it is uniquely defined by the 14 parameters $[x_0,x_1,...,x_5]$, $[y_0,y_1,...,y_5]$, d_0 and d_5. For a random value x between the interval $[x_k,x_{k+1}]$, the value of spline is

$$\begin{aligned}MTF_{fit}(x) =&h_{00}(t)\cdot y_k + h_{10}(t)\cdot d_k + h_{01}(t)\cdot y_{k+1}+\\&h_{11}(t)\cdot(x_{k+1}-x_k)\cdot d_{k+1},\end{aligned} \tag{9}$$

where, $t = (x-x_k)/(x_{k+1}-x_k)$ and Hermit basis function
$h_{00}(t) = 2t^3-3t^2+1$,
$h_{10}(t) = t^3-2t^2+t$,
$h_{01}(t) = -2t^3+3t^2$,
$h_{11}(t) = t^3-t^2$,

We can get the final texture MTF with the following constraint condition
$MTF_{fit}(0) = y_0 = 1$,
$MTF'_{fit_+}(0) = d_0 = 0$,
$MTF'_{fit_-}(0.1) = MTF'_{fit_+}(0.1)$,
$MTF'_{fit_-}(0.2) = MTF'_{fit_+}(0.2)$,
$MTF'_{fit_-}(0.3) = MTF'_{fit_+}(0.3)$,
$MTF'_{fit_-}(0.4) = MTF'_{fit_+}(0.4)$,
$MTF'_{fit_-}(0.5) = d_5 = 0$,

Figure 2 shows examples of texture MTF results under different quantities of illumination.

Automatic Tuning Method

The block diagram of Automatic Tuning Method is shown in Fig. 3. Instead of the mobile camera, we use a Qualcomm simulator which can simulate the camera image processing in a mobile camera device. There are two main parts in the automatic tuning block diagram: 1). Generation of the perception model; 2). The optimization process. The perception model is used to evaluate the image quality,which can output an error metric. For a reasonable perception model, smaller error metric corresponds to a higher image quality. For the second part, the searching/optimization method is an iterative algorithm. We will introduce the implementation details for the two parts in this section.

A. Generate Perception Model

Since the CPIQ standard is a specific image quality standard for mobile cameras and it already implements subjective metrics and a perception model, our perception model is based on the CPIQ standard. However, the VN metric in the CPIQ standard only applies to the visual noise with a Gaussian distribution. As a result of that, the CPIQ NV metric has two problems for working directly as a part of the perception model: 1) the VN metric is less sensitive to sharp impulse noise since it always calculates the mean noise of all pixels; 2) the VN metric does not have the ability to evaluate large chroma noise which can be recognized through the minus before the $\sigma^2(b^*)$ item in formula (4). Figure 4 shows the examples of the two kinds of visual noise, pattern a) is a manually tuned result which has less noise in our eyes than pattern b) and c), but both (b) and (c) may correspond to smaller error values in the CPIQ VN metric calculated by the formula (4). Therefore, the update for the CPIQ VN metric is necessary.

The basic idea to update the VN metric is to only focus on the more noisy regions instead of all the pixels in the uniform gray pattern, through this, the VN metric will be more sensitive to the impulse sharp noise. Additionally, the error caused by chroma noise can be reduced by combining the VN metrics in the ISO and CPIQ standards together, since the ISO VN metric has the ability to evaluate the chroma noise by positive summing the noise in both the luminance and the two chroma channels. The formula of the objective ISO VN metric is

$$ISO_{VN} = \sigma_{L^*} + 0.852\sigma_{U^*} + 0.323\sigma_{V^*}, \tag{10}$$

where, ISO_{VN} is calculated in $L^*U^*V^*$ color space, σ_{L^*}, σ_{U^*} and σ_{V^*} are the standard deviations in the L^*, U^* and V^* channels respectively.

