VTT INFORMATION TECHNOLOGY

RESEARCH REPORT TTE4-2004-4

MHP Project

Illumination-Adaptive Control of Color Appearance: a Multimedia Home Platform Application

Version 1.0-3

15.1.2004

Janne Laine and Mikko Kojo

Contact information

Janne Laine
VTT Information Technology
P.O. Box 1204, FIN-02044 VTT, Finland
Street Address: Metallimiehenkuja 10, Espoo
Tel. +358 9 456 5230, fax +358 9 455 2839
Email: janne.laine@vtt.fi
Web: http://www.vtt.fi/tte/staff/lne/

Mikko Kojo
VTT Information Technology
P.O. Box 12041, FIN-02044 VTT, Finland
Street Address: Tekniikantie 4 B, Espoo
Tel. +358 9 4566014, fax +358 9 456 7024
Email: email-address@vtt.fi
Web: http://www.vtt.fi/tte/

Last modified on 15 January, 2004
H:\MHP\IlluminationAdaptiveControlOfColorAppearance1.0-3.doc

Key Words display, monitor, color appearance, color appearance model, CIECAM02, viewing conditions, ambient illumination, ambient light, television, digital television, MHP, multimedia home platform, Java

Abstract

This report describes a Java application for adapting the output of a self-luminous display to varying ambient illumination conditions. The application models the effects of the lighting conditions around the display on the colors perceived by a viewer, and compensates for these effects by adjusting the displayed image. The goal of these adjustments, based mainly on previously developed methods, is to maintain the perceived colors as constant as possible despite the changes in ambient illumination. An MHP (Multimedia Home Platform) version of the application, initially developed on a PC, was built using a development tool for interactive digital television applications. This MHP version of the application was tested on a commercial set-top-box for digital television.

Preface

This report describes the results of the project "MHP Development Environment", which was done at Media research area of VTT Information Technology in 2003. The title of the project is slightly misleading, and it was often referred as "MHP Color", or "MHP Väri" in Finnish. The project was funded by VTT.

The goal of the project was to acquire an MHP development tool suitable for VTT's needs, and then use it to develop a real MHP application with significant research value. Java application for MHP environment for adapting the output of a self-luminous display to varying ambient illumination conditions was developed. The application was based on our existing software and knowledge on the subject. This document deals only with the application and its background research, not with the development environment.

The streering group of the project consisted of persons from VTT as well as Nokia. The following people were in the steering group: Helene Juhola (VTT), Olli Nurmi (VTT), Ville Ollikainen (VTT), Ari Siren (Nokia), and Caj Södergård (VTT, chairman). In addition to the authors of this document Otto Korkalo participated in the project.

Contents

1 Introduction 4

2 Viewing Condition Compensation 6
  2.1 Color Appearance Modeling: CIECAM02 6
  2.2 Flare Compensation 9
  2.3 Viewing Condition Compensation Transforms 11

3 A Demo Application 13
  3.1 PC Demo Application 13
  3.2 MHP Demo Application 17

4 Discussion and Conclusions 24

1 Introduction

We all have witnessed the effects of ambient light on our visual perception when watching TV in a typical living room, every corner of which is subject to daylight from the windows, some corners receiving more of it than others. In such a viewing environment the ambient illumination is bound to vary depending on the time of the day and weather outside (even if curtains are used to decrease this variation by blocking some of the daylight). Most of us would probably choose to watch TV in a dim rather than a bright room if given a choice. Direct sunlight hitting the TV screen can actually make it impossible see anything but faint shapes behind a shroud of light. In this case it is mainly the excessive intensity of *flare* (reflections of ambient light from the screen), compared to the intensity of image-forming light emitted by the TV itself, that ruins our viewing enjoyment. Even when ambient illumination is at a more reasonable level, it still has an effect on the colors of pictures we see on the screen.

Some of the effects are due to the physical changes caused by the aforementioned flare. Project some white light on the TV screen and every color on the screen gets brighter because of this additional light. This is not that bad for some colors, such as white which just gets brighter, but mixing white light with the image does decrease the colorfulness of saturated colors such as vivid reds and greens, bringing them closer to neutral grays. It also makes dark colors less dark, changing relationships between different colors in the image and thus playing havoc with the perceived contrast of the image. Add some chromatic light (colored light, in other words), such as light from a typical domestic light bulb (which is actually quite yellow, which may not be that easy to realize when your eyes are used to it, *adapted* to its yellowness), and you cause the color of the picture to shift not only towards lighter tones but also towards the yellow hue of the light.

All of the examples so far have dealt with changes in the actual physical color stimulus. It is easy to understand that when the light from the TV screen changes because of flare our perception of that light (the *color* we see) also changes. But the ambient light (and the viewing environment around the TV screen or some other media in general) can also influence our perception by changing our *state of adaptation*. The human visual system interprets physical visual stimuli (the light reflected or emitted by objects) in the context of the surrounding scene. There are many mechanisms in our eyes and brain, some of them physiological and some of them psychological, that help our visual system to adapt to a wide variety of viewing conditions, ensuring that our perception of color and other visual attributes is stable enough to allow us to recognize objects under very different lighting conditions.

The key here is to note that color is a perception. The light reflected by an object such as a printed picture on paper or the light emitted by a self-luminous medium such as a TV screen is the physical stimulus that stimulates our visual system to produce a response, a perception. That response of the visual system to a physical stimulus depends not only on the stimulus itself but also on the state of the viewer's visual system, which is affected by the total environment in the viewer's field of view around a given stimulus. Taking a stimulus from one viewing environment to another therefore changes its color appearance even if there are no physical changes in the stimulus. So, even if no flare was present, the picture on a TV screen would appear different when viewed in a dark room than when viewed in a room lit by daylight through the windows, for example.

These effects that change the perceived color of a color sample without affecting the sample itself physically are called *color appearance phenomena*. To put it another way: if physically identical samples seem to be of different color then there are one or more color appearance phenomena at play.

Successful reproduction of color in systems involving various media and/or various viewing environments requires that both flare and color appearance phenomena are somehow taken into account. Only when viewing conditions are constant (same media, lighting, etc.) it is safe to assume that matching color is achieved by simply reproducing the original stimulus as closely as possible. On the other hand, when viewing conditions change (someone turns off the lights when we are watching television, say) the stimulus *must* change in order to produce an approximately matching color perception between the original and the reproduction. Of course, often the final viewing conditions of a picture are not known beforehand and it is impossible to modify the picture afterwards to suit specific viewing conditions. In those cases the best way is to target the reproduction for certain typical viewing conditions and simply accept the fact that some color changes occur when the picture is viewed in different conditions. However, in the case of self-luminous displays such as TV-screens, or any media with a non-permanent picture, it is indeed possible at any moment to adjust the color reproduction to better suit certain viewing conditions.

