Segmented Images Colorization Using Harmony

Abstract

In the colorization process, artists fill image's regions with colors following harmony rules. Some region colorizations are more significant than others. In this paper we introduce a model for computer assisted colorization of segmented images using Itten's proportion contrast. Our model is able to process images composed of totally blank or partially filled regions. Though, each region must be filled with only one color. The user defines sectors representing colors on the chromatic hue wheel and may redefine each color's proportion. Then a set of colors is selected to compute the desired harmony. To proportionally colorize an image, already existing colors can be taken into account or the harmony can be applied only to blank regions. Due to the fact that regions cannot be divided, exact proportion contrast colorization of segmented images is a difficult algorithmic problem. In order to solve this problem we use different strategies and propose different methods to fill the regions. If they exist, exact solutions are proposed, otherwise different approximation methods are used to determinate combinations of colors-regions close to the defined proportions. Once the image is processed, the user can adapt hue (in the chosen sector), saturation and value of each region to enhance the result. Our model is flexible and specifically designed to help artists, illustrators, comics creators or any other user to automatically colorize (partially or totally) their images.

1. Introduction

People generally agree on the visual perception pleasantness provided by color relationships in an image. This is the definition given by Johannes Itten for the word harmony. Artists use this knowledge to create visual effects even though each artist possess his own palette of colors.

As shown in Figure 1, the first step of an illustration is a sketching. Then the artist roughly defines objects and atmosphere of the scene with uniform colors. During the colorization process, artists first focus on regions with important semantic meaning. Others regions are filled following the desired harmony. At this step harmonization is already taken into account for future details. The last step consists in adding refinements in the picture: shades of colors and details.

Figure 1. Different steps to create an illustration by Pierre Le Pivain alias LePixx

There exist different types of harmony. Work in computer graphics literature mainly focus on color relationships and forget that artists also specifically choose the quantity of each color they use. The contrast of proportion, used almost since the impressionist period, described by Itten [7] and largely used by artists has been largely neglected.

We propose a model to help users (principally artists, illustrators, comics creators...) to automatically fill and refine their images according to proportion harmony. Our work focuses on the second step realized by the artist (roughly fill areas).

Our model performs colorization on an input image, split in different regions, following Itten's color ratios. Each region is filled with one uniform color. Some of the regions of the input image can be precolorized by the user or the image can be left totally blank. Applying the harmony, the contribution of precolorized regions may, or may not, be taken into account.

Exact colorization of segmented images, given desired proportions, is a difficult problem. We propose different strategies in order to provide exact solutions if they exist or good approximations otherwise. Once the image is totally colorized, the user may adapt hue (in the chosen sector), saturation, value and create color shades on each region to enhance the image. Our model is flexible and well designed to help users to automatically fill (totally or partially) their images.

In section 2 we briefly summarize related work on color harmony. As we focus on the proportion of harmony defined by Itten, the section 3 presents the terminology defined by Sauvaget *et al.* in [14] and used in our model to describe harmony proportion. Our model is then described in section 4. We describe the colorization problem and present our different strategies and methods. We then exhibit our results and draw our conclusions.

2 Related work

Colors have always been used to convey emotions. For that reason, several domains have shown interest in color relationships. Combinations of color have been widely studied and different theories define them as harmonious or non-harmonious. First theories, Goethe [17], Chevreul [2], Young and Maxwell [10], have allowed to introduce new representations of colors [11, 13, 7]. Since then, color harmony has been described using representations of colors and rules to depict pleasant combinations. Most of these pleasant combinations correspond to spatial arrangements and others, introduced by Granville and Jacobson [6], use a quantitative representation.

Most of existing tools and ongoing research [12, 4, 1], use these theories except proportion one. Moreover, most of them propose sets of colors to colorize an image but do not help the user to colorize it.

Different issues from the main theories of color harmony are described by Westland *et al.* [18] who also notice that the principle of quantity has been widely forgotten.

Some studies of psychological assessment of color harmony with color combination pairs on a gray background [9], are about combinations of two colors whereas images we perceive generally consist in more than two colors.

Cohen-Or *et al.* [3] provide a model for the harmonization of colors using predefined harmonic templates composed at most of two sectors. The user may change the orientation of the hue template to globally change the colors of the image. An extension for videos has been proposed in [15]. As explained above, the harmony is restricted to two sectors.