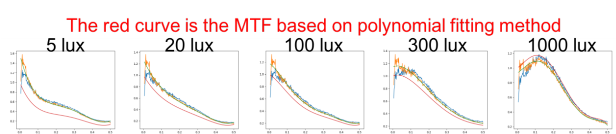


Figure 2. *Texture MTF of CPIQ standard: The red smooth curve in each graph is the final texture MTF after curve fitting and the coarse texture MTF curves are also showed in these graphs by other colors.*

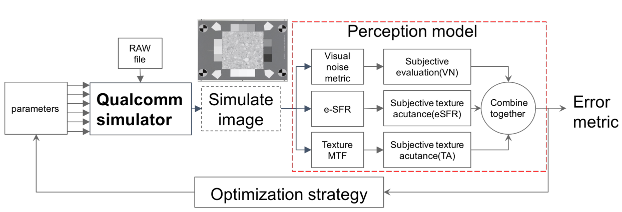
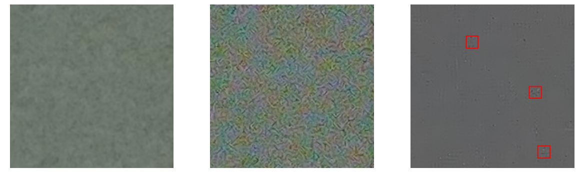


Figure 3. *Block diagram of automatic tuning method.*



(a) Manual tuning gray pattern (b) Large chroma noise (c) sharp impulse noise

Figure 4. *Examples of chroma noise and impulse sharp noise.* ***(a)****: is the manual tuning result of high gain (low light density) image, which has less visual noise compare with* ***(b)*** *and* ***(c)*** *patterns;* ***(b)****: has large chroma noise but corresponds to a small VN value in CPIQ standard;* ***(c)****: contains sharp noise which can't be reflected accurately in CPIQ VN metric.*

In summary, the steps to update VN is shown as follows:
First, transfer the image to $L^*U^*V^*$ color space in order to combine with ISO VN metric

$$\begin{aligned} L^* &= \begin{cases} (\frac{29}{3})^3 Y/Y_n & Y/Y_n \le (\frac{6}{29})^3 \\ 116(Y/Y_n)^{\frac{1}{3}} - 16 & Y/Y_n > (\frac{6}{29})^3 \end{cases}, \\ u^* &= 13L^* \cdot (u' - u'_n), \\ v^* &= 13L^* \cdot (v' - v'_n), \end{aligned} \tag{11}$$

where, $u' = \frac{4X}{X+15Y+3Z}$ and $v' = \frac{9Y}{X+15Y+3Z}$.
Second, extract the largest errors in each of the three channels

$$\begin{cases} \varepsilon_{L^*} = L^* - \mu_{L^*}, & \varepsilon_{L^*_{max}} = max(\|\varepsilon_{L^*}\|) \\ \varepsilon_{U^*} = U^* - \mu_{U^*}, & \varepsilon_{U^*_{max}} = max(\|\varepsilon_{U^*}\|) \\ \varepsilon_{V^*} = V^* - \mu_{V^*}, & \varepsilon_{V^*_{max}} = max(\|\varepsilon_{V^*}\|) \end{cases}, \tag{12}$$

where, μ_{L^*}, μ_{U^*}, μ_{V^*} are the mean values and ε^*_L, ε^*_U, ε^*_V are the noise values in the L^*, U^*, V^* channels respectively; $\varepsilon^*_{L_{max}}$, $\varepsilon^*_{U_{max}}$, $\varepsilon^*_{V_{max}}$ indicate the largest noise values in the three channels.
Third, regard the half of $\varepsilon^*_{L_{max}}$, $\varepsilon^*_{U_{max}}$, $\varepsilon^*_{V_{max}}$ as three threshold values in the channels, and only focus on the pixels whose noise is larger than the threshold value

