Software for Simulating Dichromatic Perception of Video Streams

María José Luque,[1] Dolores de Fez,[2]* Pablo Acevedo

[1]Departament d'Òptica, Universitat de València, València, Spain

[2]Departamento de Óptica, Farmacología y Anatomía, Universidad de Alicante, Alicante, Spain

Abstract: We have designed a configurable stand-alone Matlab-based software to simulate dichromatic perception of video streams. The algorithm used is an extension for video streams of the "corresponding pair algorithm" by Capilla and coworkers for simulation of dichromatic perception of images. The software allows the user to upload a video sequence and to process it using different dichromatic color vision models and viewing conditions. The output video may be generated in different spatial and temporal resolutions and file formats. The functions for Matlab environment and a stand-alone application may be downloaded from the Repository of the University of Alicante. © 2013 Wiley Periodicals, Inc. Col Res Appl, 00, 000–000, 2013; Published Online 00 Month 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/col.21816

INTRODUCTION

Dichromatic color vision forms are characterized by the lack of contribution of one of the three cone-types, usually due to mutations and rearrangements in the genes encoding a given photopigment, that mainly affect their spectral sensitivities, although changes resulting in abnormal cone numbers, in non-functional photodetectors or in alterations in the topography of the cone mosaic have also been reported (see the recent review by Neitz and Neitz[1]). According to the missing cone contribution, dichromats can be categorized into three types: protanopes (lacking the long-wavelength L), deuteranopes (middle-wavelength M), and tritanopes (short-wavelength S). Even though no genetic differences between normals and dichromats have been found to affect the post-receptorial mechanisms, the missing cone input would cause one of the chromatic mechanism to become non-opponent (the Red–Green mechanism in protanopes and deuteranopes and the Blue-Yellow in tritanopes).[2–4] The loss of a cone type causes the subjects to confuse colors that differ only in the excitation of the class of cones that they lack and the loss of a chromatic mechanism limits their range of color perceptions. For instance, protanopes and deuteranopes perceive colors as either blue or yellow, with different saturation and brightness[5–7] (but these experimental results may be conditioned by the way the experiments were conducted[8]). Under these conditions, processing of color-based information is seriously impaired. The simulation of how a subject with variant vision perceives a given scene constitutes an interesting image and video processing application that could help to understand the problems these subjects encounter.

Simulating dichromatic perception of images is not a novel idea and different algorithms may be found in the literature.[2,9,10] One of these algorithms has been used in a free online application, Vischeck,[11] which simulates colorblind vision by means of software based on the Brettel, Vienot and Mollon algorithm.[2] Once the user loads the problem image in the application, protanopic, deuteranopic and tritanopic perceptions are simulated. The main limitations of this application are that neither adaptation conditions nor the characterization of the visualization device can be specified by the user. Moreover, only a single image can be processed at a time.

Our research team has developed in the last few years an algorithm[12–15] that describes to normal subjects how subjects with variant vision perceive stationary images, in such a way that the user may (1) employ and compare different color vision models to explain a given anomaly, including the influence of observation conditions

*Correspondence to: Dolores de Fez (e-mail: dolores.fez@ua.es)

and (2) configure certain visualization device-dependent parameters. The different versions of this algorithm, which we explain in more detail below, have been implemented using a Matlab-based library, Colorlab,[16] developed in the Optics Department of the University of Valencia. This library contains functions for visualization device characterization, color generation, colorimetric transforms and color models.

Using these tools, we aim to develop an application, VideoProcessor, to simulate how color defective subjects perceive video signals, designed to be used both by expert and novice users. To this end, the code of the Matlab functions we have generated is provided, along with a stand-alone closed application. This application allows the user to simulate the appearance of the desired image or video file for the selected type of dichromat. The advanced user may configure certain parameters, including the choice of a particular color vision model, the adaptation and viewing conditions and the output video format.