Recently Sauvaget *et al.* [14] have proposed a proportion color harmony model that enhances image aesthetic in term of human color visual perception. Starting with a photograph or a completly filled image, this model detects proportion of harmony or the lack of harmony, finds a solution that verifies constraints defined by the user and proposes different possibilities to recolor the image by shifting hues.

But all of these methods cannot colorize an image or a part of it.

Sýkora *et al.*[16] have presented a semi-automatic colorization of foreground objects from two temporally different images of a cartoon. The goal is to make easier the step by step colorization of artists. One is a grayscale image and the other contains the reference colored object from which colors are deduced for each corresponding region in the grayscale image. Other work [8] proposes to colorize an image from a grayscale image and color scribbles using a nonlocal graph regularization. But in these models, color harmony is never considered.

We propose a model to help users (principally artists, illustrators, comics creators...) to automatically fill and refine their images according to proportion harmony defined by Itten. Our model allows the user to automatically colorize segmented images following a desired quantitative harmony and to perform refinements.

3 Using contrast of proportion

We use Itten's color harmony definition and principles. Itten defined harmony as a simultaneous judgment on two or many colors and explained that global perception of an image is necessary to perceive its contrast.

The seven contrasts of Itten use a hue/saturation/value model to describe each color. In that way, all colors can be placed on Itten's color wheel. We present in details Itten's proportion contrast and the terminology proposed by Sauvaget *et al.* [14] used hereafter in our model.

Itten divided the color wheel into six sectors to represent six different colors (yellow, orange, red, violet, blue and green). The hue is used to classify colors in these sectors. Desaturated areas are not taken into account as Itten considered that grayscale images are harmonious.

The proportion harmony is obtained when there is a quantitative balance between colors.

Table 1 summarizes the set of values describing the area proportions for each color. The sum of these proportional values is 360 degrees and defines the entire hue wheel (see Figure 2).

yellow	orange	red	violet	blue	green
30	40	60	90	80	60

Table 1. Itten's harmonic area proportions.

All proportion relationships between two to six colors can be deduced from this table. For example, to obtain harmonic color proportion between yellow, blue and green, the amounts for the yellow area should be $\frac{30}{30+80+60}$, $\frac{80}{30+80+60}$ for the blue and $\frac{60}{30+80+60}$ for the green. Following Itten's model, Sauvaget *et al.* define six sectors on the hue wheel. A sector $S = (\alpha_S, \beta_S, p_S)$ is then described by two angles α_S and β_S and a proportion value p_S. Default values proposed in the model are: $Y = (40, 75, 30)$, $O = (10, 40, 40)$, $R = (340, 10, 60)$, $V = (250, 340, 90)$, $B = (150, 250, 80)$, $G = (75, 150, 60)$.

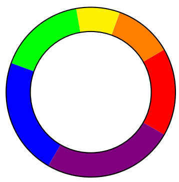

Figure 2. Itten's proportion color wheel

The user can redefine the number of defined sectors (from 2 to 6), the sectors angles, the proportion values and can specify the number of sectors thereof to be considered for the computation of harmony. In this harmonic scheme, sectors are not necessary contiguous but the intersection of all sectors must be empty. If the user changes the proportion values or the order of the color sectors, he creates new proportion considerations and we do not ensure that the resulting image will be harmonic as defined in different color harmony theories. In the following CS represents the list of considered sector to compute the harmony and S represents one of the six possible color sectors (Y, O, R, V, B, G).

4 Our model

Figure 3 presents our model work-flow. First, we introduce one assumption with respect to semantic aspect of colorization. We consider that the regions left by the artist for the computer assisted colorization process do not contribute to the semantic meaning of the picture. In other words, blank regions are filled according only to the specified harmony, without any consideration of the adequacy of the color and the colorized items. Our model can then be defined using five different steps. Note that after user refinements, the image obtained can be reintroduced in our model in order to harmonize colors taking into account newly defined blank regions (see figure 3).

1. Input segmented image: totally blank or partially filled;
2. User can colorize some of the regions (optional);
3. User redefinition of each sector (α_S, β_S, p_S) (optional);
4. Harmonization of the image using our methods;
5. User refinements (optional).

Following Sauvaget *et al.* model, the user is allowed to create CS and redifine sectors (sector angles and proportion) respecting the sector intersection constraint. By default we use the default values described in this model.