Here we concentrate on systematic model-based adjustment of the output of self-luminous displays based on prevailing ambient lighting conditions. Use of mathematical tools known as *color appearance models* in automatically controlling the appearance of images shown on self-luminous displays under varying ambient illumination was studied in earlier projects [1-4]. In this project a Java application was developed that uses similar methods to control color reproduction on self-luminous displays. A basic version of the application, with five preset lighting condition categories and compensation for illumination-related color appearance phenomena using a color appearance model, was developed on PC platform. This version was transported to Multimedia Home Platform (MHP), the application standard for digital television. We wanted to study the capabilities of MHP and actual terminals in practice, and at the same to experiment with the MHP development environment purchased during the project. In the past we have studied the possibilities and boundaries of MHP in a more theoretical way [5]. A more comprehensive PC version of the application with tentative flare compensation, automatic estimation of lighting conditions using a web-cam, and possibility for the user to manually specify viewing conditions and display characteristics in a more detailed manner was also developed. In chapter two the general principles of color appearance modeling and flare compensation methods used in the application are presented. The application itself is described in chapter three.

2 Viewing Condition Compensation

2.1 Color Appearance Modeling: CIECAM02

Color appearance model is a mathematical construction that predicts the appearance of a given stimulus in given viewing conditions using numerical correlates for perceptual attributes such as lightness, chroma, and hue.

Rather than delving deep into the mathematics of color appearance modeling here, we briefly describe the principles of color appearance modeling and the main parts of the CIECAM02 model, which is used in the demo application. The details of the current CIE color appearance model, CIECAM97s, can be found in [6]. CIECAM02 is a revision of CIECAM97. CIECAM02 is going through the CIE acceptance process at the time of writing this. The use of CIECAM02 in this project was based on a draft obtained from the CIE Technical Committee 8-01 [7]. Preliminary information about CIECAM02 can also be found in conference papers [8-9].

CIECAM02 takes as its input a stimulus (or a sample) specified by its XYZ tristimulus values and a number of viewing condition parameters that describe the surroundings of the sample in a simplified manner. XYZ tristimulus values are triplets of numbers that define matching color samples for a typical observer in identical viewing conditions. Due to the tri-chromatic nature of human vision, which is based on three types of cone cells in the retinas of our eyes, color matches can be indicated by three numbers: samples with identical tristimulus values will appear identical, but only when seen in identical viewing conditions. So, these XYZ values indicate whether or not two samples match in color but give no information about their appearance.

CIECAM02 takes these XYZ values, along with viewing condition parameters, and transforms them to appearance correlates lightness, chroma, hue, brightness, colorfulness, and saturation, numerical values that predict the appearance of the samples in given viewing conditions. For our purposes three of the six appearance correlates are sufficient: lightness (J), chroma (C), and hue (h). Lightness indicates how much light the sample appears to reflect or emit relative to the light reflected by a perfectly white object in the scene. A perfectly black sample would have lightness of zero, while a perfectly white sample would have a lightness of 100. Chroma indicates how colorful a sample appears, judged in proportion to how much light a similarly illuminated white area appears to reflect. A neutral gray would have a chroma of zero and a vivid, deep red, green, blue, etc. would have a chroma of one hundred or more. Hue indicates the similarity of the sample's color to the unique hues red, yellow, green, and blue. Both lightness and chroma are relative appearance correlates in the sense that they are judged relative to the prevailing lighting level. When reproducing colors of original scenes or high dynamic range pictures on typical low dynamic range media, it is more feasible to aim at reproducing lightness and chroma than the corresponding absolute appearance attributes brightness and colorfulness: it is impossible to reproduce the brightness (perceived absolute amount of emitted light) of a shining white car on a sunny summer day using a display device whose maximum white intensity is a fraction of the intensity of the light reflected from the car's surface, for instance.

Let's go through the main parts of CIECAM02 on a general level now, pointing out their relevance to the task of adapting a TV-screen's output to given ambient illumination along the way. Other color appearance models can be broken down to more or less similar basic components.

As mentioned above, the color element whose appearance is being predicted (i.e. the sample or the stimulus) is encoded by calculating its XYZ tristimulus values. First these values are put through a *chromatic adaptation transform*. This transform models the process by which the human visual system adjusts its output according to the chromaticity of the prevailing lighting, so as to decrease the perceived variation of object colors with varying illumination. For instance, under a yellowish light source, which has less energy on the short wavelengths, the short wavelength sensitive ('blue) channel of the human visual system increases its sensitivity compared to the medium and long wavelength sensitive ('green' and 'red') channels, which somewhat balances the bias of the illumination. Even though the light reflected by a white object under a yellow light source is also yellow the increased sensitivity to the short wavelengths brings the response of the 'blue' channel close to that of the 'green' and 'red' channels. The result of this is that the white object appears *approximately* white even though the physical stimulus (light reflected by the object) is 'yellow'. In the case of TV-screens the ambient illumination draws the adaptation of the viewer away from the neutral equal energy stimulus, which has same amounts of energy on all visible wavelengths. A viewer adapted to a yellow ambient illumination, for instance, would perceive as bluish a TV screen that seemed perfectly neutral white to her in a dark room. If these kind of adaptation effects can be modeled accurately, the displayed image can be modified so that the elements meant to be neutral white in the picture would still appear neutral white under any ambient illumination and the overall color balance of the picture would also appear as intended. In practice this means showing a slightly yellow image to a person adapted to yellow lighting.

Although it is next to impossible to demonstrate various adaptation effects on a printed document (or even an electronic document) as they occur when viewing a self-luminous display under different types of light sources, let us look at some example pictures based on the results of previous psychometric experiments. In these experiments the observers compared the appearance of photographic pictures shown on a computer display under varying but controlled ambient illumination conditions. Below left is the original picture as it might have appeared to the observers when displayed in a dark room. The simulated picture in the middle shows an estimate of how the exactly same picture looked to the observers when seen under a typical yellowish office lighting. As discussed above, when a person's visual system adapts itself to yellow ambient illumination, the appearance of the displayed pictures shifts towards blue. The right hand side picture is the one we should display under the yellowish office light to produce a match to the original in a dark room. This adjusted picture was calculated using the CIECAM97s model but CIECAM02 would have produced similar results. Flare was eliminated in these experiments. Note that while some people might prefer the bluish water in the middle picture, preferred color reproduction and image enhancement is a separate issue. Although such "preference transforms" could very well be used in combination with viewing condition compensation transforms, here we are concerned only with the reproduction of the appearance of the original.

Figure 1. Effects of chromatic adaptation

For the CIECAM02 chromatic adaptation transform the XYZ values of the sample and the *adopted white point* are transformed to a 'cone response space'. The cone responses for the sample are balanced using the adopted white point. Basically the adopted white point specifies the white in the viewing environment. The adopted white point and a parameter indicating the degree of the viewer's adaptation to the adopted white point are the key parameters that control the output of the chromatic adaptation transform. In the adaptive display demo presented below both of them are set according to the assumed ambient illumination. The adopted white point is set equal to the XYZ values of the ambient light. The degree of adaptation parameter is set according to the absolute luminance of ambient white point and the relative level of ambient illumination compared to the intensity of the display white point. This parameterization could be developed further by adding a chromaticity-dependency to the degree of adaptation parameter, decreasing the modeled degree of adaptation to increasingly chromatic ambient light sources (there is some evidence that chromatic adaptation is less complete for highly chromatic light sources).

