Automated Colorization of Segmented Images Based on Color Harmony

Catherine Sauvaget[1], Jean-Noël Vittaut[1], Jordane Suarez[1], Vincent Boyer[1], Stéphane Manuel[2]
[1]L.I.A.S.D University Paris 8
2 rue de la Liberté
93526 Saint-Denis. FRANCE
{cath, jnv, suarez, boyer}@ai.univ-paris8.fr

[2]UPEC ESIAG
71 rue Saint Simmon
94017 Créteil
stphn.manuel@gmail.com

ABSTRACT: *We propose in this paper an automated colorization of segmented images based on a color harmony rule defined by Itten called the contrast of proportion. Artists create an image drawing a sketch and filling its regions following harmony rules. The most significant regions are generally filled first because of their meaning. Then the other regions are filled depending on the previous colors. Our model is able to process images composed of totally blank or partially filled regions. The user defines sectors representing colors on the chromatic hue wheel and may redefine each color 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. Each region is filled with only one color and cannot be divided. Thus, 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 three different approximation methods are used to determine 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 results show that our model is flexible and well designed to help artists, illustrators, comics creators or any other user to automatically colorize (partially or totally) their images.*

Keywords: Harmony, Colorization, Stylization, Enhancement, Image Processing

Received: 1 September 2010, Revised 29 September 2010, Accepted 8 October 2010

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 [1]. Artists use this knowledge to create visual effects even though each artist possesses his own palette of colors.

As shown in Figure 1, the first step of an illustration process is a sketching. Then artists roughly define 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.

Different types of harmony exist. Work in computer graphics literature mainly focuses on color relationships and Illustration provided by Pierre Le Pivain aka LePixx. forgets 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 [1] and largely used by artists has been widely neglected.

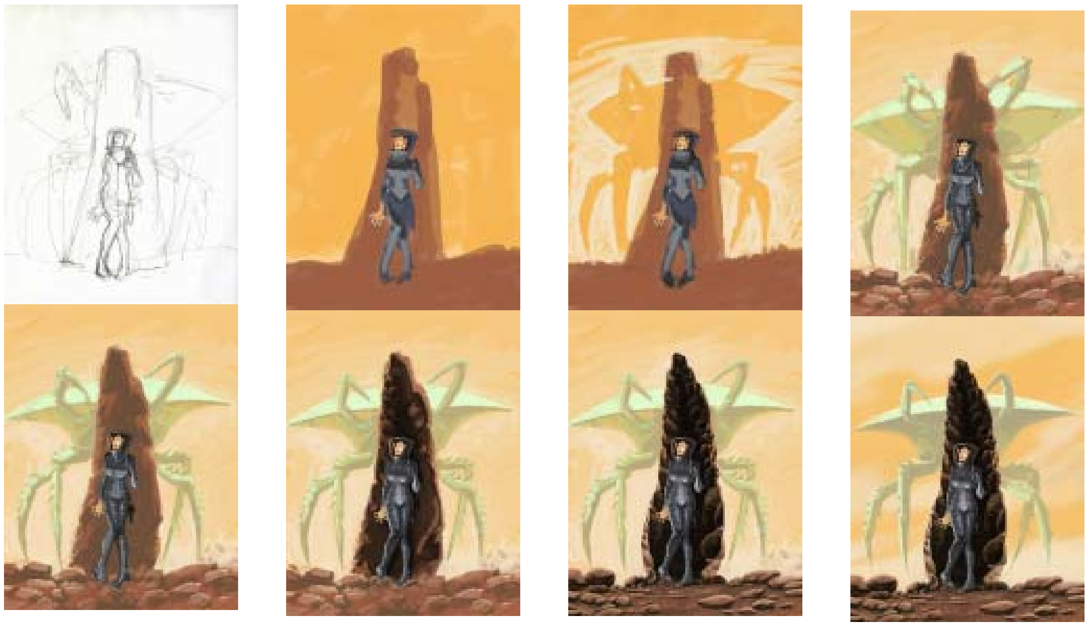

Figure 1. Example of consecutive steps needed during the creation process. Illustration provided by Pierre Le Pivain aka LePixx

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 filling 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 II we briefly summarize related work on color harmony. As we focus on the proportion of harmony defined by Itten, section III presents the terminology defined by Sauvaget *et al.* in [2] and used in our model to describe harmony proportion. Our model is then described in section IV. We describe the colorization problem and present our different strategies and methods. We then present 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 colors have been widely studied and different theories define them as harmonious or non-harmonious. First theories by Goethe [3], Chevreul [4], Young and Maxwell [5], allowed to introduce new representations of colors [6], [7], [1]. 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 [8], use a quantitative representation.

Most of existing tools and ongoing research ([9], Schemer1 or Kuler2) use these theories except the quantitative one. An example is provided Figure 2. Moreover, most of them propose sets of colors to colorize an image but do not help the user to colorize it.