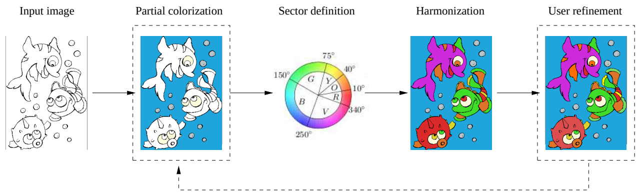

Figure 3. Framework overview: successive steps of our model where dash lines represents optional steps.

Hereafter we introduce a description of the colorization problem and briefly discuss the difficulty to solve it exactly. We then propose the different strategies and methods in order to obtain exact solutions if they exist or good approximations otherwise.

4.1 Problem description

The image colorization problem can be represented as following. Let's consider a hyperspace of dimension equal to the number of colors used to compute the harmony. Each color proportion gives one coordinate of the particular point which needs to be reached in order to respect exactly the desired harmony. However, the displacement on this hyperspace is not continuous. The set of regions to be filled defines the number of steps to be done and the size of each step. A graphical representation of this problem in dimension 3 is shown Figure 4.

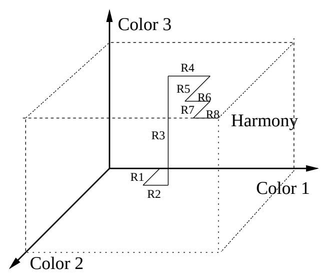

Figure 4. Representation of the colorization problem.

Although out of the scope of this paper, it is possible to demonstrate that solving this problem can be reduced to solve an instance of the multiple subset sum problem [5] which is an NP-Complete problem. One can find in the literature, algorithms able to give exact solutions of an instance of this problem or to show that there is no exact solution. However the time/memory cost of these algorithms is prohibitive for the scope considered in this paper. Thus, we propose different strategies in order to give, with reasonable time/memory resources, exact solutions if they are reachable or good approximations otherwise.

The search of exact solutions is done using dynamic programming (DP) and detailed section 4.3. In order to find good approximate solutions, we consider three strategies:

- Decreasing filling (DF) uses the greedy algorithm approach on a sorted list of blank regions (see section 4.4).
- Error minimization (EM) consists in minimizing an evaluation function on colorization error and is described section 4.5.
- Displacement maximization (DM) method considers the projections on the colors axes of the displacement in the hyperspace (see section 4.6)

The choice to use different strategies became obvious during sampling tests. Indeed, the strategies we propose have demonstrated their complementarity when observing the results obtained. In practice, our model computes for a given image the global harmony error using all strategies and keeps the result that minimizes the error. We discuss this issue in section 5.1.

4.2 Region listing and harmony computation

The input image is composed of regions which can all be blank or part of them may be already filled by the user. We assume that the colors used in the filled regions are the colors the user really desires. Thus, we never modify these colors while computing the specified harmony : only blank regions are filled.

A region map is created to contain filled and blank regions. We note $\mathcal{U}$ the list of unfilled regions and $\mathcal{F}$ the list of regions already filled. In the following, R_k denotes a region in $\mathcal{U}$. In order to compute the harmony, we have to calculate the number of pixels in each sector. The number of pixels of each sector of CS can easily be obtained from the number of pixels of each region. Considering that the pixels of a given region have the same hue, this number of pixels is assigned directly to the corresponding sector. We note $size(S)$ the number of pixels of sector S and $size(\mathcal{U})$ number of pixels to be filled in the image (the sum of the number of pixels composing all R_k. We note $ideal(S)$ the ideal number of pixels for sector S. $ideal(S)$ is computed using CS and following proportions (an example with yellow, blue and green is given section 3). According to $\mathcal{U}$ and $\mathcal{F}$ two types of proportion harmony can be realized. Equation 1 ensures that the existing number of pixels $size(S)$ in a sector is equal to $ideal(S)$ (left part of the difference in the equation). A value of $size(\mathcal{F})$ different of zero means that the colors used in pre-filled regions are taken into account. Otherwise, already filled regions do not contribute to the computation of the harmony.

$$\forall S \in CS, \ \frac{size(S)}{size(\mathcal{U}) + size(\mathcal{F})} = \frac{p_S}{\sum\limits_{X \in CS} p_X} \qquad (1)$$

In order to evaluate the solutions obtained, we define the global harmony error Δ as :

$$\Delta = \sum_{S \in CS} |ideal(S) - size(S)| \qquad (2)$$

In the following, we detail our colorization methods. These methods are built in order to exactly reach the specified harmony or at least to minimize the global harmony error.