$$\begin{cases} \sigma_{L^*} = \sqrt{\frac{1}{n_{L^*}}(\sum_{\|\varepsilon_{L^*} \ge \frac{1}{2}\varepsilon_{L^*_{max}}\|} \varepsilon^2_{L^*})}, & n_{L^*} = size(\|\varepsilon_{L^*} \ge \frac{1}{2}\varepsilon_{L^*_{max}}\|) \\ \sigma_{U^*} = \sqrt{\frac{1}{n_{U^*}}(\sum_{\|\varepsilon_{U^*} \ge \frac{1}{2}\varepsilon_{U^*_{max}}\|} \varepsilon^2_{U^*})}, & n_{U^*} = size(\|\varepsilon_{U^*} \ge \frac{1}{2}\varepsilon_{U^*_{max}}\|) \\ \sigma_{V^*} = \sqrt{\frac{1}{n_{V^*}}(\sum_{\|\varepsilon_{V^*} \ge \frac{1}{2}\varepsilon_{V^*_{max}}\|} \varepsilon^2_{V^*})}, & n_{V^*} = size(\|\varepsilon_{V^*} \ge \frac{1}{2}\varepsilon_{V^*_{max}}\|) \end{cases}, \tag{13}$$

Finally, we get the updated ISO VN metric by formula (10). Since, the CPIQ standard already has a subjective VN metric, the conversion from ISO VN to CPIQ VN is conducted in order to take advantage of the CPIQ perception model, and the mapping of data between the ISO VN and CPIQ VN is shown in Table 5. According to the data in Table 5, it's reasonable for us to assume that there is a linear relationship between CPIQ VN and ISO VN. As a result, we can get the mapping formula

$$Q_{VN} = CPIQ_{VN} = 0.2978 \cdot ISO_{VN} + 0.2436, \tag{14}$$

In the CPIQ standard, the subjective quality loss for visual noise is calculated as

$$QL_{VN} = \begin{cases} 0 & Q_{VN} \le 0.319 \\ \frac{Q_{VN}-a}{b} - \frac{c \times (1 + b \times \frac{Q_{VN}-a}{c})}{b^2} & Q_{VN} > 0.319 \end{cases}, \tag{15}$$

Table 5: mapping data between ISO and CPIQ VN metrics

ISO objective visual noise	CPIQ objective visual noise	ISO objective visual noise	CPIQ objective visual noise
2.1095	0.6129	3.481	1.6293
1.3535	0.3627	2.1633	1.2587
1.1376	0.5107	9.3344	2.071
0.7297	0.2831	5.8191	1.6754
4.2659	1.3252	4.8333	1.9011
2.7734	1.0016	2.9023	1.4957
2.1395	1.1681	9.8664	2.1163
1.2531	0.7692	3.9274	1.3486
7.239	1.8486	1.8468	0.761
4.6111	1.4794	0.9046	0.3336

Similarly, the subjective quality losses for eSFR and TA is expressed as

$$QL = \frac{0.00336 - 2.34B + 164B^2 - 192B^3 + 16.3B^4}{1 - 0.0866B + 0.968B^2 - 2.31B^3}, \quad (16)$$

where

$$B = \begin{cases} 0.886 - Q & Q \leq 0.886 \\ 0 & Q > 0.886 \end{cases}, \quad (17)$$

and Q can be slanted-edge SFR or texture acutance getting from CPIQ standard.

The CPIQ standard also provides a perception model which contains seven metrics, we borrow this model but only use three metrics (VN, eSFR, TA) which are relate with image noise and sharpness, so that the final perception model can be expressed as

$$QL = [(QL_{VN})^{n_{max}} + (QL_{SFR})^{n_{max}} + (QL_{TA})^{n_{max}}]^{1/n_{max}}, \quad (18)$$
$$n_{max} = max(QL_{VN}, QL_{SFR}, QL_{TA}),$$

where, QL_{VN}, QL_{SFR}, QL_{TA} are the metrics for updated VN, eSFR and TA.