Figure 2. Snapshot of Kuler's tool. This tool proposes palettes of colors generated according to several harmony rules. However, quantitative representation rule is not provided

Different issues from the main theories of color harmony are described by Westland *et al.* [10] who also notice that the principle of quantity has been widely forgotten. Li-Chen Ou and Ronnier Luo [11] proposed a color harmony model for two color combinations. This model is based on the results of psychological assessment for color combination pairs on a neutral gray background (see Figure 3). However, the images we are used to perceive, generally consist in more than two colors.

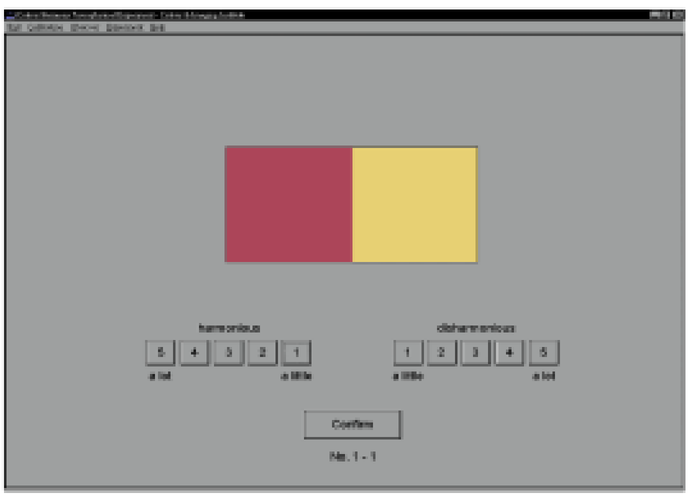

Figure 3. Image of Li-Chen Ou and Ronnier Luo (curtsy of Luo) psychological studies. The experimentation consists in asking the user to quantify how harmonious is a pair of colors

Cohen-Or *et al.* [12] provided a model for the harmonization of colors using predefined harmonic templates composed of two sectors at most (see Figure 4). After finding the template that best fits the image, the user may change the orientation of the hue template to globally change the colors of the image. A graph-cut optimization is used to enforce a continuous modification of colors in the image space. An extension for videos has been proposed in [13]. Moreover a cut-and-paste system is described that allows the user to realize color harmonization and compose scenarios. For example, color harmonization can be realized independently for the background and foreground and a final image can be created by composition of the latter two. As explained above, the harmony presented here has some limitations. The harmonic templates are composed of at most two sectors and the contrast of proportion cannot be applied with this method.

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

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

Recently Sauvaget *et al.* [2] proposed a model of proportion color harmony that enhances image aesthetic in terms of human visual perception of colors. Starting with a photograph or a completely filled image, this model detects proportion of harmony (or lack of harmony), finds a solution that verifies user-defined constraints and proposes different possibilities to adjust image colors by shifting hues.

However, the models and tools cited above do not provide methods for colorization.

S´ykora *et al.*[14] presented a semi-automatic colorization of foreground objects from two temporally different images of a cartoon. The goal is to make easier for the artists the step by step colorization. One is a gray-scale image and the other contains the reference colored object from which colors are deduced for each corresponding region in the gray-scale image.

Lezoray *et al.* [15] proposed to colorize a gray-scale image from color scribbles using a non-local graph regularization. In the same context, Qu et al [16] apply this kind of colorization to mangas. They propagate color of user scribbles over region on manga images that contain many strokes, hatching or halftoning. They find regions using Gabor wavelet filters to measure the pattern continuity. They propose different colorization methods. But in these articles color harmony is not considered.

Following previous research [17], we propose the first model taking into account both approaches: harmony computation and colorization. The model we describe in this article is designed to provide an automated colorization process based on color harmony for users (principally artists, illustrators, comics creators. . .). Our tool allows the user to automatically fill and refine segmented images according to Itten's proportion harmony rules.