4.3 Finding an exact solution

In order to obtain exact solutions, we use a method based on dynamic programing. Our algorithm focuses in finding a subset of regions for each color considered in the harmony computation. These subsets are computed in parallel in order to obtain dissociated subsets of regions for each color (a region cannot belong to different sectors). A backtrack search is also used to build colorization solutions.

Our algorithm can be summarized in two steps. In the first step, a sequence of sets is computed summarizing the set of all possible sums of the regions' size. This sequence is stored as a table. Each line of the table, noted E_i corresponds to the possible subsets considering i regions of $\mathcal{U}$. For example, we suppose that $\mathcal{U}$ contains 6 regions of sizes 8, 6, 6, 5, 2 and 1. We compute the harmony with $CS = (Y, O, R)$ and proportions respectively equal to (10,9,9) in pixels and we consider the Y sector. We obtain the following table (2):

Set	size(R_k)	0	1	2	3	4	5	6	7	8	9	10
E_0	$\emptyset$	1	0	0	0	0	0	0	0	0	0	0
E_1	8	1	0	0	0	0	0	0	0	1	0	0
E_2	6	1	0	0	0	0	0	1	0	1	0	0
E_3	6	1	0	0	0	0	0	1	0	1	0	0
E_4	5	1	0	0	0	0	1	1	0	1	0	0
E_5	2	1	0	1	0	0	1	1	1	1	0	1
E_6	1	1	1	1	1	0	1	1	1	1	1	1

Table 2. Each column represents a goal and each line all possible subset sizes. A path to find a subset of size 10 is shown in red (leading to $\{2, 8\}$). A path to find a subset of size 7 is shown in blue (leading to $\{2, 5\}$).

For the second step, we know if we can find subsets of regions matching color proportions. Nevertheless, we cannot guarantee that these subsets are disjoint due to the fact that we compute them one after another. Then the second step consists in reconstructing all possible solutions from the last set. In order to find disjoint subsets we compute them in parallel using the same approach. The storage table has to be as large as the biggest sector. At the end of the process, either many choices are available, either there is no choice left. In this case, we have to go backward and reconsider the last choice we made using a backtrack search.

To begin, we point each 1 of the last row corresponding to our proportions. Each step consists in moving all the pointer straight up except one which have to go up left (a region can only be filled with one color). Many choices can be available or none. If there is no choice, we have to go backwards and reconsider the last choice we have made.

Following our example, we try to find a subset of total size 10 shown in red in table 2. As a 1 is present at the last row E_6 column 10, a 10 pixel combination of regions can be computed. Let's set a pointer to this position. Two choices are possible at E_i, j (where j is the column): we can move the pointer up to row E_{i-1} if a number 1 exist or we can move it to $E_{i-1}, j - size(R_i)$. In this example, we can move the pointer up to row E_5 but not to row E_4 because of a 0. This means that there is no solution without region R_5. So we move the pointer up left, 2 columns before (size of R_5) and we obviously find a number 1. This means our solution includes region R_5. Then, we start again from this point and we move the pointer straight up until row E_1. The last step consists in the selection of the last region R_1 (of size 8) by moving the pointer up left. We obtain $(2 + 8 = 10)$. In summary, going straight up means not selecting the region whereas to go up left means selecting the region.

The dynamic programming method (DP) provides all exact solutions if any exist. The user may then choose the colorization he visually prefers in these solutions. However, this approach uses significant memory and may take long time due to the backtrack search. That is why we introduce, in the following, approximation methods.

4.4 Decreasing filling

The decreasing filling method (DF) is based on the greedy algorithm applied to the list $\mathcal{U}$ sorted by size of regions in a decreasing order. It consists in filling the regions of the sorted list $\mathcal{U}$ assigning greatest regions first. Our algorithm consists in the following steps:

1. For each sector S, the difference between the ideal number of pixels $ideal(S)$ and the actual number of pixels $size(S)$ is calculated.

2. We fill the region R_k with the color of S corresponding to the greatest difference.

3. The region R_k is marked as filled and $\mathcal{U}$ and $size(\mathcal{U})$ are updated accordingly. The number of pixels of region k is added to $size(S)$.