Simulating How a Stimulus Appears to a Dichromat: The Corresponding Pair Algorithm

Basically, for a given image **T** (test), the corresponding pair algorithm[12–15] computes the image **S** (simulation), verifying that a subject with variant color vision (problem or variant subject) perceives **T** in the same way as **S** would be perceived by a normal (or reference) subject. If we have models to predict the perception of normal and problem subjects (denoted by m_n and m_p, respectively), the image **S** may be found by solving the following equation:

$$m_n(S) = m_p(T) \tag{1}$$

The quality of the predictions depends on how well the models used describe the visual system. Although in subsequent versions models incorporating both chromatic and spatial properties of the visual system have been proposed,[12–15] the algorithm was applied in the first place to simulate images perceived by dichromatic subjects by implementing Eq. (1) with color vision models working separately on each image pixel.[12,13] Any color vision model with both normal and dichromatic versions can be used, provided that the definitions and ranges of the color descriptors in each model are the same and that the normal version is invertible. To ensure that the models met the first of these two conditions, the dichromatic models were derived by changing the values of certain parameters of the normal color vision model. These changes included either cone substitution or cone nulling[3] and nulling the responses of certain opponent mechanisms, at different stages of the model.[17–20] Different dichromatic models derived from a particular normal color vision model were used to predict the results of a set of psychophysical experiments (such as chromatic thresholds, for instance), and by comparing these predictions with experimental data in the literature, we decided which

dichromatic version of a model predicted best the behavior of a particular dichromat.[12,13]

We must emphasize that Eq. (1) does not imply that the variant observer would perceive the simulation and the original image as identical, though if a unilateral dichromat is found, the normal eye seeing **S** and the dichromatic eye seeing **T** should perceive the same scene. Note that Brettel and co-worker's algorithm, use colors a dichromatic and a normal eye would perceive as equal as a basis to construct the simulated image.[2] Algorithms based on zone-models have indeed important limitations, since they do not fit all the experimental data on dichromatic vision,[12,13] they disregard spatial aspects and some colors do not have a generable corresponding pair.[12,13] However, under these conditions, they still provide a useful approximation of dichromatic vision, and at the very least they provide a good approximation of the colors that are confused by dichromats regardless of whatever hues they perceive.

In the software we present in this article, dichromatic color vision models have been obtained by modification of normal color vision models, using the hypotheses that, according to previous testing,[12,13] yielded the best performance, which be briefly describe below:

a. At the cone stage, we assume cone substitution for protanopes and deuteranopes, and cone nulling for tritanopes.[13] Under the cone substitution hypothesis, dichromacy arises from the substitution of a photopigment by another (M by L for protanopes, L by M for deuteranopes). The subsequent neural circuitry is assumed to be normal,[1,3] at least as far as the inputs to the post-receptorial mechanisms. The changes we have described are incorporated into the model by means of a matricial transform, M_C, which modifies the output of the cone stage of the normal model.
b. At the post-receptorial stages, we assume that one of the opponent chromatic mechanisms (red–green for protanopes and deuteranopes, blue-yellow for tritanopes) is nulled in some or all of the opponent stages of the model.[17–19] These changes are included into as many matrixes, M_O, as opponent stages of the model, which modify the output of these stages.

We must be aware of the fact that both the normal and the variant models we are using fail to explain certain phenomena.[13] The software we describe allows the user to choose the model that for a given problem yields the more accurate results.

METHODS

The VideoProcessor software is available at the Repository of the University of Alicante (RUA).[21] The Installer.exe file includes the Matlab Compiler Runtime – in case Matlab is not installed in the computer-. The Colorlab[16] and the video processing ffmpeg[22] libraries must also be installed. The user can choose the directory where the
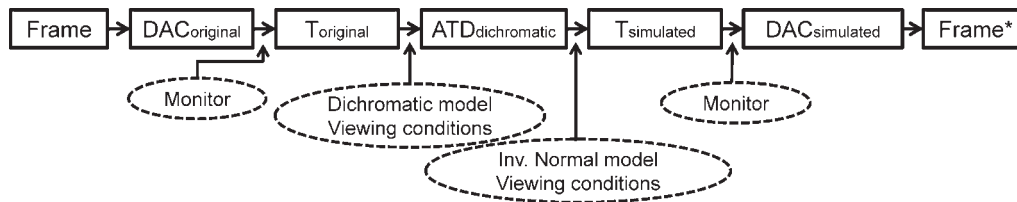
FIG. 1. The different stages of the transform simulating dichromatic color perception, as applied to a single frame of the video file.

application is installed –Documents and Settings\Matlab is the default - and is informed about the files being copied in the computer.