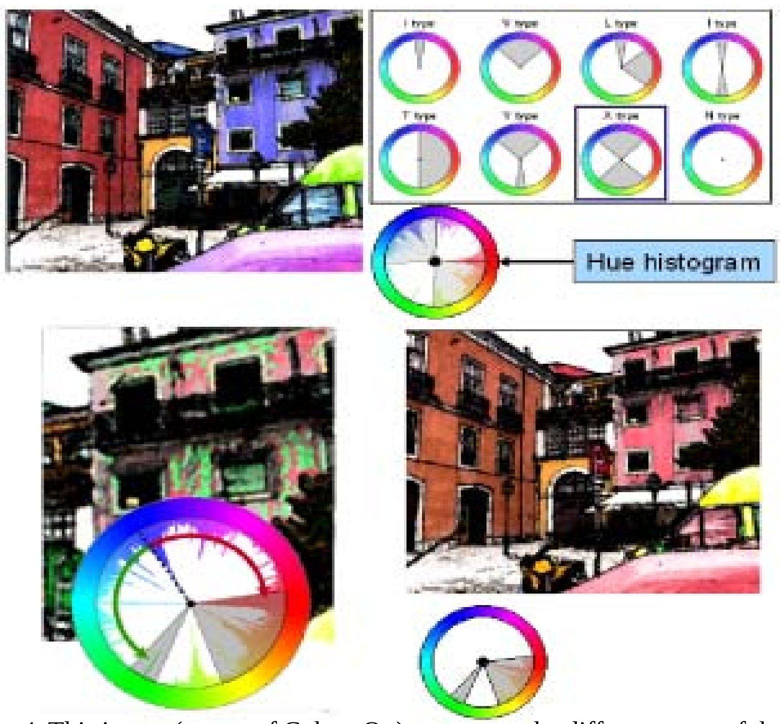

Figure 4. This image (curtsy of Cohen-Or.) represents the different steps of the model provided by Cohen-Or *et al.*. First step consists in finding the closest template from the picture's histogram. Then the hue of pixels which are not in the harmonious areas, is modified accordingly

3. Using Contrast of Proportion

We use Itten's color harmony definition and principles. Itten defined harmony as a simultaneous judgment of 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.* [2] 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. Completely desaturated areas are not taken into account as Itten considered that gray-scale images are harmonious.

The proportion harmony is obtained when there is a quantitative balance between colors. Table I 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 5).

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

Table 1. Itten's Harmonic Area Proportions

All proportion relationships from 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.* [2] defined six sectors on the hue wheel. A sector S = (S_α, S_β, S_p) is described by two angles S and Sand a proportion value Sp. The 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).

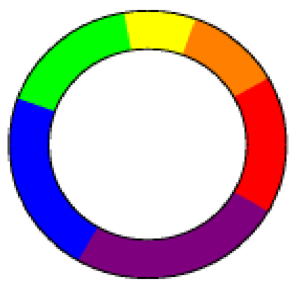

Figure 5. Itten's proportion color wheel

In the following, DS represents the list of defined sectors of color. Our system allows the user to create DS respecting the sector intersection constraint. Figure 6 presents two different possible sector configurations and one impossible configuration. For the first one, we use the default values described above and DS = (Y,O,R, V,B,G). For the second one, only 4 sectors are defined DS = (Y,R,V,B). Note that Y_β is no more the default value and sectors V and R are no more contiguous. The last one illustrates an incorrect configuration since R $\cap$ O $\neq \emptyset$ (i.e. angle $R_\alpha O_\beta$ is counterclockwise and RO is clockwise).

The user can modify the number of defined sectors (from 2 to 6), the sectors angles, the proportion values and specify the number of sectors 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. Changing proportion values or the order of the color sectors creates new proportion considerations. In such case, we do not ensure that the resulting image will be harmonious 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

First, we introduce an 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 illustrated Figure 7. 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.

1) Input segmented image: totally blank or partially filled;

2) User can colorize some of the regions (optional);

3) User redefinition of each sector (S_{α}, S_{β}, S_{p}) (optional);

4) Harmonization of the image using our methods;

5) User refinements (optional).

The user is allowed to create CS and redefine sectors (sector angles and proportion) respecting the sector intersection constraint. If sectors are not redefined we use the default values described in this model.

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 fast and good approximations otherwise.

4.1 Problem description

The image colorization problem can be presented as follows. 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 in Figure 8.

This problem can be viewed as a multiple knapsack problem where color proportion is a knapsack and a region is an item with same weight and value. It is known to be NP-Complete [18]. Solving this multiple knapsack problem allows to give each region a color according to the proportions. If an exact solution is found for the multiple knapsack problem, it is a solution for our 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.

4.2 Region listing and harmony computation

The input image is composed by 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 C$\mathcal{S}$ 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 $size(\mathcal{S})$ 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($\mathcal{S}$) the ideal number of pixels for sector S. *ideal* ($\mathcal{S}$) is computed using C$\mathcal{S}$ and following proportions (an example with yellow, blue and green is given section III). According to $\mathcal{U}$ and $\mathcal{F}$ two types of proportion harmony can be realized. Equation 1 ensures that the existing number of pixels $size(\mathcal{S})$ in a sector is equal to *ideal*($\mathcal{S}$). 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.

$$\forall \; S \in CS, \frac{size(\mathcal{S})}{size(\mathcal{U}) + size(\mathcal{F})} = \frac{ps}{\sum_{X \in CS} px} \tag{1}$$

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

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

5. Solving Strategies

In this section, we detail the strategies we propose in order to solve and/or approximate the image colorization problem. In fact, these strategies describe the colorization methods built in order to reach exactly the specified harmony or at least to minimize the global harmony error. The search of exact solutions is done using dynamic programming (DP).

In order to also find fast and good approximate solutions, we consider three strategies:

• Decreasing filling (DF) uses the greedy algorithm approach on a sorted list of blank regions,

• Error minimization (EM) consists in minimizing an evaluation function on colorization error,

• Displacement maximization (DM) method considers the projections on the colors axes of the displacement in the hyperspace.

