

High Contrast Color Sets under Multiple Illuminants

Simone Bianco and Andrea G. Citrolo

University of Milano-Bicocca,
Viale Sarca 336, ed. U14, 20126 Milano, Italy
{simone.bianco, andrea.citrolo}@disco.unimib.it
<http://www.ivl.disco.unimib.it>

Abstract. In the modern digital society colors are often used to encode information. Nevertheless the selection of set of colors that maximizes class discriminability for nominal coding is a non-trivial problem. In this work we compare four different heuristics for the selection of sets of colors with fixed cardinality and maximum dissimilarity. The performance of each algorithm is evaluated both on single and multiple illuminants, on a sample of 1268 colors from the Munsell atlas, using ΔE_{76} euclidean metrics on the perceptually uniform CIE $L^*a^*b^*$ space. Results are presented for color sets with cardinality up to 25.

Keywords: High contrast color set, greedy algorithm, simulated annealing, genetic algorithm, local search.

1 Introduction

The pervasiveness of color devices has brought a rapid increase in the number of applications in which color is used to convey information. At least two different coding can be performed through the use of color set: nominal coding that is used to represent unordered classes, and ordinal coding that is used to represent a scale of values. The perceptual process associated to the decoding of categorical images is complex and relies on many variables, making it very hard to select a proper set of colors for a satisfactory coding when more than six or seven colors are required.

Several heuristic procedures have been proposed to define high-contrast sets of colors. Kelly [8] has conceived a list of 22 maximally contrasting surface colors, such that each color of the list is maximally different from the one immediately preceding it. In 1982, Carter and Carter [1] formulated the first algorithm to compute easily discriminable sets of colors. Several authors [9–13] have devised algorithms that can also fulfill a number of ergonomical requirements. In 1995, Campadelli et al. presented an abstract formulation of the problem of selecting high-contrast color sets [14], defining it as a combinatorial optimization problem on graphs.

More recently Carter et al. [15] have studied the ability of different metrics and several color spaces to enhance the discriminability of small visual targets with

ultra-large color differences. Glasbey et al. [7] proposed a greedy method for the selection of set of colors for categorical image and showed that its performance is comparable with that of a simulated annealing algorithm. Leonard et al. [16] proposed an algorithm for generating color scales for both categorical and ordinal coding; their method uses a positional space partition strategy to generate a lightness scale and then applies the method developed from Campadelli et al. [2] to select discriminable colors according to the lightness constraint. Rodriguez-Pardo and Sharma [17] proposed a dynamical solution to the problem using a hierarchical clustering followed by a simple truncation to select the desired number of high contrast colors at run time.

In this work we address the algorithmic aspect of the selection of high-contrast color sets under single or multiple illuminants, and present an effective algorithm based on Local Search, which is fast enough to be used in real time applications. The sets are extracted from the Munsell atlas and have cardinalities from 2 to 25. The method is also generalized to cope with multiple illuminants simultaneously. The results are compared with those obtained by three different heuristics for the selection of sets of colors with fixed cardinality and maximum dissimilarity in the state of the art.

2 Problem Formulation

According to Carter and Carter [1], and Campadelli et al. [2] selecting a subset of highly contrasting colors from a given range of colors means choosing a subset such that in it the minimal distance among all possible couples of colors is maximal. Formally, let $C = \{c_1, \dots, c_n\}$ be the given set of n colors, and K any subset of C with cardinality $|K| = k \ll n$. Denoting with \mathcal{C} the set of all possible subsets of C with exactly k elements, by d_{ij} the distance between color c_i and color c_j , the problem can be formulated as follows:

$$S = \max_{\mathcal{C}} \min_{c_i, c_j \in K} d_{ij} \quad (1)$$

The solution S thus satisfies the Maximal Dissimilarity among Colors (MDC) criterion [2]. It has been shown [2] that this combinatorial optimization problem is NP-complete.