The ffmpeg library was introduced to avoid the limitations set by Matlab's memory-management with video signals and to increase the number of recognized and supported video formats. The functions in this library are used to extract the video frames that are subsequently loaded and processed using functions for Matlab. Other useful features in the ffmpeg library are the possibility of re-scaling and time re-sampling video sequences.

Simulating Image and Video Perception

Calling the VideoProcessor.m routine, a Matlab's Graphical User Interface (GUI) is opened, allowing the user to select the image or the video file to be processed, the type of dichromacy to be simulated, the color vision model used in the simulation and the adaptation conditions, when the chosen model includes adaptation mechanisms. The interface will be described in detail in the next section. Here, we deal with the different processing stages.

Once the software has loaded the desired video file, the user may choose between viewing a single frame (in which case the software calls the *convertone.m* function) or the complete sequence (the *convert*.m function is therefore called). *Convert.m* extracts the different video frames and saves them as temporary files, to await frame-by-frame processing. The different processing stages of a single frame, as outlined in Fig. 1, begin after loading the color-matching functions of a reference color space (CIE1931 is the default) and the colorimetric characterization of the visualization device (by default, data from a standard CRT monitor, contained in the std_crt.mat file, is used, though if users characterize their own device using Colorlab, the proper files may be loaded through the GUI's toolbar).

The DAC values of the frame must be converted to tristimulus values in the reference color space, using the colorimetric characterization of the device. The Colorlab function carrying out this transformation (*val2tri.m*) works with the image's color palette. For this reason, DAC true-color images must be first transformed to the indexed image + DAC color-palette format. The DAC palette is then transformed to the CIE1931 XYZ color palette, using the *true2pal.m* Colorlab function. To reduce processing time, the final palette contains 256 colors by default, though this restriction can be removed. The color

descriptors of the image in the chosen dichromatic color vision model are obtained from the XYZ tristimulus values using function *xyz2atda.m*. The XYZ tristimulus values of the simulation are obtained by applying the inverse of the normal model on the dichromatic descriptors, using *atd2xyz.m*. The colorimetric characterization of the color device is used again to convert, with function *tri2val.m*, the XYZ color palette of the simulation into DAC values. With the new color palette and the original indexed image we obtain, using *pal2true.m*, a true-color image for the new frame of the simulation, which is saved in a temporal file. The process is repeated for the next frame in the original video sequence and the final video file is created with sampling and format options specified by the GUI.

Colorlab's functions *xyz2atda.m* and *atd2xyz.m* include different linear and nonlinear color vision models: Hurvich and Jameson,[23] Ingling and Tsou,[24] Boynton,[25] different versions of Guth's ATD model[26–29] and Derrington, Krauskopf and Lennie's color space (DKL).[30] Certain models include user-defined parameters, depending on adaptation or observation conditions. The software works with the default values of these parameters, though the interface allows the advanced user to choose between four possible adapting stimuli: black, device's white point, the average of the scene or of the sequence previous to a given frame, and user-defined adaptation. Although certain models (such as the different versions of Guth's ATD model[26–29]) include adaptation mechanisms with weighted contributions of a test and a background, the software assumes that only the defined adapting stimulus contributes to the adaptation mechanisms.

User Interface

When designing the GUI's contents and applications we have grouped the different options according to the natural hierarchization of the processing stages. Our intention is to facilitate the task both for the user with a background in color vision and for the unspecialized user.