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

5.1 Searching for 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. With this method, all exact solutions can be calculated if they exist.

Our algorithm can be described in two steps. In the first step, a sequence of sets is computed summarizing the set of all possible sums of the size of the regions. This sequence is stored as a table. The second step consists in reconstructing all possible solutions using this table and a backtrack search algorithm.

First step: Our goal is to compute all possible sizes of $\mathcal{U}$ subsets. In that way, we consider one region at a time.

First of all, if we have not seen any region, we already know the subset of size 0 is possible. We denote it $E_0 = \{0\}$.

Now, we add region R_1 to our problem. The possible subsets are $\emptyset$ and $\{R_1\}$. So the possible sizes are in E1 = $\{0, size(R_1)\}$.

Let's add region R_2 to our problem. Now the possible subsets are $\emptyset$, $\{R_1\}$, $\{R_2\}$ and $\{R_1,R_2\}$. So the possible sizes are in $E_2 = \{0, \mathit{size}(R_1), \mathit{size}(R_2), \mathit{size}(R_1) + \mathit{size}(R_2)\}$.

When a region Rk is added, we compute E_k as

$$E_k = E_{k-1} \cup \{x + \mathit{size}(R_k), \forall\, x \in E_{k-1}\}$$

Finally, when all n regions of $\mathcal{U}$ have been processed, the set En contains all $\{R_1, \ldots, R_n\}$ subset sizes.

For each subset size not in En, we already know that there is no exact solution. Let's suppose we want to construct a subset of size x where $x \in E_n$. There are two possibilities:

• if $x \in E_{n-1}$ then it means that we can find a $\{R_1, \ldots, R_{n-1}\}$ subset of size x and region Rn will not be part of the final solution.

• if $x \in \mathit{size}\,(R_n) \in E_{n-1}$ then it means that we can find a $\{R_1, \ldots, R_{n-1}\}$ subset of size $(x - \mathit{size}(R_n))$ and region R_n will be part of the final solution.

Each line of the table, noted Ei corresponds to the possible subsets considering i regions of $\mathcal{U}$. For example, we suppose that U contains 6 regions of sizes 8, 6, 6, 5, 2 and 1. We choose an harmony with CS = (Y,O,R) and proportions respectively equal to (10,9,9) in pixels. We compute the Table II. This table has to be as large as the biggest sector (Y in our example). If we consider for example the Y sector which has size 10, one path to find a subset of size 10 is shown in red.

Second step: From the table values, 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. 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 pointers straight up except one which has 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 made.

Following our example, we try to find a subset of total size 10 shown in red in table II. 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 exists 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 E4 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 the region R5. 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 that the region is not selected whereas going up left means that it is selected.

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 a long time due to the backtrack search. That is why we introduce approximation methods in the following.

Set	size(R_k)	0	1	2	3	4	5	6	7	8	9	10
E_0	∅	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})

5.2 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 Rk with the color of S corresponding to the greatest difference.

3) The region Rk 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.

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

$$ideal(X) = ideal(X) \times h(X) \quad (3)$$

where

$$h(X) = 1 - \frac{size(X) - ideal(X)}{\sum\limits_{\substack{Y \in CS \\ ideal(Y) - ideal(Y) > 0}} ideal(Y)}$$

5.3 Error minimization

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

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

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

- For each sector S and each region Rk 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.

5.4 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 and 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: ω = 0, ω = 1 and ω = 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,Rk) is maximal and we update $size(S)$ accordingly. The process is repeated until there is no more blank region left.

6. Results

This section is divided in two parts. First, we present statistical evaluations of our approximation methods realized on generated instances. Then we show results obtained by our different methods on real images.

6.1 Statistical evaluation

We carried out various experiments to compare and evaluate our different methods. In order to do so, we 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, 25, 50, 75, 100, 250, 500, 1000, 2500, 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 a uniform distribution the sizes of the color 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. We present in Table III the measures we obtained. The global harmony error is computed for all cases and all methods and we consider that the best approximation is reached when the global harmony error is minimal. The first line of each table cell presents the number of times, given in percentage, when the method was the only one to reach the minimal global harmony error. The second line gives the percentage of best approximation when several methods converged to the global harmony error.

At first, we observe that approximation methods reach exact solutions in a substantial number of cases, particularly when there is a great number of regions (1000, 2500, 5000). This is due to the fact that the number of exact solutions grows with the number of blank regions. For a limited image size (in our experiments, the maximum size of an image is equal to 500 000 pixels), when the total number of regions increases, the number of regions with very small size grows significantly. The existence of such small regions significantly helps to converge to exact solutions. Figure 12 represents the data of third lines in each row of Table III and provides an illustration of the evolution of the number of exact solutions reached by our methods for each parameter. Furthermore, we notice that at least three methods converge to the same approximation in most cases.