3 Algorithms Compared

In this section the algorithms compared are described. They are the Greedy Algorithm, Genetic Algorithm, Simulated Annealing, and the proposed algorithm based on Local Search.

3.1 Greedy Algorithm

The Greedy Algorithm follows the problem solving heuristic of making the locally optimal choice at each iteration [3]. It is a generative algorithm, which starts from

a starting color c_{s_1} and then adds one color at time until the solution S has the desired cardinality k . At each iteration the algorithm adds to the current solution S_i the color sample with the largest distance from all the samples already in the solution, i.e.:

$$S_{i+1} = S_i \cup \{c_{s_{i+1}}\} \tag{2}$$

where

$$c_{s_{i+1}} = \arg \max_{c_j \in C \setminus S_i} \left(\min_{c_i \in S_i} d_{ij} \right) \tag{3}$$

3.2 Local Search

This simple iterative heuristic is based on the concept of neighborhood. The neighborhood is built through the definition of a neighborhood function of the following type:

$$N_h(K) = \left\{ \tilde{K} \subset C : |c \in \tilde{K} \wedge c \notin K| = h, h < k \right\} \tag{4}$$

In this work the 1-neighborhood N_1 is used, which means that two solutions are 1-neighbors if and only if they differ for just one color. Starting from a random solution with k colors in each iteration, an exhaustive search over the 1-neighborhood of the current solution is performed: the solution that leads to the higher improvement according to the MDC criterion is then selected as the new current solution. If no improving solution is found, the program returns. To speed up the calculation of the optimal solution, a filtering is applied to the neighborhood of each candidate solution: only the pair of colors with the lowest dissimilarity in current solution are allowed for replacement (any replacement that does not alter the worst pair has no effect on the MDC value).

Let (c_1, c_2) be the pair of colors with the lowest dissimilarity in the current solution S , i.e.

$$(c_1, c_2) = \arg \min_{c_i, c_j \in S} d_{ij} \tag{5}$$

then the distances d_{1k} and d_{k2} are computed $\forall c_k \in C, c_k \notin S$. Among all the distances d_{1k} and d_{k2} only the colors k_1 and k_2 such that $d_{1k_1} > d_{12}$ and $d_{k_2 2} > d_{12}$ are used to form the new candidate solutions S^* , which are 1-neighbors of the current solution S .

The best of these neighbors is selected as the new current solution.

3.3 Simulated Annealing

Simulated Annealing (SA) [6] is a stochastic algorithm that uses the concept of neighborhood but it allows a wider sampling capacity adding the possibility to move toward solutions that degrade the quality with respect to the current candidate solution. Given the current solution S_j at iteration j , the candidate solution S_{j+1} at the next iteration is randomly selected from the 1-neighborhood of S_j . The acceptance rule for S_j as the current solution, is defined as follows:

$$P(S_{j+1}) = \begin{cases} 1 & \text{if } F(S_{j+1}) \geq F(S_j) \\ e^{-\frac{F(S_j) - F(S_{j+1})}{kT}} & \text{if } F(S_{j+1}) < F(S_j) \end{cases} \quad (6)$$

At each iteration a random number is extracted from an uniform distribution in $[0, 1)$ and compared with $P(S_{j+1})$ and if it is smaller than $P(S_{j+1})$ the neighbor S_{j+1} is accepted as the new current solution.

The parameter T (usually called temperature) is used to modulate the stringency of the acceptance rule during the search process. The aim of stringency modulation is that of escaping local minima through an alternation of local search phases ($T \approx 0$) and quasi-random search phases ($T = T_{max}$). The algorithm starts with $T = T_{max}$ and after a fixed number of iterations or accepted solutions, T is decreased by a fixed quantity dt until it reaches 0. Then the cycle restarts with $T = T_{max}$. The values of the parameters T_{max} and dt are the major determinants in the performance of the algorithm: the value T_{max} has to be great enough in order to escape local minima in few iterations, on the other hand if the temperature decrease is too slow there is a significant probability of missing interesting regions of the search space. For the SA we used 1-neighborhood with uniform random selection between the neighbor solutions that replace one of the colors of the worst pair, without any preprocessing or exhaustive search. If there is a given number of consecutive refuses, a 2-neighborhood is generated replacing one of the colors of the worst pair and one other random color in the solution.