A *non-linear response compression* follows the chromatic adaptation transform. This part accounts for the dynamic non-linear cone response in the visual system. It is modeled by a hyperbolic function that converges to finite values at increasingly high and low intensities. This function can be compared to tone reproduction functions used in various imaging systems that include so called toe and shoulder portions to compress the output tone range at low and high intensities, respectively. The shape of the function is controlled by a parameter that is set according to the absolute luminance level of the viewing environment. When considering modeling of visual effects that depend on the absolute luminance level, it should be noted that the luminance level of a self-luminous display does not follow the luminance level of the surrounding viewing environment. Therefore, in the adaptive display application, the parameters that depend on the absolute luminance level are assigned different values at different parts of the CIECAM02 model. For instance, when estimating the degree of chromatic adaptation to ambient illumination it is the ambient luminance level that is used in the calculations, but when considering response compression it is the display luminance level that is used.

Both display and ambient luminance levels are considered when assigning values for the surround parameters that control the modeling of effects that depend on the relative luminance level of the surround compared to the luminance level of the display. These parameters come to play when the compressed cone responses are transformed (using certain scaling functions, which are not discussed here) to *appearance correlates* via temporary Cartesian representations. By setting the surround parameters to proper values the adaptive display application compensates for the reduction in perceived contrast that results from low relative surround luminance levels.

When the CIECAM02 model has thus been used to calculate the JCh (lightness, chroma, and hue) appearance correlates for image elements (each pixel of the image in practice) in certain reference viewing conditions, it can be reversed to calculate which XYZ tristimulus values must be used to reproduce the desired color appearance in given target viewing conditions. These target conditions are actually the current conditions in which the display is assumed to be viewed. The parameterization is done as discussed above, although the model itself is used in reverse.

2.2 Flare Compensation

In addition to serving as the viewing-condition-independent connection point for the corresponding color stimuli in color reproduction between two or more viewing conditions, the lightness, chroma, and hue appearance correlates of CIECAM02 also form a color space that is well suited for image manipulation. CIECAM02 represents the current best knowledge in color appearance modeling, and has close correlation to most available visual data, overcoming, for instance, some known limitations of earlier models like CIELAB. Thus it makes sense to perform image operations aimed at manipulating the appearance of a picture using the J (lightness), C (chroma), and h (hue) coordinates of CIECAM02.

In some ways flare compensation resembles perceptual gamut mapping: both try to help in reproducing the appearance of the original picture on output devices that are physically unable to produce some of the stimuli that make up the original picture. Flare limits the output gamut (the range or color stimuli that can be reproduced on a display) by increasing the luminance of black, and by de-saturating chromatic colors. Although the reflected light increases the intensity of all tones equally, the effect is relatively and perceptibly largest at the dark end, and because of this the contrast of the image is decreased. The physically measured contrast between display white and black cannot be changed by any image manipulation, but the perceived contrast, which is formed in a more complicated way as a function of all the tones present in the image can be manipulated. Simply put, the intent if flare compensation is to increase perceived contrast by making the transition from dark tones to light tones steeper. (Similarly, perceived chromatic contrast could be increased by making the transition from neutrals to high-chroma colors steeper. This has not yet been implemented in the display adaptor.)

The current tentative flare compensation algorithm used in the advanced mode of the PC version of the display adaptor application uses a sigmoidal contrast enhancement function described in [10] to re-map the lightness values of the image pixels. The shape and position of this function can be controlled by two parameters. The first parameter controls the slope of the approximately linear mid-region of the function that resembles an elongated letter 's' (see below). As the slope at mid-tones is increased the shadow and highlight values are compressed in equal amounts by the curving ends of the function ('toe' and 'shoulder'). It is this parameter that controls the amount of increase in perceived contrast. The second parameter can be used to move the function left or right to target compression more on the highlights or the shadows, respectively. Three lightness re-mapping curves are depicted in figure 2. The light gray curve is actually a straight line that maintains the original lightness values. The mid-gray curve increases contrast significantly, and the black curve results in a very large increase in contrast.

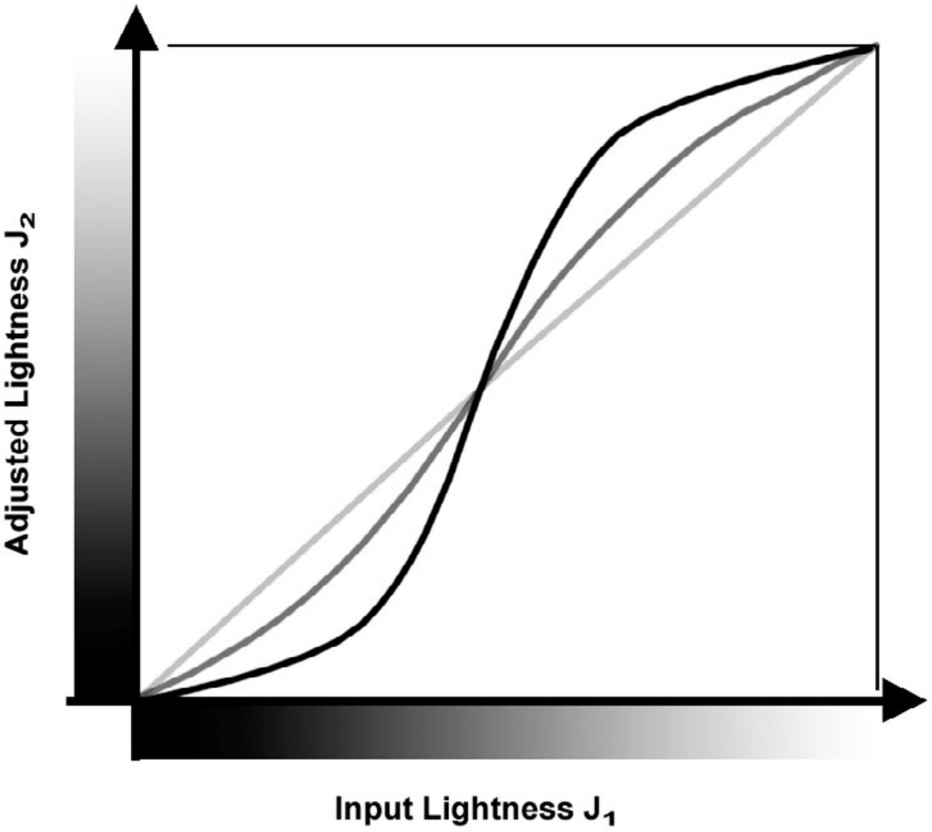


Figure 2. Lightness re-mapping curves for contrast enhancement

Figure 3 shows, clockwise from top left, an original picture and three adjusted images, whose lightness contrast has been manipulated using sigma values of 40, 30, and 20, respectively. As can be seen, contrast increases with decreasing sigma.