We can see on Figure 9 that the global harmony error average obtained using our approximations decreases quickly when the number of regions grows. This observation shows that our approximation methods are efficient. We can also notice that, for each parameter, the decreasing filling method (DF) gives the lowest global harmony error in average.

Our observations demonstrate the complementarity of the methods we propose. From the percentages given in the first lines of Table III (see also Figure 11), it appears that each method exclusively covers some particular subspace of the solution space. In those subspaces only one method reaches the best approximation. A particular method is, respectively to its subspace, significantly more efficient than others. For example, the displacement maximization method (DM1) is the only one to reach the best approximation in more than 30% of the cases for the parameters 75, 100 and 250. Figure 11 shows that this complementarity is mandatory for small and average parameters.

Figure 10 gives an illustration corresponding to the second lines in each row of Table III. The error minimization method (EM) has the lowest efficiency. However, this method provides solutions for particular subspaces. For parameter 10, the (EM) method provides the best results according to the number of exact solutions found. That is why, even if it tends to be globally less efficient than other methods, it is still useful.

From the observation of Figure 10 and Figure 11, one can see that the maximization method (DM1) seems to be the most efficient globally. However, we already noticed that the decreasing filling method (DF) gives better results in average. Due to the nature of NP-Completeness, there exists neither polynomial algorithm, nor combination of polynomial algorithms, effective for all instances of the problem. However, our statistical evaluations show that our approximation methods have good behavior, particularly for a great number of blank regions.

In Table IV, we present the results obtained with the dynamic programming (DP) and approximation methods for parameter 10. The first line of the table gives the number of exact solutions found. The second line details the number of times (in percentage) a method reached the best possible approximation. Obviously, this percentage is equal to 100% for the dynamic programming method. The last line shows the average relative error of the global harmony. This average is computed whenthe best possible approximation is not reached by the approximation methods. For parameter 10, we were able to compute exact solutions or best possible approximations for all the 10 000 experiments. For higher parameters, the dynamic programming method often failed to compute exact solutions (and/or best possible approximations) in reasonable time. This is why we do not present such a table for the other parameters.

6.2 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 artists' illustrations, drawings or segmented images extracted from photographs. We present six examples : Figures 13, 14, 15, 16 17, and 18. 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's constraints (color harmony proportions).

	DF	EM	DM0	DM1	DM2
10	10.35	4.29	4.73	4.75	6.33
	63.49	42.16	51.27	48.82	52.75
	0.02	0.06	0.02	0.02	0.04
25	10.01	5.61	4.80	21.46	13.00
	44.75	22.40	37.54	41.18	34.34
	0.12	0.09	0.12	0.11	0.9
50	9.49	5.25	3.94	29.21	16.81
	38.21	14.65	31.52	40.33	29.02
	0.46	0.28	0.46	0.37	0.41
75	9.17	4.65	3.75	32.17	19.16
	34.58	11.52	28.63	40.93	28.58
	0.75	0.47	0.75	0.88	0.79
100	8.51	4.64	4.03	33.35	19.79
	33.23	10.32	28.25	41.11	27.78
	1.28	1.00	1.28	1.38	1.23
250	5.75	4.04	3.87	32.43	20.45
	31.34	10.45	29.97	45.80	33.69
	5.89	3.42	5.89	6.82	6.01
500	2.79	3.49	3.70	26.36	16.40
	33.52	16.51	38.41	54.70	43.92
	18.52	10.81	18.52	20.61	18.06
1000	1.51	2.46	2.19	17.62	12.33
	52.32	27.85	55.71	68.71	59.31
	42.41	24.19	42.41	46.40	42.05
2500	0.61	0.93	0.73	7.04	5.06
	79.97	52.17	80.89	87.31	80.30
	75.91	50.75	75.91	78.86	73.65
5000	0.27	0.48	0.33	3.52	2.40
	90.31	70.12	90.96	93.56	88.92
	88.40	64.92	88.40	89.27	85.54

Table 3. Percentage Of Best Approximation. For Each Parameter, The First Line Indicates The Number Of Times (Given In Percentage) When The Method Was The Only One To Reach The Best Approximation. The Second Line Gives The Number Of Times When The Method And At Least Another One Reached The Best Approximation. Last Line Gives The Number Of Times When The Method Found An Exact Solution (I.E. Global Harmony Error Is Equal To 0)