B. Optimization Process

With the perception model, which evaluates the image quality, already generated, the next important step is to develop an optimization method to find the optimal tuning parameters. Traditional optimization methods fail for the automatic tuning problem since there are infinite local minimums. Besides, the simulator, which spends dozens of seconds to generate a single image, has a strong impact on optimization efficiency. As a result of that, two problems are encountered if we use the traditional optimization methods (e.g., Gradient Descent): 1). It's difficult for us to search the global minimum because of infinite local minimums. 2). The optimization process costs too much time caused by the large calculation and slow simulator. A special optimization method must be generated which can find the optimal parameters in a short time.

For camera tuning, different tuning parameters will generate different images. However, a small change of one parameter can only affect the image quality slightly and it is difficult for human eyes to distinguish the difference. This motivated us to quantization of the tuning parameters. The following are the detailed steps of the specific optimization method for automatic tuning:

First, default the feasible region of each parameter based on our

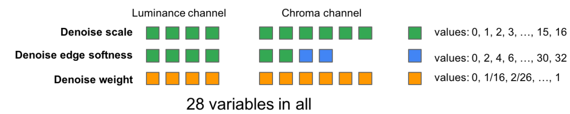


***Figure 5.** Quantization of the tuning parameters.*

manual tuning experience (shown in Fig. 5). As shown in Fig. 5, each color box means a tuning parameter, and the WNR block contains ten denoise scale parameters (Four for the luminance channel and six for the chroma channel corresponding to different frequencies), eight denoise edge softness parameters (Four for the luminance channel and four for chroma channel corresponding to different frequencies), ten denoise weight parameters (Four for the luminance channel and six for the chroma channel corresponding to different frequencies). According to our manual tuning experience, let the green boxes express the parameters with a feasible region of $[0, 16]$, let the blue boxes express the parameters with a feasible region of $[0, 32]$, and let the yellow boxes express the parameters with a feasible region of $[0, 1]$.

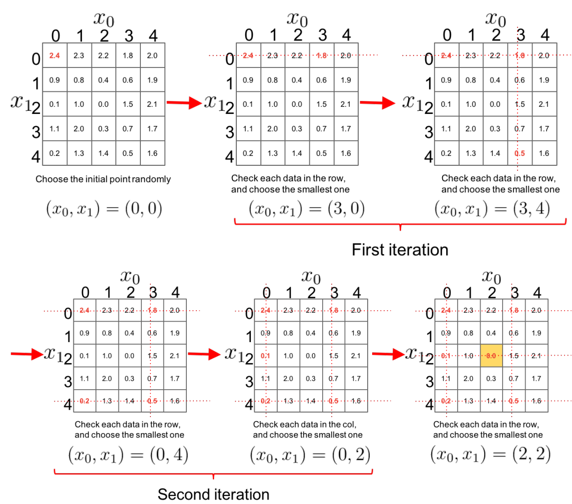


***Figure 6.** 2-D example of searching strategy. Start from a random initial point and optimize the parameters dimension-by-dimension.*

By performing a quantization for each parameter, the continuous optimization problem is converted to a discrete optimization problem. In this project, we chose seventeen discrete uniform values for each parameter (shown in Fig. 5) to reduce the complexity of optimization, as a result of that, there are $17^{28} > 2 \times 10^{34}$ combinations in all. Though the problem is already transferred to a discrete optimization problem, the computation complexity is still too high to efficiently search all combinations for the optimal solution.

Instead of exhaustively searching for the optimal solution, our searching strategy optimizes the parameters dimension-by-dimension. This process can be explained through a simple 2-D example shown in Fig. 6. In the example, there are five discrete feasible values in each dimension and the loss errors of perception model are shown in the figure corresponding to each combination of the two parameters . Our search strategy starts from a random point, such as the $(x_0, x_1) = (0, 0)$ point, searches all values in one dimension at one time and then chooses the parameters which corresponds to a minimum loss error of the perception model. Then, update the parameter in current dimension and process the next dimension until the error metric converges to the minimum value.

This method, which is related to the searching over of all dimensions, can not guarantee our result is the global minimum however. In order to get the approximate optimum, we borrow the idea of Particle Swarm Optimization (PSO) algorithm, instead of generating a single initial point randomly, we generate a group of initial points randomly and search from them at the same time, then assume the parameters corresponding to the smallest minimum error is the final optimal solution.