Figure 3. Different levels of flare compensation

The effect of flare on the colorimetry of the display should also be estimated and taken into account in the transform from the XYZ tristimulus values to RGB control signals of the display. The colorimetric characteristics of the display under the influence of flare can be modeled simply adding the XYZ values of estimated flare to the XYZ values of the flare-free output of the display. This has not yet been implemented in the display adaptor.

2.3 Viewing Condition Compensation Transforms

The complete viewing condition compensation transform is comprised of several steps that can be understood as transforms from one color space to another. The original image is in a certain RGB space, which can be either a standard RGB space or the RGB space of a specific device. These RGB values are transformed to XYZ tristimulus values using a transform derived from the colorimetric characterization of the device or standard color space. The simple version of the application uses the sRGB color space [11] as the default space for both the input image data and the display. In the comprehensive version this characterization can be read from an ICC profile. The MHP version uses sRGB primaries but allows the user to set gamma values based on visual evaluation of the tone reproduction of the TV screen. Note that the sRGB color space has been designed to be compatible with the implied reference display of the ITU-R BT.709-2 television standard, and is the default color space given in the MHP specification [12].

The XYZ values are then used as input to the CIECAM02 model. The model is used to transform these XYZ values to JCh coordinates, which represent the appearance of the displayed image in reference viewing conditions.

The flare compensation, when used, re-maps the lightness values to compensate for the reduction in perceived contrast caused by flare. Similar re-mapping could be applied to chroma values to compensate for the reduction in perceived chromatic contrast. In addition to flare compensation, other kinds of image appearance manipulations, such as sophisticated gamut mapping, could be done at this point (currently we use simple clipping of the final RGB values as our gamut mapping algorithm, when needed).

The JCh correlates, after possible manipulation, are transformed to corresponding XYZ values in current viewing conditions using the CIECAM02 model in reverse. Here the viewing condition parameters are set according to the assumed current viewing conditions.

Finally the corresponding XYZ values are transformed to RGB digital counts used to control the output of the computer display or TV screen. This transform is derived from the colorimetric characterization of the display.

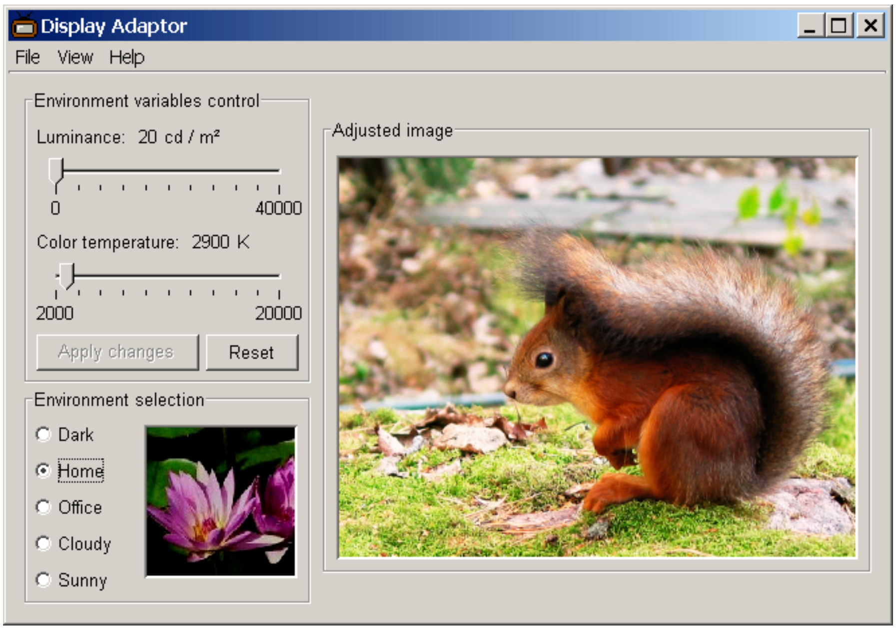
3 A Demo Application

3.1 PC Demo Application

The techniques described above were employed in an application intended for controlling the appearance of colors shown on self-luminous displays under varying ambient lighting conditions. Since the ultimate goal of the project was to have the application running on an MHP terminal the programming was done using the Java language. Java was a good choice also because of its platform-independent nature: a Java application can relatively easily be modified to run on a number of platforms, for example as a demonstration on a web page.

The application was first developed on a standard PC using the Java 2 Software Development Kit (SDK), version 1.4. The normal mode of the this PC version was designed to have the same functionality that would also be implemented in the MHP version (see chapter 3.2 for the description of the MHP application). In the advanced mode that was added to the PC version the color management framework for illumination-adaptive displays was developed further.

The application, known simply as Display Adaptor, starts in the "normal" mode by default. In the normal mode the user can choose the assumed viewing environment class from the five presets shown in the lower left corner of the application window: "Dark", "Home", "Office", "Cloudy", and "Sunny" (see the figure below). Selecting one of these will cause the image displayed on the right side of the window to be immediately adjusted for these conditions. This viewing condition compensation is comprised of all the steps described in chapter 2, except the flare compensation step, which is not used in the normal mode. When one of the viewing condition classes is selected the color temperature of the ambient illumination and the luminance of a perfect white in the viewing environment are set to values that are representative of these viewing situations. These ambient color temperature and luminance values, shown in the top left corner of the application window, can also be directly set by the user. The numerous viewing condition parameters of CIECAM02 for the target conditions are derived from these two values and the assumed characteristics of the display. The colorimetric characteristics of the display as well as the reference viewing condition parameters are set according to the sRGB standard by default and cannot be changed in the normal mode. (Since a viewing condition compensation is always a transformation between two viewing conditions, we always need specified source or reference viewing conditions: if we cannot model the intended color appearance of the picture elements in the source conditions then we do not know what perceived color we are trying to reproduce in the current viewing conditions.)