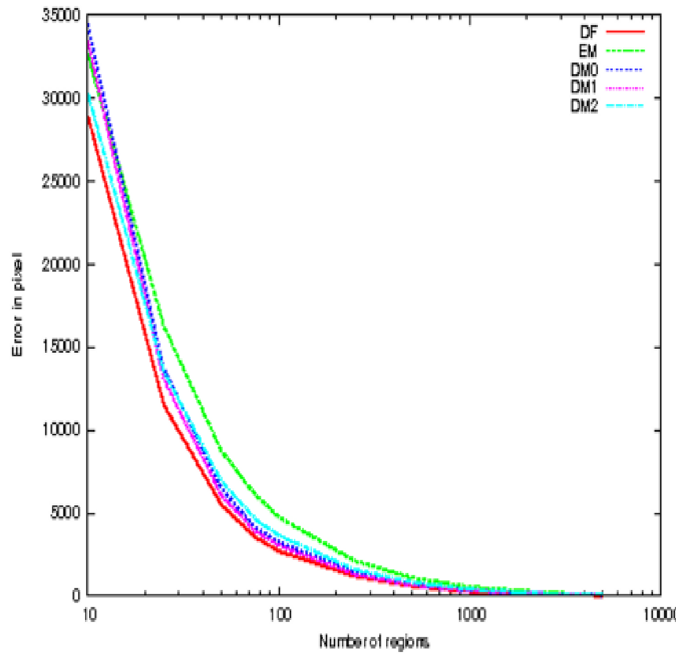

Figure 9. Average value of the global harmony error. ”Number of region” axe is displayed with a log scale. The decreasing filling method (DF) gives the lowest global harmony error in average for each parameter

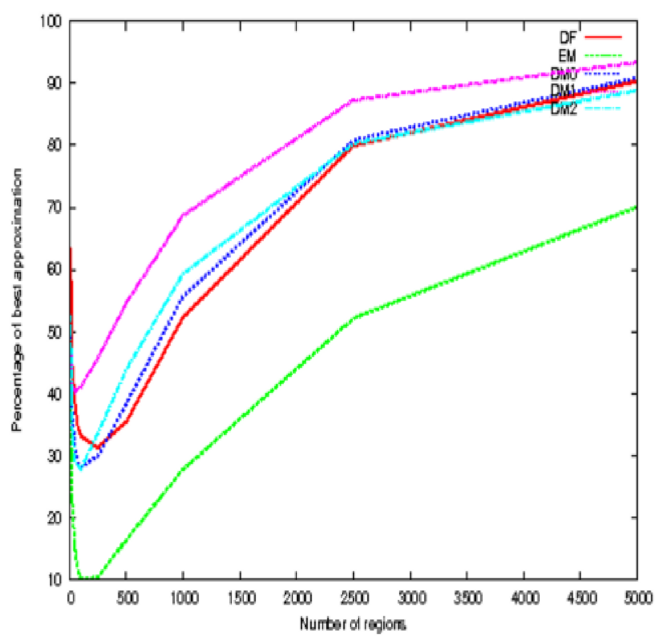

Figure 10. Graphical representation of the percentage of best approximation when several methods converged to the minimal global harmony error

	DP	DF	EM	DM0	DM1	DM2
Number of exact solutions found	10	2	6	2	2	4
Percentage of cases where best possible approximation is reached	100.00	50.35	32.55	41.39	38.74	40.45
Percentage of relative average global harmony error	0.00	1.26	2.74	3.49	3.05	1.81

Table 4. Absolute Efficiency Of Approximation Methods For Parameter 10. The Column DP Presents The Results Obtained Using The Dynamic Programming Method. THESE Results Are The Best Possible Colorizations Which Can Be Obtained With Respect To Global Harmony Error Minimization

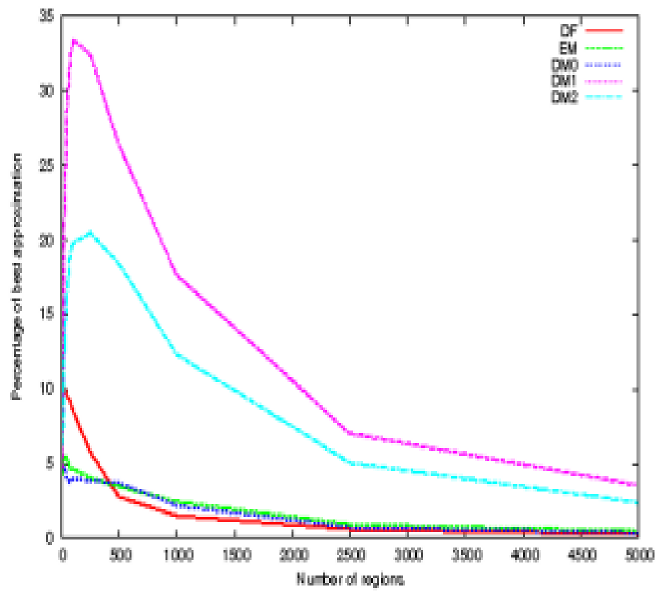

Figure 11. Graphical representation of the percentage of best approximation when a method reached the minimal global harmony error alone