4. At this point, $size(S)$ may become greater than $ideal(S)$. In that case, we need to update $ideal(X)$ for each sector X of CS following Equation 3. These different steps are realized until $\mathcal{U}$ is empty.

   $\forall X \in CS$, if $(size(X) - ideal(X) > 0)$ then

   $$ideal(X) = ideal(X) \times h(X) \tag{3}$$

   where $h(X) = 1 - \frac{size(X) - ideal(X)}{\sum_{Y \in CS \ ideal(Y) - size(Y) > 0} ideal(Y)}$

4.5 Error minimization

The main idea of the error minimization method (EM) is based on computing an evaluation function of the colorization error defined by:

$$f(S, R_k) = \frac{|ideal(S) - size(S) - size(R_k)|}{ideal(S) - size(S) + size(R_k)} \tag{4}$$

This evaluation function is built in order to favor the matching between regions and colors leading to minimal colorization error with respect to the size of the pair color/region considered. Our algorithm runs as follow:

- For each sector S and each region R_k we compute $f(S, R_k)$.

- We fill the region R_k with color of S for which $f(S, R_k)$ is minimal and update $size(S)$ accordingly.

The process is repeated until there is no more region to fill.

4.6 Displacement maximization

The strategy of displacement maximization consists in defining a norm in order to measure the differential displacement between the current position in the hyperspace and the ideal position to be reached. This norm is based on projections with respect to colors axes. Each projection is evaluated independently, then the best displacement is chosen considering every projection evaluation.

The displacement maximization methods use norms of order 2 weighted by a factor. This factor is equal to the projection of the ideal position elevated to power ω. The three methods we propose correspond to the three values: $\omega = 0$, $\omega = 1$ and $\omega = 2$. We name these different proposals respectively DM0, DM1, and DM2. Our algorithm runs as follow:

- For each sector S and each region R_k we compute:

$$g(S, R_k) = \frac{(D)^2 - (D - size(R_k))^2}{ideal(S)^{\omega}} \quad (5)$$

where $D = ideal(S) - size(S)$.

- We fill the region R_k with color of S for which $g(S, R_k)$ is maximal and we update $size(S)$ accordingly.

The process is repeated until there is no more blank region left.

5 Results

This section is divided in two parts: statistical evaluations of our approximation methods for computer assisted image colorization; results obtained by our different methods on real images.

5.1 Statistical evaluation

We have run various experiments to compare and evaluate our different methods. In order to do so, we have generated ”virtual” segmented images and colorization rules. The virtual images are composed of a maximum of 500 000 pixels and a maximum number of blank regions varying in the parameter set $\{10, 50, 100, 500, 1000, 5000\}$. This set of parameters has been chosen specifically to correspond to the different steps followed by artists during image colorization process, for example in comics realizations. The colorization rules have been defined by selecting a number of colors for the harmony between 2 and 6. The experimental procedure is defined as follows:

- Pick at random the size of the image, the number of blank regions and the number of colors,
- Generate from an uniform distribution the sizes of the colors' proportion and of each blank region,
- Finally, colorize the segmented image with each method and measure the global harmony error (see Equation 2).

This experimental procedure has been repeated 10 000 times for each parameter. Due to the lack of space, we present in Table 3 only a small part of the measures we have obtained. For all cases and all methods, we compute the global harmony error. The first line of each table cell presents the percentage when the considered method is the only one that obtains the minimal global harmony error. The second line presents the percentage when the considered method and at least another one obtain the minimal global harmony error.

The first observation we made is that approximation methods reach exact solutions in a substantial number of cases, particularly when there is a great number of regions (1000, 5000). This is due to the fact that the number of exact solutions grows with the number of blank regions. Our choice to use different strategies is confirmed by the observation that each method exclusively covers some particular subspace of the solution space. Within her subspace, a particular method is significantly more efficient than others. Furthermore, we notice that at least three methods usually converge to the same approximation in a large part of the cases.

Due to the nature of NP-Completeness, there exists none polynomial algorithm, nor combination of polynomial algorithms, effective for all instances of the problem. However, our statistical evaluations shows that our approximation methods have good behavior, particularly for great numbers of blank regions.