3.4 Genetic Algorithms

Genetic Algorithms (GAs) [4, 5] are the oldest and most known kind of evolutionary algorithms. Their peculiarity is that potential solutions that undergo evolution are represented as fixed length strings of characters or numbers. The iterative process of GAs can be summarized by the following pseudocode:

- Generate a population P composed of an even number N of individuals.
- Generation := 0.
- Repeat until a termination condition is satisfied:
 - Calculate the fitness of all the individuals in population P .
 - Create a new empty population P' .
 - Repeat until population P' is composed of exactly N individuals:
 - * Select two individuals i_1 and i_2 from population P using the chosen selection algorithm.
 - * Perform the crossover between i_1 and i_2 with probability p_c , and let j_1 and j_2 be the offspring (if crossover is not applied, let $j_1 = i_1$ and $j_2 = i_2$).
 - * Mutate each character of j_1 and j_2 with a certain probability p_m , and let k_1 and k_2 be the offspring.
 - * Insert k_1 and k_2 into population P' .
 - Perform the copy: $P := P'$ and delete P' .
 - Generation:=Generation+1.

Examples of termination conditions are: a predetermined number of generations or time has elapsed, a satisfactory solution has been found, or no improvement in solution quality has been taking place for a predetermined number of generations. In some cases, another genetic operator is added to crossover and mutation: elitism, i.e., the copy of the best individual unchanged into the newly generated population at each generation. In that case, either an odd number of individuals N is chosen or another individual in P is selected, mutated, and inserted into P' at each generation.

4 Experimental Results

In this work the Munsell Atlas has been used as the set C , which is composed of $n = |C| = 1269$ color samples. The algorithms described in Section 3 have been used to select from the set C , the subset which maximizes the MDC criterion. The subset extracted has cardinalities ranging from 2 to 25 colors. The distance used is the ΔE_{76} , which is the Euclidean distance in the CIE $L^*a^*b^*$ color space. Two different experiments are run: in the first one, only one illuminant has been considered. In the second one, the subset extracted had to maximize the MDC criterion under three different illuminants simultaneously. In the first experiment, the CIE $L^*a^*b^*$ color values of the samples have been computed from the spectral reflectances of the color samples using the XYZ color matching functions and the CIE D65 standard illuminant. In the second experiment, together with the CIE D65 illuminant, also the CIE A and CIE F2 standard illuminants have been considered.

Table 1. Algorithm characteristics

Algorithm	Deterministic	Population-based	Dependence on starting point	speed
Greedy	yes	no	yes	very high
LS	yes *	no	yes *	high
GA	no	yes	no	slow
SA	no	yes	no	slow

* LS is deterministic once the starting point is fixed.

4.1 Single Illuminant

The minimum distance among the samples in the color sets extracted by the different algorithms tested are reported in Table 2. They are computed under the CIE D65 standard illuminant. For the greedy algorithm the result reported is the best among all the results obtained starting from each different sample in the set C . Also for the other methods, the result reported is the best obtained over 100 independent runs. For comparison, we have also reported the distances of color palettes available in Color Brewer 2.0 (<http://colorbrewer2.org/>).

Table 2. Best solutions found by the different methods considered. The values reported are the minimum distance over the selected sets. They are relative to the colors under the CIE D65 illuminant.