Luminance: 20 cd / m²

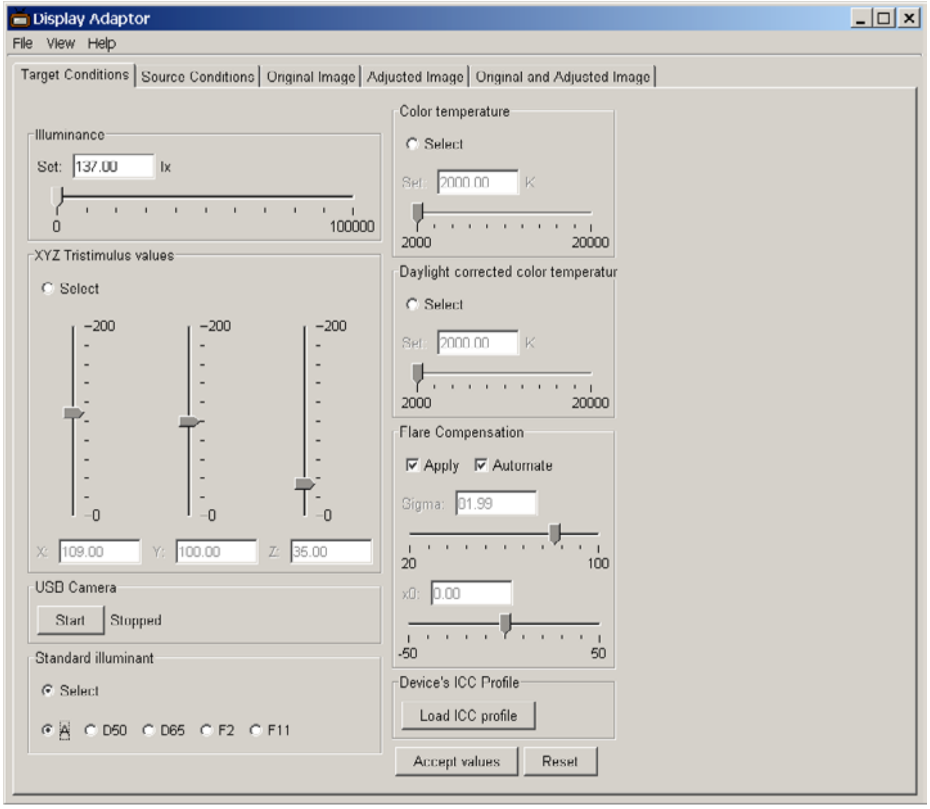
Color temperature: 2900 K

○ Dark
● Home
○ Office
○ Cloudy
○ Sunny

Figure 4. The user interface of the PC application in the normal mode

The advanced mode of the application, accessed from the View menu, offers more options for controlling the adjustment parameters. While the normal mode shows how viewing condition choices might be presented to the user in a practical interactive application, the advanced mode serves as a research tool for developing, experimenting with, and demonstrating the adjustment algorithms. A more user-friendly version of the advanced mode with further enhancements could also be useful for graphic designers working on images intended for reproduction in various media and for viewing in various viewing environments. "How will this image look as a two meters high illuminated poster on a bus-stop at night?" While it is by no means a simple task to develop an application that could in practice answer such a question with any accuracy by simulation on a computer display, such viewing-condition-aware soft-proofing is certainly an area where the general concept of color appearance modeling could be useful.

Back to the present application. The advanced mode has separate tabs for specifying source and target viewing conditions, as shown in figure 5. Here the intensity of ambient illumination is given as illuminance, which ranges from total darkness to 100 000 lux of a bright sunny day (the luminance of white used in the normal mode can be calculated simply by dividing the illuminance by pi). The chromaticity of the ambient illumination can be given directly as relative XYZ tristimulus values or by specifying a standard illuminant, or a color temperature or correlated color temperature value. Additionally, the current ambient illumination can be automatically estimated by analyzing the images read from a camera connected to a USB port. Armed with such a detector the system can operate in a fully automatic mode, continuously analyzing the ambient illumination conditions and updating the color reproduction parameters. The colorimetric characterization can be read from a user-chosen ICC profile. This profile can be different for the source and target viewing conditions. In practice the original source images would often come into the system carrying an assumed or an explicitly specified color space and associated viewing condition information, and the target display would be a specific display unit on which the application is viewed under the current ambient illumination. If the original source image has not been previously rendered into a certain display space (typically it has been), the original image values can be understood as measurements of an original scene when it was captured in certain lighting conditions. Note that while the ICC specification, with its perceptual rendering intent, provides means for storing data required for viewing condition compensation transforms between the various device spaces and the ICC profile connection space, the ICC profiles are used by the Display Adaptor only to characterize displays colorimetrically; the viewing condition transforms are calculated on the fly (but they could be encoded as ICC profiles for use with other applications that support ICC profiles).



Illuminance Set: 137.00 lx

XYZ Tristimulus values ○ Select

X: 109.00 Y: 100.00 Z: 35.00

USB Camera: Stopped

Standard illuminant ● Select

● A ○ D50 ○ D65 ○ F2 ○ F11

Color temperature ○ Select

Set: 2000.00 K

Daylight corrected color temperatur ○ Select

Set: 2000.00 K

Flare Compensation ☑ Apply ☑ Automate

Sigma: 01.99

x0: 0.00

Figure 5. The user interface of the PC application in the advanced mode

The advanced mode also provides an option to use flare compensation. When automatic flare compensation is chosen the system sets the "sigma" parameter of flare compensation automatically, increasing the steepness of the lightness re-mapping curve at mid-tones to compensate for the decreasing perceived contrast resulting from increasingly high level of ambient illumination and flare, as described above in chapter 2.2. This automatic flare compensation does not use the "x0" parameter (it is kept constant), which controls the relative amounts of compression between highlights and shadows by altering the position of the re-mapping curve. As mentioned above, this parameter should be set in an image-specific manner for optimal results. Both of the two flare compensation parameters can be set manually to experiment with their effects.

The three remaining tabs in the advanced mode are for viewing the original and adjusted images separately or side by side. In both the normal and the advanced mode the adjusted image can be viewed in full-screen mode (through the View menu). The adjustment parameter values used by the application are in fact optimized for images covering all of the display area. Visible display background can have a significant effect to the chromatic adaptation and other adaptation effects and parameterization should indeed be changed if images are viewed on a background (or otherwise the background should be adjusted the same way as the image).

The viewing condition compensation transforms are implemented as three 3-dimensional look-up tables from which the new pixel values are found by interpolation. The tables are built by taking a suitable number of samples from the input RGB space and putting them through the transform described in chapter 2.3, and recording the resulting RGB values of the display space in the tables. The tables are 3-dimensional because each of the output R, G, and B values is a function of the input R, G, and B values. The tables can be understood as cubes. The input RGB values of a pixel are used as the three coordinates in the first cube and the output R values is found by linear interpolation between the four closest node points. The output G and B values are found similarly from the other two cubes. These cubes can be visualized for the current transform in the application by choosing CLUT windows in the View menu (CLUT, color look-up table). The three axes in each plot correspond to the input RGB values and the lightness of the sphere at each point corresponds to the output value sent to the display.

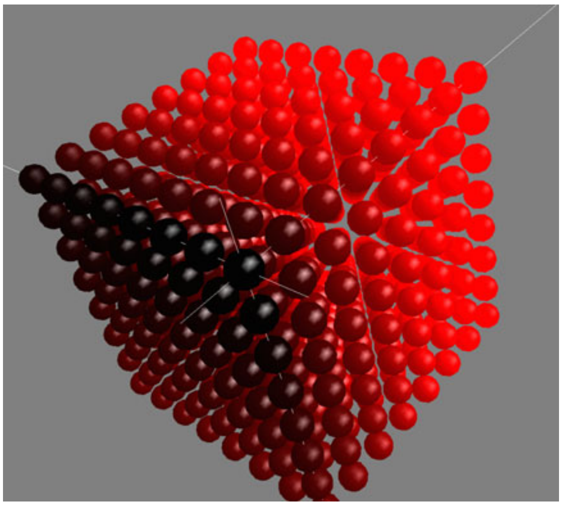


Figure 6. Color look-up table for the display control value R.