The GUI's (Fig. 2) separates the color configuration options from the pre-view and output file options, since the choices to be made in each of these three blocks are mutually independent. The last two blocks, in fact, do not determine the accuracy of the predictions, but increase the versatility of the software. A video-preview window has been added, along with the option to process single frames with the specified options to get an idea of how the final result will appear.
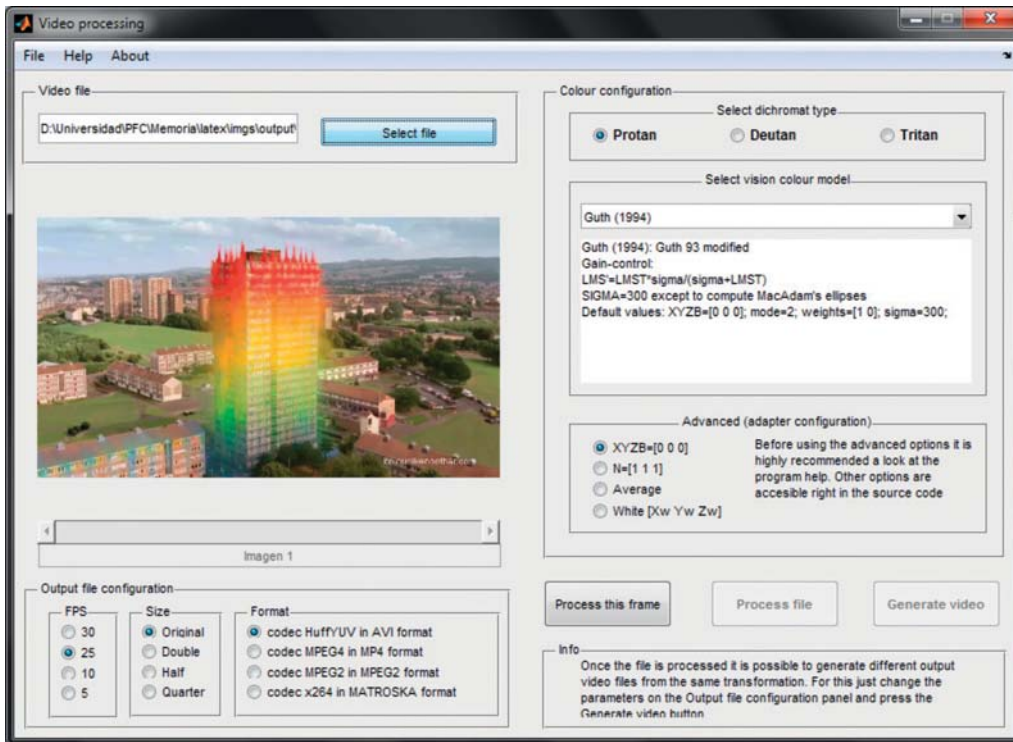
FIG. 2. Screen-shot of the user's interface, showing the different menus and options.

The left side of the GUI contains the file loading dialog, the preview window and the configuration options of the output file. The 'Select file' button opens a dialog to select the file to process. The full path appears to the left of the button. The preview window will show the video to process, allowing the user to navigate to individual frames through the slider. This window and the corresponding buttons remain inactive until a video is loaded.

The bottom part is dedicated to the configuration of the output video file. Two features had been implemented that can help to adapt to the user's need of accuracy and processing speed. One is the configuration of the frame rate per second. For a detailed visualization this value can be chosen between four predefined options: 30, 25, 10, and 5 fps (frames per second). Another feature is the video resolution or size of the video frames. There are four predefined options as well: Original, Double, 1/2 and 1/4. This option can be very helpful for processing high resolution video files since the time needed for the calculations may be very long. The last configuration option is the output video file format, just for compatibility.

The right side of the GUI contains all the options regarding color. First, a dichromat type has to be chosen: protan, deutan, or tritan. Next, there is the option to choose between different color vision models. The dropdown dialog lists all the available models the program has to process the video signal. Selecting a model modifies the information box underneath the dropdown dialog, where the characteristics of each model are specified (for experienced users). To ease the task for the non experienced user the vision color model chosen by default is

the most adequate to observe the changes produced in the perception of the different type of dichromats, according to the authors. As previously mentioned, another set of options has been added for the users with advanced color vision knowledge. This section gathers the options about the adapting stimulus, which acts over the perception of the scene/sequence. By default, no adapting stimulus is considered and the user can choose between three options: monitor white, the average of the previous frames computed individually and a user specified adapter that has to be specified in tristimulus values.