Result

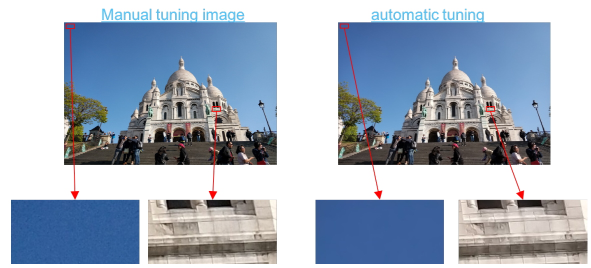
With the automatic tuning method shown above, the tuning parameters are iteratively optimized. Figure 7 shows one example of the optimization process. As shown in this figure, the error metric of perception model keeps on reducing, meaning that the image quality increases until the optimization algorithm converges to the minimum value. The comparison between manual tuning and automatic tuning methods with different light densities are shown in Fig.8. The automatic tuning can get similar or even better image quality in some cases than the manual tuning method. On the other hand, manual tuning costs several works to get the reasonable tuning parameters, but the automatic tuning method can get the optimal parameters in several hours.

		parameters in each iteration	Model output
initial point	DS	[5.0, 5.0, 5.0, 5.0], [5.0, 5.0, 5.0, 5.0, 5.0, 5.0]	
	DE	[5.0, 5.0, 5.0, 5.0], [5.5, 5.5, 10.0, 20.0]	
	DW	[0.0, 0.25, 0.4, 0.5], [0.0, 0.0, 0.0, 0.0, 1.0, 1.0]	24.04583817
first iteration	DS	[0.0, 0.0, 1.0, 2.0], [0.0, 5.0, 1.0, 0.0, 0.0, 1.0]	
	DE	[1.0, 9.0, 1.0, 6.0], [1.0, 5.5, 10.0, 20.0]	
	DS	[0.0, 0.25, 0.0, 0.0625], [0.0, 0.0, 0.125, 0.0, 0.0, 0.0625]	17.71459291
second iteration	DS	[0.0, 0.0, 1.0, 1.0, 0.0, 0.0], [1.0, 1.0, 0.0, 0.0, 0.0, 1.0]	
	DE	[2.0, 2.0, 1.0, 5.0], [1.0, 1.0, 10.0, 20.0]	
	DW	[0.0, 0.25, 0.0, 0.0625], [0.0, 0.0, 0.125, 0.0, 0.0, 0.0625]	17.59356879
third iteration	DS	[0.0, 0.0, 1.0, 1.0], [1.0, 1.0, 0.0, 0.0, 0.0, 1.0]	
	DE	[1.0, 8.0, 1.0, 5.0], [1.0, 1.0, 10.0, 20.0]	
	DW	[0.0, 0.25, 0.0, 0.0625], [0.0, 0.0, 0.125, 0.0, 0.0, 0.3125]	17.58215595

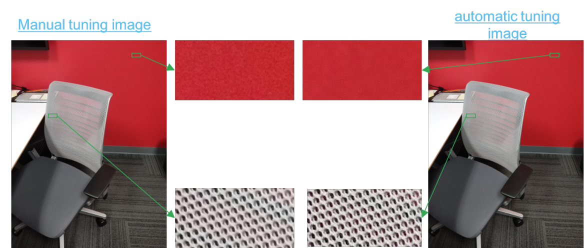
Figure 7. *Example of optimization process.*