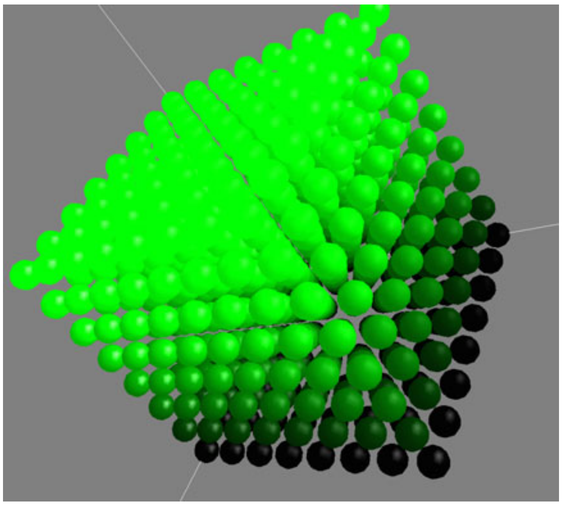


Figure 7. Color look-up table for the display control value G.

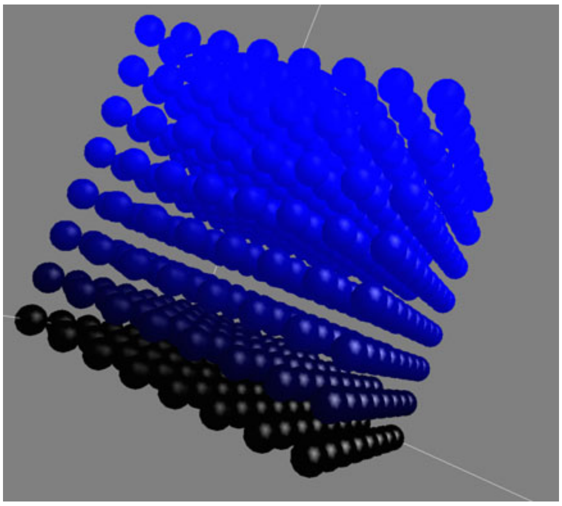


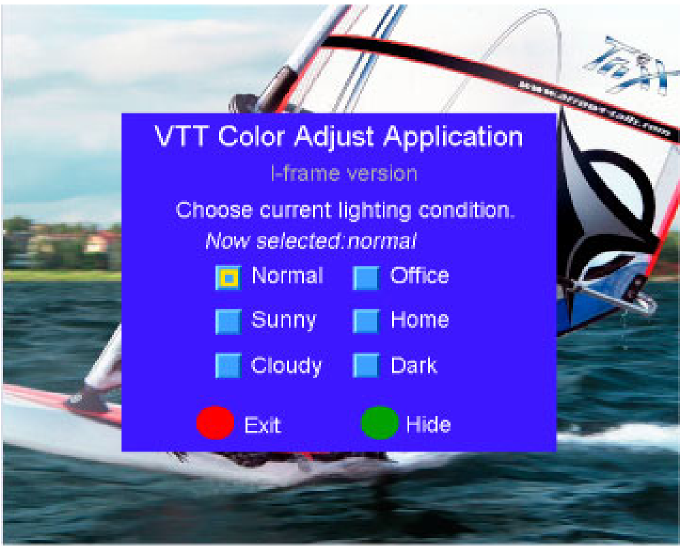
Figure 8. Color look-up table for the display control value B.

3.2 MHP Demo Application

Based on the PC demo application, an MHP demo application was also developed. Two separate applications were designed and implemented: an application that uses pre-calculated images, and an application that uses run-time viewing condition compensation. First the application using pre-compensated images is presented.

Computing power of MHP terminals is quite restricted, and MHP terminals do not necessarily support 24 bit color for regular images. Only 188 color palette is required [12]. However, in addition to PNG, JPG, and GIF images, MHP applications can use MPEG-2 I-frames as background images. By displaying images as MPEG2 I-frames we can get around the restricted color palettes that some terminals use for still images. MPEG-2 videos consist of three different kinds of frames: I-frames, P-frames, and B-frames. In I-frame (intracoded frame) all the picture information of a frame is encoded, while P- and B-frames are predictions based on other I- and P-frames. In practice, I-frame is a video clip consisting of only one frame, which can be shown, or played, on the background of the application.

We wanted to develop an application that will use true color palette on all the MHP terminals, and which can still perform the image adaptation in a reasonable time. First a simple demo application that utilizes I-frames was developed. The drawback of I-frames is that they cannot be created during the runtime. Image compensated for one particular lighting condition had to be created in advance and delivered with the application. We decided to use five different common lighting conditions (sunny, cloudy, office, home, and dark) to keep the number of I-frames reasonable. A sample image was created on a PC to match the selected lighting conditions, and then it was transcoded to a MPEG-2 I-frame. All the I-frames, a total of six, were delivered with the application to the terminal. The application itself is really simple. The user selects a lighting condition from the options that best matches the current lighting, and the application then changes the background I-frame to the corresponding one. Below are screen captures of the application. All the screen captures are taken from the PC simulation environment instead of the actual television set. In the first picture application's simple user interface is presented. In the next image an unadjusted picture as an MPEG-2 I-frame is presented with the application hidden. In the last one an image compensated for 'home' environment is presented. Note that no conclusions can be drawn from the appearance of adjusted images in this document, as discussed above, but the relative differences between them can be clearly observed.



VTT Color Adjust Application

I-frame version

Choose current lighting condition.

Now selected: normal

Figure 9. User interface of the I-frame version of the demo application

Figure 10. The original image with the application hidden.

Figure 11. Image adjusted for home lighting conditions with the application hidden

The I-frame version of the application proved to be effective and good looking. It is a fine way of demonstrating the illumination-adaptive color appearance adjustments on the television and MHP platforms. The approach is not very practical though if the number of images grows, or there is need for finer setting of the illumination parameters.

In addition to the I-frame version discussed above, we wanted to test if it is possible to adapt the image to lighting conditions during the runtime. We were aware that the MHP terminals do not provide nearly as good performance as PCs, but we did not know how poor their performance was. We decided not to try to optimize the code beforehand, but test the PC application's compensation part on the MHP platform. However, during the PC application development the MHP platform was taken into account to make the porting of the application easier. There are several classes in the current Java SDKs that are not required to be present on MHP terminals, and even if the classes are available, they conform to the older Java version (JDK 1.1). For example some methods are missing in the class java.lang.Math. Fortunately those were simple methods like converting radians to degrees and vice versa. For matrix calculations the public domain Jama[1] package became useful.