	DF	EM	DM0	DM1	DM2
10	10.37 63.58	4.02 42.56	4.99 51.12	4.71 49.09	5.86 53.46
50	9.74 38.38	4.92 14.17	4.14 31.70	28.64 39.58	16.88 29.39
100	8.30 33.98	4.40 10.17	3.69 29.01	32.26 40.36	20.49 28.79
500	2.63 38.26	3.30 15.73	3.27 41.02	25.55 53.23	17.50 42.83
1000	1.42 52.86	2.42 27.20	1.95 55.91	17.21 69.00	12.23 59.86
5000	0.40 88.69	0.50 67.69	0.40 89.30	4.10 92.50	2.30 87.30

Table 3. For each parameter both lines give the percentage of best approximation for each method. First line indicates best approximation reached alone, second line best approximation without distinction.

5.2 Real application on images

We also applied our model to real images of various types. We are able to provide exact colorizations and good approximations on more than 30 artist's illustrations, drawings or segmented images extracted from photographs. However due to lack of space, we present in this paper five examples : Figures 5, 6, 7, 8 and 9. Those examples illustrate that our model is well designed for computer assisted image colorization. We obtain good approximations and the colorizations are done accordingly to artist constraints (color harmony proportions).

Our algorithm also proposes an extended functionality as a side effect. Indeed, there exist different solutions leading to minimal global harmony errors. The use of different strategies leads to different colorizations respecting the desired harmony. An example of this functionality is illustrated Figures 6 and 7.

We present Table 4 the computation time for some examples. The computations have been done on a PC with 2.5Ghz frequency and 3Go memory. The average computation time is equal to 0.69 seconds which we consider to be reasonable. However, the CPU time needed to produce such pictures mainly depends on the number of regions and their respective size.

We recommend to first run the approximation methods before searching an exact solution. Indeed, it appears that our approximation methods frequently succeed to find an exact solution. Furthermore, the dynamic programming approach may be slow (due to the solution reconstruction steps) and memory consuming (depending on the number of regions).

	DP	DF	EM	DM0	DM1	DM2
Fish	0.08	0.07	0.10	0.05	0.06	0.09
Dragon	0.46	0.64	0.60	0.67	0.65	0.69

Table 4. Computation time is given in seconds.

Figures 8 and 9 present all steps of our work-flow. The artist has made a sketching and partially colorizes some regions. Then our system produces solutions to achieve colorization according to proportion harmony constraints given by the artist. The result is finally refined by the artist and note that colors can be shaded or changed while the sector definition is preserved.

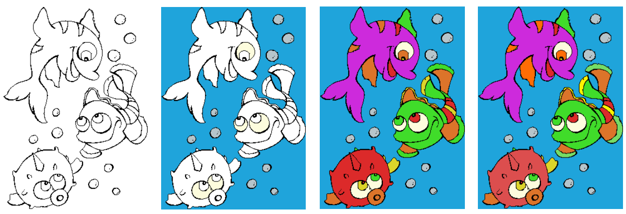

Figure 5. From left to right: sketching (260×364 **pixels); partial colorization; harmonization with** $CS = (Y, O, R, G, V)$ **and DF method which has produced an exact solution; artist refinement.**

6 Conclusion

In this paper, we have presented a new model for computer assisted colorization of segmented images according to a selected harmony based on Itten's proportion contrast. While defining our model, we make the assumption that the regions left by the artist for the automated colorization process do not contribute to the semantic meaning of the picture.

The problem of exact colorization of segmented images given a harmony is a difficult algorithmic problem. Thus, we propose different strategies in order to provide them to the user, with reasonable time/memory resources, exact solutions if they exist or good approximations otherwise. We described several methods based on those strategies. We conducted statistical evaluations and also real image computations which demonstrate the effectiveness of our methods and model.

Our model provides a helpful computer assisted tool to either basic users or accomplished artists who desire to create specific harmony while coloring their images. This tool is flexible and allows the user to dispose of different colorizations equivalent with respect to the specified harmony. After the harmonization process, the user may proceed hues, saturations and values refinements for a better visual appearance.

In future work, we plan to improve our model by allowing the user to indicate weighted preferences in the choice of colors for specific regions. We also plan to consider coupling our approach with Kuler's model to obtain sets of colors based on the six other types of harmony compatible with our proportion model.

References

[1] Adobe Kuler, 2006. http://kuler.adobe.com.