Set cardinality	Greedy Algorithm	Local Search	Genetic Algorithm	Simulated Annealing	Color Brewer
2	128.67	128.67	128.67	128.67	116.60
3	91.12	96.41	96.41	94.31	92.86
4	78.74	78.74	77.91	78.29	63.84
5	68.99	68.99	65.34	68.90	47.11
6	58.41	61.44	59.02	58.89	40.97
7	53.52	55.61	52.76	51.35	39.32
8	48.55	52.73	47.58	47.58	36.70
9	46.69	48.62	42.37	43.83	32.74
10	43.62	45.25	38.67	39.64	24.12
11	41.33	42.58	35.28	37.06	23.64
12	39.67	42.56	33.45	35.20	23.64
13	37.13	39.04	31.46	33.07	n.a.
14	36.15	37.94	29.40	31.33	n.a.
15	33.70	36.52	27.40	30.85	n.a.
16	33.55	35.04	25.73	28.95	n.a.
17	31.70	33.47	24.53	28.54	n.a.
18	30.65	32.61	24.35	27.56	n.a.
19	30.06	32.15	22.78	26.37	n.a.
20	29.66	31.15	22.32	25.43	n.a.
21	28.53	30.39	20.50	25.11	n.a.
22	27.85	29.55	19.99	24.49	n.a.
23	27.02	28.78	19.42	23.69	n.a.
24	26.73	27.77	18.27	23.16	n.a.
25	26.35	27.47	18.61	22.35	n.a.

The best sets found are also reported in Figure 1 where they are plotted in the sRGB color space under the D65 illuminant.

From the results reported in Table 2 it is possible to see that the best results are always obtained with the Local Search (LS) algorithm. It is also possible to notice that the Greedy algorithm gives good results, and that the improvement of the LS over the greedy reduces as the number of colors in the subset increases.

The results of the Simulated Annealing could certainly improve if a larger number of iterations is selected. The number of iteration used has been selected in order to solve the optimization problem in reasonable time.

Genetic algorithm has shown to perform poorly on this problem. This could be surprising considering that GAs are among the best heuristics for many well known hard optimization problems. In this case the reason can be founded in the structure of the solutions that are fully connected graph: this means that the fitness according to MDC depends on the shortest connection. Since this connection can be found between any pair of colors in the solution, the preservation of good solutions relies on the preservation of long schemes that is an unlikely event according to the schema theorem [5]. To partially overcome this limitation we built a positional structure in the solutions so that each position maps a given portion of the chromatic plane: in this way short scheme should describe dissimilarity relation between sets of similar colors. Presented results are relative to this implementation.

4.2 Multiple Illuminants

For the experiment using multiple illuminants, the cost function reported in eq.1 is modified as follows:

$$S = \max_C \min_{\text{illuminants}} \min_{c_i, c_j \in K} d_{ij}, \quad (7)$$

i.e. the solution found maximizes the minimum distance between the worst couple in the set under the illuminants considered.

The minimum distance among the samples in the color sets extracted by the different algorithms tested are reported in Table 3. They are the minimum distance under the three different illuminants considered: CIE D65, CIE A, and CIE F2. For the Greedy Algorithm the result reported is the best among all the results obtained starting from each different sample in the set C . Also for the other methods, the result reported is the best obtained over 100 independent runs. The results for the Color Brewer samples are approximated converting the sRGB values to the CIE XYZ color space and using the BS-CAT [18] to change



Fig. 1. Best solutions found by the different methods considered. The colors selected are rendered in sRGB under the CIE D65 illuminant.

Table 3. Best solutions found by the different methods considered. The values reported are the minimum distance over the selected sets over the three illuminants considered.