The processing buttons are placed at the bottom of the right side of the GUI. There is a button for processing a single frame that calls the algorithm in such a way that shows the resulting image for the specified options in a Matlab window. Another button processes the whole video. Once processed, an output video file can be generated with the specified video options. Different video files can be generated from the same colorimetric processing. The menu bar gives the option to load a personalized colorimetric characterization file of the monitor and also provides access to the help files of the program and the Colorlab libraries.

RESULTS

To exemplify the results, a single frame of a video sequence has been processed with two different color vision models (Boynton's[25] and Guth's ATD1995[29]) for the three types of dichromats (Fig. 3). Note that the two models coincide in their prediction of tritanopic

FIG. 3. From left to right, simulation of the perception of protanopic, deuteranopic and tritanopic subjects of the image shown at the upper-left part of the screen-shot in Figure 2. Results obtained with Boynton's (top row) and Guth ATD95 (bottom row) models.

perception and agree on the main characteristics of the images perceived by the red–green defectives -the fact that dichromats perceive two hues and the different relative brightness of reds and greens when perceived by protanopes and deuteranopes-. The particular hues perceived by red–green defectives depend however on the model. We have shown elsewhere[12,13] that the prediction derived from Guth's model is more accurate. A sample processed video can be downloaded from RUA.[30] The reader can access a demonstration of the use of the program and a sample processed video from RUA.[30,31]

This software has three important limitations: the visualization device, the storage space and the processing time. The images or the video sequence will be displayed in a particular monitor. CRT monitors present the advantage of having a standard characterization curve response, so that the default calibration file will not differ much from one monitor to another. But if the user aims to accurate color reproduction, it will be necessary to provide a colorimetric characterization of the display device. Said file has to be in the specific format the software uses, so it is recommended to take a look at the Colorlab help and follow the instructions to generate it. Not using the characterization of the particular viewing device will probably not mask the first order effects on the image, but the finer effects will be lost.

During the processing of each one of the frames, the JPEG file format is used. This format uses lossy data compression to reduce the size. This means that every time an image is compressed it will not be identical to the original one. It is possible to use other intermediate formats, TIFF for example. This format uses lossless data but processing is slow and file size big. The JPEG compression algorithm is based on the tolerance of the human visual system: we are more sensitive to changes in luminance than in chromaticity and, furthermore, we notice small changes in brightness in homogeneous zones better than in zones where the change is bigger, for example, on the border of objects. The compromise solution has been to use JPEG as the intermediate format, basing our choice in the human visual system tolerance.

As of the processing time, it has to be said that it is much longer than expected at the beginning. This is caused by some steps in the algorithm, such as the transformation of the color vision model descriptors to tristimulus values, which is done by minimization when the model's transform are not mathematically invertible. The minimization stage requires a loop that progressively approaches the desired values. The approximate time the algorithm takes to process one DVD size frame is between 1 and 4 seconds due to this limitation. Users have the possibility to access the source code and modify the different parts of the algorithm according to their needs. In particular, the Colorlab functions converting tristimulus values to DAC may be modified to suit different monitor types and the minimization algorithms used by different Colorlab functions could be optimized.

CONCLUSIONS

We have developed an open source application for the processing of video signals with the purpose of simulating the perception of dichromats. Users without knowledge in colorimetry and visual perception are just required to load a video. The advanced user can manipulate different parameters regarding the visualization device and the color vision models used in the simulation. Users interested in the source code can download the files for Matlab environment from RUA.[21] The authors will be grateful for any suggestion the users of the research community might give to improve the software.

1. Neitz J, Neitz M. The genetics of normal and defective color vision. Vision Res 2011;51:633–651.
2. Brettel H, Viénot F, Mollon JD. Computerized simulation of color appearance for dichromats. J Opt Soc Am A 1997;14:2647–2655.

3. Sharpe LT, Stockman A, Jägle H, Nathans J. Opsin genes, cone photo-pigments, color vision, and color blindness. In: Gegenfurtner KR and Sharpe LT, editors. Color Vision. From Genes to Perception. Cambridge University Press; 1999, p 3–51.

4. Viénot F, Brettel H, Ott L, Ben M'Barek, A, Mollon JD. What do color blind people see? Nature (London) 1995;376:127–128.

5. Judd DB. Color perceptions of deuteranopic and protanopic observers. J Res Natl Bureau Standards Res. Paper RP1922 1948;41:247–271.