[2] M. E. Chevreul. *De la loi du contraste simultané des couleurs et de l'assortiment des objets colorés*. Paris, 1839.

[3] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu. Color Harmonization. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 25(3):624–630, 2006.

[4] Color Schemer, 2000. http://www.colorschemer.com.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms (second edition)*. MIT Press and McGraw-Hill, 2001.

[6] W. C. Granville and E. Jacobson. Colorimetric specification of the color harmony manual from spectrophotometric measurements. *Journal of Optical Society of America*, 34(7):382–395, 1944.

[7] J. Itten. *Kunst der Farbe*. Ravensburg: Otto Maier Verlag, 1961.

[8] O. Lezoray, V.-T. Ta, and A. Elmoataz. Nonlocal graph regularization for image colorization. In *ICPR*, pages 1–4, 2008.

[9] u. Li-Chen and M. Ronnier Luo. A colour harmony model for two-colour combinations. In *Color Reasearch and Application*, 2006.

[10] C. Maxwell. On the theory of compound colours. In *Philosophical Transactions*, number 150, pages 57–84, 1860.

[11] A. H. Munsell. *A Grammar of Colors*. Mittineague : Strathmore Paper Company, 1921.

[12] F. Nack, A. Manniesing, and L. Hardman. Colour picking: the pecking order of form and function. In *ACM Multimedia*, pages 279–282, 2003.

[13] W. Ostwald and F. Birren. *The Color Primer*. New York: Van Nostrand Reinhold Company, 1969.

[14] C. Sauvaget and V. Boyer. Harmonic colorization using proportion contrast. In *Afrigraph*, pages 63–72, 2010.

[15] N. Sawant and N. J. Mitra. Color harmonization for videos. In *ICVGIP*, pages 576–582, 2008.
[16] D. Skora, J. Burinek, and J. ra. Unsupervised colorization of black-and-white cartoons. In *In Proc. NPAR (2004*, pages 121–127, 2004.
[17] J. W. von. Goethe. *Zur Farbenlehre*. Stuttgart: Cotta'sche Buchhandlung, 1810.
[18] S. Westland, K. Laycock, V. Cheung, P. Henry, and F. Mahyar. Colour harmony. In *SDC : Society of Dyers and Colourists*, 2007.

Figure 6. This 798×885 pixels image illustrates different strategies leading to different colorizations. The harmonization is defined by $CS = (O, R)$: original input image, DP, DF, DM0,DM1 and DM2 methods where $\Delta = 0$ pixel for DP and $\Delta = 1$ pixel for the others.

Figure 7. Sketching (651×560 pixels); partial colorization; harmonization with $CS = (Y, O, R, G, B, V)$ and DF and EM methods and their artist refinement with shading.

Figure 8. Sketching (570×806 pixels); partial colorization; harmonization with $CS = (Y, O, R, V, B)$ and DP method; artist refinement with shading.

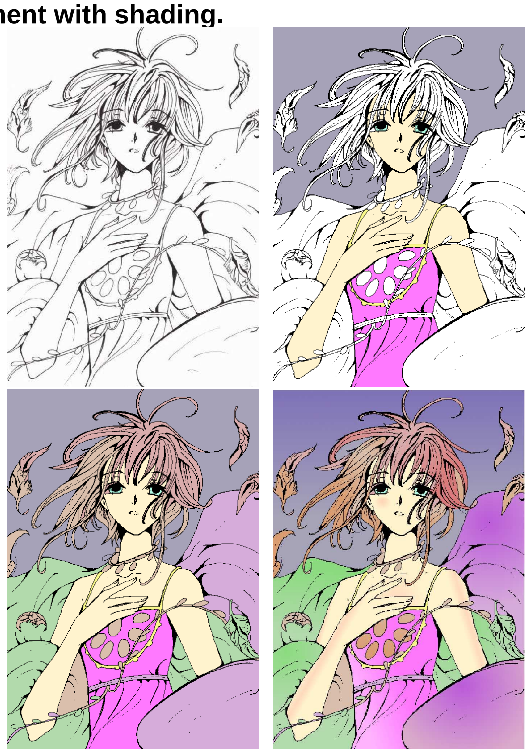

Figure 9. Sketching (500×714 pixels); partial colorization; harmonization with $CS = (O, R, G, V)$ and DP method; artist refinement with shading.