Conclusion

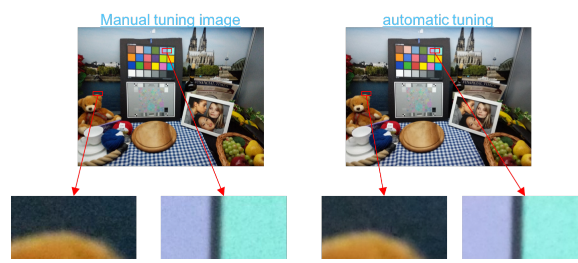
In this paper, we proposed and implemented an automatic tuning method. Meanwhile, we developed three different part of this algorithm. First, we improved the CPIQ standard metrics for better evaluation of image quality. Second, we developed a perception model for WNR automatic tuning. Third, we introduced a optimal searching strategy for automatic tuning. The automatic tuning method presented here is still a simple method, and there is still much work left to be done in the future to improve the automatic tuning method. Such as increasing the accuracy of the perception model, optimizing the optimization strategy, and reducing the computation time. But according to experimental results, our method achieves a promising result and the method can converge within a reasonable period of time. It also proves that automatic



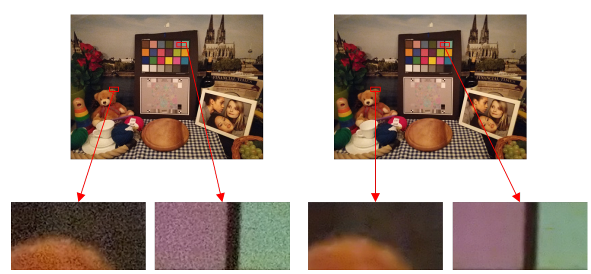
(a) Comparison between manual tuning and automatic tuning results with 2 gain illuminant



(b) Comparison between manual tuning and automatic tuning results with 8 gain illuminant



(c) Comparison between manual tuning and automatic tuning results with 16 gain illuminant



(d) Comparison between manual tuning and automatic tuning results with 32/ 64 gain illuminant

Figure 8. *Comparison between manual tuning and automatic tuning methods with different light intensities.*

tuning of camera blocks is viable and can replace manual tuning in the future.

References

[1] F. Cao, F. Guichard and H. Hornung, ”Measuring Texture Sharpness of a Digital Camera”, Proc. SPIE 7250, 72500H, 2009.

[2] Ann B. Lee, David Munford, Jinggang Huang, ”Occlusion a texture metric for camera phone images using a texture-based softcopy quality ruler”, Proc. SPIE, Vol. 7529, (2010).

[3] ISO 20462-3:2012 Photography - Psychophysical experimental methods for estimating image quality, 2012.

[4] McElvain J.S., Campbell S.P, Jin E. Miller J., ”Texture-Based Measurement of Spatial Frequency Response Using the Dead Leaves Target: Extensions, and Application to Real Camera Systems”, Electronic Imaging 2010.

[5] Elaine W. Jin, Jonathan B. Philips, Susan Famand, Margaret Belska, Vinh Tran, Ed Chang, Yixuan Wang, Benjamin Tseng, ”Towards the Development of the IEEE P1858 CPIQ Standard - A Validation Study”, Electronic Imaging, Image Quality and System Performance XIV, pp.88-94(2017).

[6] Damon M. Chandler, Sheila S. Hemami, ”VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images”, 2007 IEEE.

[7] Mark D. Fairchild, Garrett M. Johnson, ”Image Appearance Modeling” (2003).

[8] Kathy T. Mullen, ”The Contrast Sensitivity of Human Color Vision to Red-Green and Blue-Yellow Chromatic Gratings”, the journal of physiology, 1985.

[9] F. Cao, F. Guichard and Hornung, ”Dead Leaves Model for Measuring Texture Quality of a Digital Camera”, Proc. SPIE 7537, 75370E, 2010.

[10] $P1858^{TM}$/D2 Draft Standard for Camera Phone Image Quality, v24, 2016.

[11] Phillips, J.B. and Christoffel, D.W., ”Validating a Texture Metric for Camera Phone Images Using a Texture-Based Softcopy Quality Ruler”, Proc. SPIE, Vol. 7529, (2010).

Author Biography

Weijuan Xi received her BS in Electrical Engineering from Northwestern Polytechnical University, China (2013). she is currently pursuing a PhD degree and working on image processing in Electrical and Computer Engineering, at Purdue University. Her work focuses on image quality, computer vision and deep learning.