6. Hendricks M, Ruddock KH, Waterfield WA. Spectral sensitivity functions of post-receptoral responses in human vision. J Physiol 1982;331:17–33.

7. MacLeod DIA, Lennie P. Red–green blindness confined to one eye Vision Res 1976;16:691–702.

8. Broackes J. Unilateral colour vision defects and the dimensions of dichromat experience. Ophthal Physl Opt 2010;30:672–684.

9. Viénot F, Brettel H, Mollon JD. Digital video colourmaps for checking the legibility of displays by dichromats. Color Res Appl 1999;24:243–252.

10. Machado GM, Oliveira MM, Fernandes LAF. A physiologically-based model for simulation of color vision deficiency. IEEE Transactions Visualization Comput Graphics 2009;15:1291–1298.

11. Dougherty B, Wade A. Available at: http://www.vischeck.com/ (Visited January 2013).

12. Capilla P, Díez-Ajenjo MA, Luque MJ, Malo J. The corresponding pair procedure: a new approach to simulation of dichromatic colour perception. J Opt Soc Am A 2004;21:176–186.

13. Capilla P, Luque MJ, Díez-Ajenjo MA. Looking for the dichromatic version of a colour vision model. J Opt A: Pure Appl Opt 2004;6:906–919.

14. Luque MJ, Capilla P, de Fez MD, García-Domene MC. Images perceived after chromatic or achromatic contrast sensitivity losses. Optom Vis Sci 2010;5:313–322.

15. Capilla P, Luque MJ, de Fez MD, García-Domene MC, Díez-Ajenjo MA. Simulating images seen by patients with inhomogeneous sensitivity losses. Optom Vis Sci 2012;10:1543–1556.

16. Malo J, Luque, MJ. COLORLAB: A color processing toolbox for Matlab. Available at: http://www.uv.es/vista/vistavalencia/software/colorlab.html (Visited January 2013).

17. Romeskie M, Yager D. Psychophysical measures and theoretical analysis of dichromatic opponent response functions. Mod Probl Ophthalmol 1978;19:212–217.

18. Hendricks M, Ruddock KH, Waterfield WA. Spectral sensitivity functions of post-receptoral responses inhuman vision. J Physiol 1982;331:17–33.

19. Ruddock K H. Psychophysics of inherited colour vision deficiencies. In: Foster DH, editor. Vision and Visual Dysfunction. London: Macmillan;1991. p 4–37.

20. Dain SJ, King-Smith PE. Visual thresholds indichromats and normals; the importance of post-receptoral processes. Vis Res 1981;21:573–80.

21. de Fez MD, Acevedo P, Garcia C. A library for Matlab to simulate video as perceived by a dichromatic subject (colorblind). RUA, Universidad de Alicante. Available at: http://hdl.handle.net/10045/23471. (Visited January 2013).

22. Available at: http://ffmpeg.org (Visited January 2013).

23. Hurvich LM, Jameson D. An opponent-process theory of color vision. Psychol Rev 1957;64:384–404.

24. Ingling CR Jr, T'sou BH. Orthogonal combinations of three visual channels. Vision Res 1977;17:1075–1082.

25. Boynton RM. A system of photometry and colorimetry based on cone excitations. Color Res Appl 1986;11:244–252.

26. Guth SL. Model for color vision and light adaptation. J Opt Soc Am A 1991;8:976–993.

27. Guth SL. Erratum. J Opt Soc Am A 1992;9:344.

28. Guth SL. Further applications of the ATD model for color vision Proc SPIE-Int Soc Opt Eng 1995;2414:12–26.

29. Derrington AM, Krauskopf J, Lennie P. Chromatic mechanisms in lateral geniculate nucleus of macaque. J Physiol 1984;357:241–265.

30. de Fez MD, Acevedo P. Simulation video colorblind vision. RUA, Universidad de Alicante. Available at: http://hdl.handle.net/10045/18628. (Visited January 2013).

31. de Fez MD, Luque MJ. Use of the VideoProcessor software. RUA, Universidad de Alicante. Available at: http://hdl.handle.net/10045/25077 (Visited January 2013).