Set cardinality	Greedy Algorithm	Local Search	Genetic Algorithm	Simulated Annealing	Color Brewer *
2	126.77	126.77	126.77	126.77	113.17
3	76.80	77.95	75.92	75.92	73.91
4	69.98	69.98	66.60	69.98	58.53
5	55.35	55.35	55.35	54.19	40.97
6	49.09	50.87	48.58	48.72	40.97
7	44.89	46.66	43.88	41.90	39.32
8	42.90	44.91	37.78	37.98	34.31
9	37.39	40.46	34.95	34.17	32.74
10	36.57	37.53	31.08	32.33	24.12
11	34.23	36.38	28.01	30.34	23.64
12	33.14	34.73	25.74	28.55	23.64
13	31.32	32.45	24.69	26.97	n.a.
14	29.17	30.88	23.68	25.99	n.a.
15	28.22	29.82	21.02	24.67	n.a.
16	27.31	29.57	20.01	23.86	n.a.
17	26.89	28.26	18.79	23.94	n.a.
18	25.66	27.31	18.77	22.12	n.a.
19	25.35	26.39	17.90	21.55	n.a.
20	24.28	25.59	17.13	21.13	n.a.
21	23.20	24.81	16.58	20.51	n.a.
22	23.11	24.67	15.58	20.25	n.a.
23	22.46	23.80	14.46	19.30	n.a.
24	22.35	22.95	14.44	19.23	n.a.
25	21.25	22.63	13.66	18.95	n.a.

* the results are approximated using the BS-CAT [18].

the white point from D65 to A and F2. The best sets found with cardinality lower than 14 are reported in Figure 2, where they are plotted in the sRGB color space under the three illuminants considered. In Figure 3 all the best subsets found are plotted in the sRGB color space under the D65 illuminant.

From the analysis of the results reported in Table 3 it is possible to see that, similarly to what happens in Table 2 for the single illuminant case, the best results are always obtained with the Local Search (LS) algorithm. It is also possible

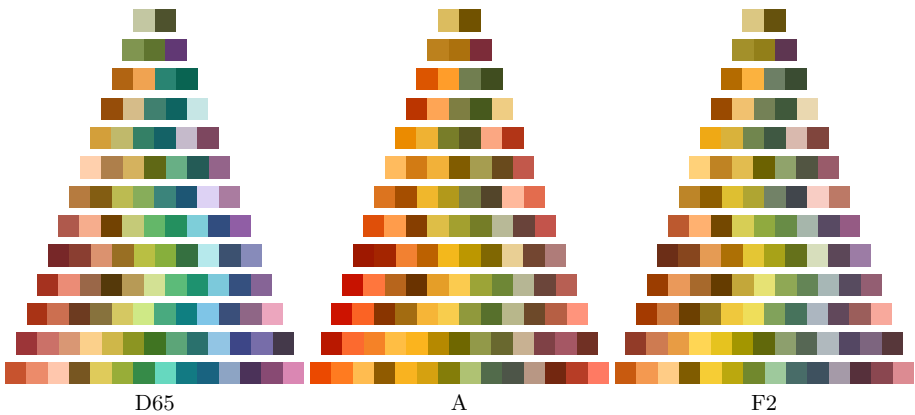


Fig. 2. Examples of the extracted subsets with cardinality up to 14 rendered under the three different illuminants considered

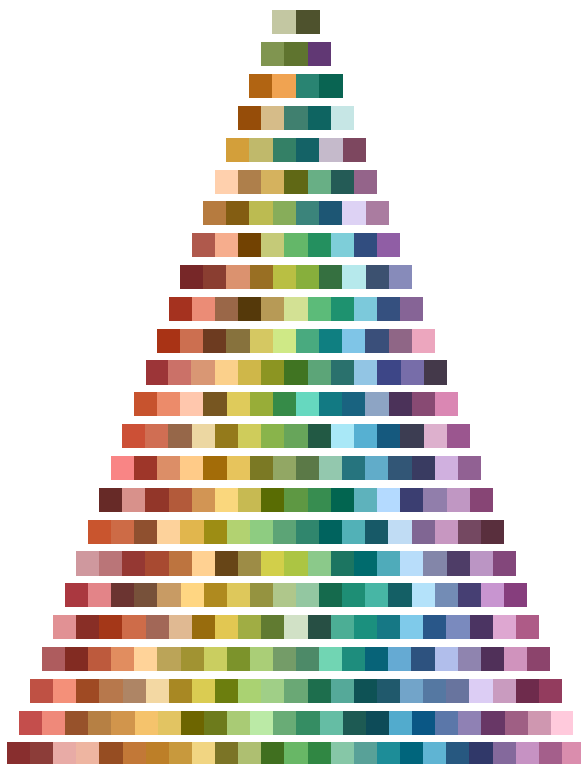


Fig. 3. Best solutions found by the different methods considered. The colors selected are rendered in sRGB under the CIE D65 illuminant.