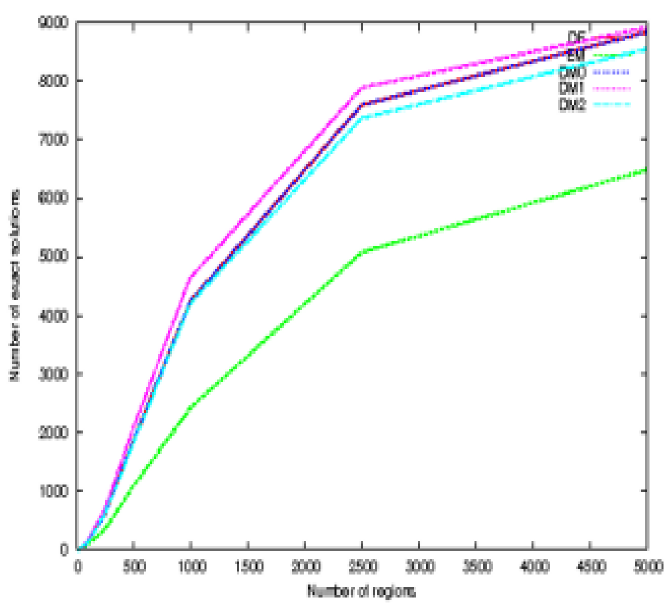

Figure 12. Evolution of the number of exact solutions reached with respect to the number of regions

	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 5. Computation Time Is Given In Seconds

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 14 and 15.

We present in Table V the computation time for some examples. The computations were done on a PC with 2.5Ghz frequency and 3GB 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.

At first, we recommend to 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 steps of solution reconstruction) and memory consuming (depending on the number of regions).

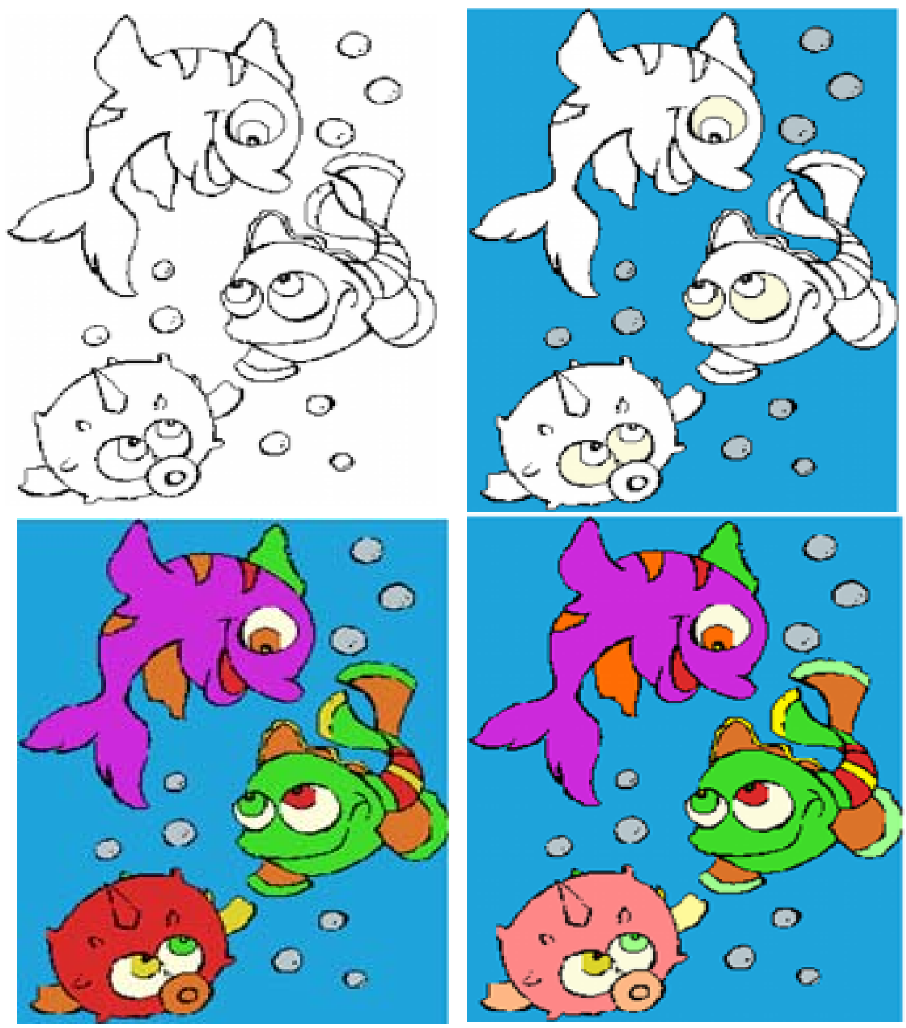

Figure 13. From left to right, top to bottom: sketching (260×364 pixels); partial colorization; harmonization with CS = (Y,O,R,G, V) and DF method which has produced an exact solution; artist refinement

Figure 14. This 798 × 885 pixels image illustrates different strategies leading to different colorizations. The harmonization is defined by CS = (O,R). From left to right and top to bottom: original input image, DP, DF, DM0, DM1 and DM2 methods where = 0 pixel for DP and = 1 pixel for the other.