Porting of the classes targeted for PC to MHP was surprisingly easy. We had to decrease the number of samples used in some color transform tables, though. Unfortunately the performance of the tested MHP terminals (of Philips and ADB brand) proved to be poor, as was suspected. We reduced the number of CLUT samples from 8·8·8 to 4·4·4. After this modification adaptation speed became reasonable for demonstration purposes if not for real-life applications where the color compensation is just one feature of the application. The slower of the tested set-top-boxes requires a few seconds under one minute to calculate the adapted image of size 180x144. On the contrary, it takes about one second to do the same on a standard PC for an image twice as wide and high. It should be emphasized that the new image is calculated using true color palette, even when the set-top-box might present the compensated image with only 188 colors. 188 colors are not enough for this kind of application, which can clearly be seen when the set-top-box presents the adjusted image. Slight change in a color in the original image and in the compensated image is lost when the terminal replaces the color with the closest color in its palette, which quite often is the same one in both cases. An MHP terminal with full color support is required, and this could actually even make the application slightly faster, because there is no need for the terminal to convert the colors to follow the reduced palette. Below is an example image presented with the colors of the MHP standard palette.

---

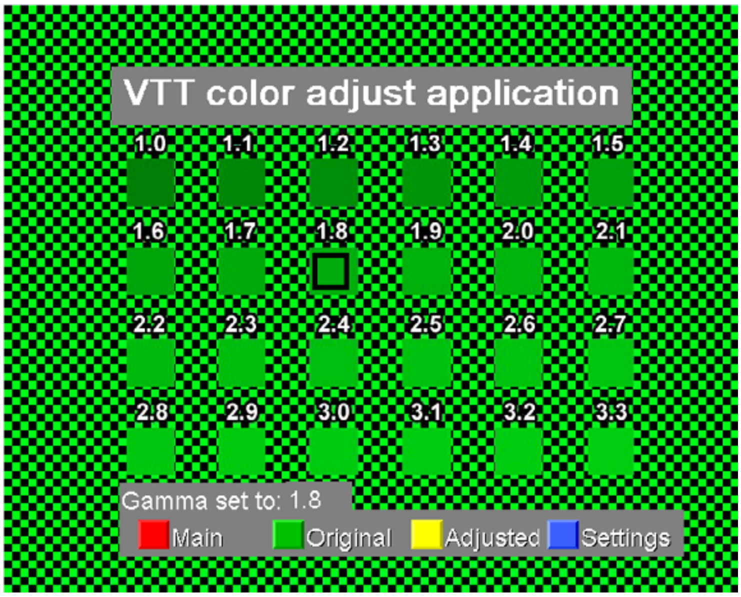
[1] http://math.nist.gov/javanumerics/jama/

Figure 12. An example image presented with the default 188 colors

There is much to optimize in the code, but we did not think it was sensible to start the optimizing. With the selected technique it is probable that even after the optimization the adaptation is too slow for real life applications. As discussed below, it is possible make the algorithms less complex by sacrificing some of their accuracy. Our goal in this project was to prove these image adjustments are possible, if not reasonable with the 1st generation set-top-boxes, in practice. This goal was achieved.

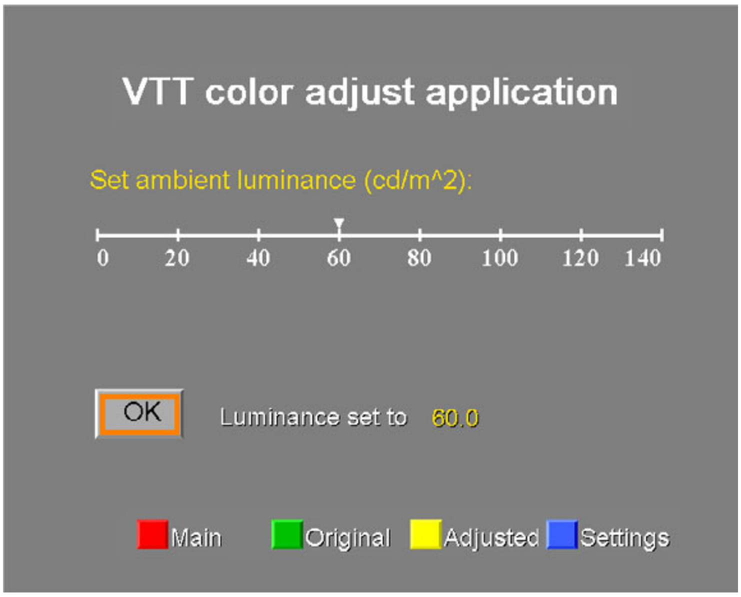
For testing purposes we developed a more advanced MHP application with dynamic compensation. With the application the user can select either a predefined lighting condition in a similar manner as in the I-frame application, or the user can set the ambient luminance and color temperature values separately. In this version the gamma values describing the tone reproduction curves of the display can be set too (otherwise the display characteristics are assumed to be as defined in the sRGB standard). That was not possible with the I-frame version, because that would have required too many images, one set of pre-calculated images for each gamma value. The application provides test images for visual evaluation of gamma value of each of the RGB channels. The gamma evaluation screen for the green channel is shown in figure 14. The dithered background pattern to which the tone of the uniform squares is compared has to be relatively large-scale since a finer pattern would be distorted when shown as an MPEG I-frame on a TV-screen.

Figure 13. More advanced application that calculates the compensated images during the runtime



Gamma set to: 1.8

Figure 14. Gamma value of the display can be set in the advanced application



VTT color adjust application

Set ambient luminance (cd/m^2):

OK

Luminance set to 60.0

🟥 Main 🟩 Original 🟨 Adjusted 🟦 Settings

Figure 15. In the advanced application the user can set the ambient luminance in candelas per square meter, as well as color temperature (with a similar slider as depicted here).

4 Discussion and Conclusions

Concerning further development and practical applications, it should be noted that handling of flare still needs much more consideration. The flare compensation algorithm, as previously noted, is a straightforward and somewhat simplified implementation of the contrast-adjustment function presented by Braun and Fairchild [10]. While that type of adjustment does indeed seem suitable for flare compensation, the parameterization and the use of the function in connection with color appearance model transforms still needs to be studied more. Currently the shaping of the lightness mapping function is done using a single parameter in an ad hoc manner based on assumed current ambient illumination in the more comprehensive PC version of the demo application (MHP version does not have any flare compensation). A similar chromatic contrast enhancement function to compensate for the decrease in chroma caused by flare is required for a more comprehensive flare compensation. As there is only a very limited amount of data publicly available on which to base flare compensation, additional subjective visual testing seems necessary to find optimal flare compensation methods.