to notice that the Greedy algorithm gives good results, and that the improvement of the LS over the greedy reduces as the number of colors in the subset increases. As one could expect, it is also possible to notice that in comparison with Table 2 the performance are lower.

5 Conclusions

In this work we have proposed a new fast method based on Local Search to select high contrast color sets. Its performance are compared with three different heuristics for the selection of sets of colors with fixed cardinality and maximum dissimilarity: the Greedy Algorithm, Genetic Algorithms, and Simulated Annealing. The performance of each algorithm is evaluated both under single and multiple illuminants, on the Munsell atlas, using euclidean metrics on the perceptually uniform CIE $L^*a^*b^*$ color space and maximizing the MDC criterion. Results are presented for color sets with cardinality up to 25.

Reproducible Results

We aim to make our research reproducible by everyone. The code used to select the samples of the high contrast color sets reported in this paper is therefore available online at <http://www.ivl.disco.unimib.it>.

References

1. Carter, R.C., Carter, E.C.: High-contrast sets of colors. *Applied Optics* 21(16), 2936–2939 (1982)
2. Campadelli, P., Posenato, R., Schettini, R.: An Algorithm for the Selection of High-Contrast Color Sets. *Color Research and Application* 24, 132–138 (1999)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT Press, McGraw-Hill (2009)
4. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York (1989)
5. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor (1975)
6. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. *Science* 220(4598), 671–680 (1983)
7. Glasbey, C., van der Heijden, G., Toh, V.F.K., Gray, A.: Colour Displays for Categorical Images. *Color Research and Application* 32, 304–309 (2007)
8. Kelly, K.L.: Twenty-two colors of maximum contrast. *Col. Eng.* 3, 26–27 (1976)
9. Carter, R.C., Carter, E.C.: Color coding for rapid location of small symbols. *Col. Res. Appl.* 13, 226–234 (1988)
10. De Corte, W.: Ergonomically optimal CRT colours for non fixed ambient illumination conditions. *Col. Res. Appl.* 13, 327–331 (1988)
11. De Corte, W.: Recent developments in the computation of ergonomically optimal contrast sets of CRT colours. *Displays* 11, 123–128 (1990)
12. Silverstein, L.D., Lepkowski, J.S., Carter, R.C., Carter, E.C.: Modeling of display color parameters and algorithm color selection. *Proc SPIE, Adv. Display Tech.* 6, 26–35 (1986)
13. Van Laar, D., Flavell, A.: A name based algorithm for generating maximally discriminable color sets. *J. Photo Sci.* 42, 33–34 (1994)
14. Campadelli, P., Mora, P., Schettini, R.: Color set selection for nominal coding by Hopfield networks. *Visual Comp.* 11, 150–155 (1995)
15. Carter, R., Huertas, R.: Ultra-Large Color Difference and Small Subtense. *Color Research and Application* 35, 4–17 (2010)
16. Breslow, L.A., Trafton, J.G., McCurry, J.M., Ratwani, R.M.: An Algorithm for Generating Color Scales for Both Categorical and Ordinal Coding. *Color Research and Application* 35, 18–28 (2010)
17. Rodriguez-Pardo, C.E., Sharma, G.: Adaptive Color Visualization for Dichromats using a Customized Hierarchical Palette. In: *Proc. SPIE: Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, vol. 7866, pp. 1–9 (2011)
18. Bianco, S., Schettini, R.: Two New von Kries Based Chromatic Adaptation Transforms Found by Numerical Optimization. *Color Research and Application* 35, 184–192 (2010)