Figures 16 and 17 present all steps of our work-flow. The artist makes 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 it is worth noting that colors can be shaded or changed while the sector definition is preserved.

Figure 18 presents a Chinese landscape. This image contains 704000 pixels. A sketching and a partial colorization are shown. Then we propose two results. First, a result with the error minimization method (EM) where $\Delta = 23880$ pixels and the artist refinement of this result. Then a result with the displacement maximization method (DM) where $\Delta = 2$ pixels and the refinement added by the artist. This example demonstrates that it is more complicated to obtain good approximation when there are some huge regions. However, it also depends on the desired proportions.

7. Conclusion And Perspectives

In this paper, we present 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 which provide, with reasonable time/memory resources, exact solutions if they exist or good approximations otherwise. We describe several methods based on those strategies. We conduct 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 have different colorizations available, respecting the specified harmony. After the harmonization process, the user may refine hue, saturation and value for better visual appearance.

Figure 15. From top to bottom and left to right: Sketching (651 × 560 pixels); partial colorization; harmonization with CS = (Y,O,R,G,B, V) and DF and EM methods and respectively at the bottom, their artist refinement with shading

Figure 16. Sketching (570×806 pixels); partial colorization; harmonization with CS = (Y,O,R, V,B) and DP method; artist refinement with shading

Figure 17. Sketching (500 × 714 pixels); partial colorization; harmonization with CS = (O,R,G, V) and DP method; artist refinement with shading

Figure 18. From top to bottom and left to right: Sketching (1000 × 704 pixels); partial colorization; harmonization with CS = (Y,O,G) and EM method; artist refinement; harmonization with CS = (Y,O,R,G) and DM method with $\omega = 0$; artist refinement.

Several perspectives can be considered.

First 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.

The most important perspective comes from the two previous ideas. We would like to create a language which allows the user to create his own rules to add different constraints on the regions or between the regions. For example, it would be interesting to couple different regions belonging to the same object. In that way, they could have the same color. Or, we can also imagine that each region of a group might have a different color: for instance, whatever their color, a group could be warm colors. Many kinds of harmony and many constraints may be added to this tool to complete the possibilities.

7. Acknowledgment

Authors would like to thank the reviewers for their valuable comments and remarks.

References

[1] Itten, J. (1961). *Kunst der Farbe*, R. O. M. Verlag, Ed. Ravensburg: Otto Maier Verlag.

[2] Sauvaget., Boyer, V (2010). Harmonic colorization using proportion contrast, *In*: *Afrigraph*, p. 63–72.

[3] von. Goethe, J. W. *(1810) Zur Farbenlehre*. Stuttgart: Cotta'sche Buchhandlung, 1810.

[4] Chevreul, M. E. *(1839). De la loi du contraste simultan´e des couleurs et de l'assortiment des objets color´es*. Paris, 1839.

[5] Maxwell, C. (1960). On the theory of compound colours," in *Philosophical Transactions*, no. 150, 1860, pp. 57–84.

[6] Munsell, A. H. (1921). *A Grammar of Colors*, M. . S. P. Company, Ed. Mittineague : Strathmore Paper Company.

[7] Ostwald, W., Birren, F (1969). *The Color Primer*, N. Y. V. N. R. Company, Ed. New York: Van Nostrand Reinhold Company, 1969.

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

[9] Nack, F., Manniesing, A., Hardman, L (2003). Colour picking: the pecking order of form and function, *ACM Multimedia*, p. 279–282.

[10] Westland, S., Laycock, K., Cheung, V., Henry, P., Mahyar, F (2007). Colour harmony, *In*: *SDC : Society of Dyers and Colourists*.

[11] Ou, L., Luo, M.R (2006). A colour harmony model for twocolour combinations, *In*: *Color Reasearch and Application*.

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

[13] Sawant, N., Mitra, N. J (2008).Color harmonization for videos, *In*: ICVGIP, p. 576–582.

[14] S'ykora, D., Burinek, J., Ra, R (2004). Unsupervised colorization of black-and-white cartoons, *In: Proc. NPAR*, 2004, p. 121–127.

[15] Lezoray, O., Ta, V.-T., Elmoataz, A (2008). Nonlocal graph regularization for image colorization, *In*: *ICPR*, 2008, p. 1– 4.

[16] Qu, Y., Wong, T.-T., Heng, P.-A. (2006). Manga colorization, *In*: *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006, p. 1214–1220. [Online]. Available: http://doi.acm.org/10.1145/1179352.1142017

[17] Sauvaget, C., Manuel, S., Vittaut, J.-N., Suarez, J., Boyer, V (2010). Segmented Images Colorization Using Harmony, in *Sixth International Conference on Signal-Image Technology & Internet-Based Systems* (SITIS)

[18] Lagoudakis, M. G. (1966). The 0-1 knapsack problem – an introductory survey, Tech. Rep.