Parameterization of the color appearance model also remains an area where there is still much room for additional investigation. In practical applications, especially fully automatic ones, the parameterization should probably be rather conservative in the sense that in uncertain situations the adjustment should err on the side of 'too little' rather than 'too much': if an adjustment algorithm over-compensates the effects of ambient illumination, it can appear to the viewer that the system is actually degrading the quality of an image that would have appeared acceptable without any adjustment. One scenario could be different parameterization schemes, which would allow the user or system designer to choose between more or less aggressive compensations. An aggressive scheme would aim to compensate as fully as possible for the assumed effects of ambient lighting, risking occasional visible errors, while a conservative scheme would set priority to the goal that no visible image degradation is ever caused by the adjustments, which could mean that most of the time the effects of illumination are only partly compensated. Especially when viewing conditions are automatically estimated it should be remembered that models like CIECAM02 have been optimized for what were considered practical imaging environments. Therefore adjustments should probably be limited in highly unusual viewing conditions (bright green ambient lighting, for example) by applying suitable limiting functions for the extreme values of model parameters. While the current application has not yet been extensively tested, it seems that the adjustments have some problems with the reproduction of highlights in the brighter ambient illumination conditions: the highlights appear to be compressed so much that a burnt out appearance (washed highlights) sometimes results. Such problems were not evident in a previous Matlab demo that used the earlier CIECAM97s model, but it is not yet known whether the current problem is caused by changes in the CIECAM02 model, improper parameterization, or simply a bug in the code (the latter two are the main suspects).

Another aspect for future consideration is image-specific adjustment. While generally good results can probably be achieved for various types of images by setting compensation parameters using only viewing condition information, the performance of adjustments can very likely be improved for specific images by adapting parameterization also to the image content. Especially in the case of contrast adjustments used in flare compensation there is strong evidence that the shape of the optimal adjustment function is image dependent and can be well estimated by analyzing image features like the distribution of tones in an image [10]. Some observations also suggest that some benefit could be gained by setting some of the CIECAM02 parameters in an image-specific way [7]. An interesting branch of color research closely related to image-specific color appearance adjustments are models that combine spatial processing to color appearance modeling more closely than the current models like CIECAM02. These types of models have been termed image appearance models [13] to differentiate them from the color appearance models with more limited modeling of spatial effects. The mechanisms of spatial processing found in these models are well suited to local contrast enhancement which can be used to subject different parts of an image to different levels of contrast enhancement in an image-specific manner. Such spatial processing, it should be noted, increases the computational complexity of the algorithms even more, and is therefore not a viable option for color transforms performed on low processing power terminals.

Indeed, when transporting this kind of application to environments where processing power is an issue, close attention must be paid to the computational complexity of the algorithms. A good example is the MHP version of the application developed in this project, which is very slow when run on a typical set-top-box, whose processing power is very limited. This is not a surprise considering that the application performs a few color space transforms and complex CIECAM02 calculations to create three 3-dimensional look-up tables on the fly. These tables are then used to calculate adjusted image data by interpolation. In a commercial application aimed at low processing power systems the code and algorithms would have to be considerably streamlined to make the adjustments fast enough. Although in CIECAM02 some parts of CIECAM97s (on which CIECAM02 is based) have been simplified, CIECAM02 is still a computationally relatively heavy model. In practical applications where viewing conditions are known only approximately it would seem proper to sacrifice prediction accuracy in favor of simplified computations. If adjustment tables were calculated beforehand for a set of viewing condition classes, on the other hand, the computational complexity of the model would not be a problem and there would be no reason to compromise the model's performance.

In summary, this project provided us a chance to build a Java demo application based on earlier work that developed methods for adapting color reproduction on self-luminous displays to prevailing ambient illumination. In addition to translating earlier Matlab algorithms to Java code, some changes were made and new features added. CIECAM97s color appearance model was replaced by CIECAM02, for instance. The comprehensive version of the application also supports the use of ICC profiles for the interpretation of input image data and display characteristics, and provides different options for specifying current ambient illumination information as well as reference viewing conditions. Taking a wider viewpoint, the application can be seen as an initial framework for a more comprehensive cross media color management system. Since the design of the application follows the principles of object-oriented programming it can be extended to handle also other types media and their associated viewing conditions in addition to self-luminous displays by adding suitable subclasses to represent different types of media and new types of color transforms. Transferring the Java application to MHP platform allowed us to gain experience about MHP application development for digital television using the obtained development environment. The simple MHP version of the color adjustment application with pre-calculated images could be run smoothly, while the heavy calculations of the complex version made the application too slow on the set-top-box for practical use, as expected.

References

[1] Janne Laine, *Results of the ADAPT Project: Framework for Adaptive Display Color Management*, Research Report, VTT Information Technology, 2001. 22 p.

[2] Janne Laine, *Results of the "Verkkosilmä" Project: Adaptive Display System Laboratory Prototype*, Research Report, VTT Information Technology, 2002. 10 p.

[3] Janne Laine, *Experiments on Adaptive Soft-Copy Color Reproduction*, Proceedings of the IS&T/SID 10th Color Imaging Conference, Scottsdale, Arizona 2002, p. 190—195.

[4] Janne Laine, *Adapting Soft-Copy Color Reproduction to Ambient Illumination*, Journal of the Society for Information Display, Number 2, Volume 11 (2003), p. 359—369.

[5] Mikko Kojo, *A method to deliver multiple media content for digital television*. MobTV-project. Research Report, 2002. 54 p. Available from http://www.vtt.fi/tte/mobtv/pub/kojo.pdf

[6] CIE, *The CIE 1997 Interim Colour Appearance Model (Simple Version), CIECAM97s*, CIE Publication 131-98.

[7] CIE TC 8-01, *A Color Appearance Model for Color Management Systems: CIECAM02*, Draft No. 12, 2003. 20 p.

[8] Nathan Moroney, Mark D. Fairchild, Robert W. G. Hunt, Changjun Li, M. Ronnier Luo, Todd Newman, *The CIECAM02 Color Appearance Model*, Proceedings of the IS&T/SID 10th Color Imaging Conference, Scottsdale, Arizona 2002, p. 23—27.

[9] Changjun Li, M. Ronnier Luo, Robert W. G. Hunt, Nathan Moroney, Mark D. Fairchild, Todd Newman, *The Performance of CIECAM02*, Proceedings of the IS&T/SID 10th Color Imaging Conference, Scottsdale, Arizona 2002, p. 28—32.

[10] Gustav J. Braun, Mark D. Fairchild, *Image Lightness Rescaling Using Sigmoidal Contrast Enhancement Functions*, Journal of Electronic Imaging, Number 9, 1999, p. 380—393.

[11] IEC, *Colour measurement and management in multimedia systems and equipment-Part 2.1: Default RGB colour space-sRGB*, IEC standard 61966-2-1, 1999.

[12] Digital Video Broadcasting (DVB), *Multimedia Home Platform (MHP) Specification 1.0.3*, June 2003. 787 p. Available from http://www.mhp.org/

[13] Mark D. Fairchild, Garrett M. Johnson, *Meet iCAM: A Next-Generation Color Appearance Model*, Proceedings of the IS&T/SID 10th Color Imaging Conference, Scottsdale, Arizona 2002, p. 33—38.