GRAPHIC DESIGN WITH COLOR
USING A KNOWLEDGE BASE

by

Louise Samson

B.A., Université de Montréal, 1969

Extended Studies Diploma, Simon Fraser University, 1983

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Computing Science

SIMON FRASER UNIVERSITY

March, 1985

APPROVAL

Name: Louise Samson

Degree: Master of Science

Title of thesis: Graphic Design with Color using a knowledge base

Examining Committee:

Chairperson: Arthur L. Liestman

Thomas K. Poiker
Senior Supervisor

Binay K. Bhattacharya

Barry Beyerstein

Robert J. Woodham
External Examiner
Department of Computer Science
University of British Columbia

4 March 1985

Date of Examination

ABSTRACT

Color has been studied fairly extensively from a bio-psychological and artistic point of view. Physicists have analysed and classified colors under a variety of schemes. Cartographers have used color on maps for a long time and have developed conventions related to the symbolism of color and its effect on the map reader. The first part of this research introduces the reader to the main aspects of these different views of colors.

Extensively used in computer graphics, color often remains a feature added last as a finishing touch, either directly by the programmer or through a restricted set of user options. Very little is done by the system itself in terms of decision-making or help to the user while, in other fields, "expert-systems" are being built which incorporate appropriate knowledge and expertise in specialized domains of application. An overview of expert-systems is given, emphasizing knowledge base representation. With the growth of computer-aided design, the incorporation of color as a field of expertise seems to be the next logical step.

The last part of this document discusses how the ordinary intuition that people have about color and the above-mentioned theories can be joined in a format suitable for representation in a knowledge base, and how appropriate heuristics can give a computer system the ability to make sound choices as to the use and harmony of colors. As cartography was chosen as the basis for the design of this system, the medium consists of maps and the rules used emphasize established cartographic conventions as well as psychological perception criteria and graphic design rules. The system, which was implemented on a microcomputer, interacts with the user to identify features and may operate in an automatic mode, where the computer makes all the choices, a shared mode, where the user selects color themes for specific features, and a manual mode, where the user dictates the colors subject to various system verifications. Contour map coloring is also available. The composition of the knowledge base and the various heuristics used are discussed and documented, and results of the experimentation are presented.

TABLE OF CONTENTS

LIST OF FIGURES

A. INTRODUCTION

Graphic Design with Color using a knowledge base is undoubtedly, as the words suggest, a truly interdisciplinary subject. "Graphics" is a field of computing science on its own, requiring geometric processing and spatial analysis from the science of mathematics. "Design" can be defined as the evolutionary process from the conception of an idea in the human brain to a finished product, often done with the use of technological tools such as a computer system. "Color" has the immediate connotation of art, but also includes sciences such as physiology, psychology, physics etc., since it is not usually an innate property of an object but rather the sensation produced in the brain of a person in response to light received by the retina of the eye. As for the "knowledge base", it is a term commonly used in the field of computing science called Artificial Intelligence, since the attempt by a computer to act in a similar way as a human expert would under similar circumstances, requires it to make use of facts and rules for manipulating these facts, which is, in essence, what a knowledge base contains. In addition, in order to channel these various concepts into a suitable area for further study, a field which encompasses art, science, intuition, precision and practical use must be chosen. Fortunately, such a natural area in which to build the prototype of an expert-system dealing with color design does exist: it is cartography, with all its spatial features being topologically arranged on a two-dimensional map designed to convey precise information on the shape of the land while leaving to the cartographer the option of emphasizing important features while ignoring certain points which would clutter the terrain; with its strict color restrictions imposed by nature itself while leaving to the designer of a thematic map all the possible color options the imagination can conceive; and finally, with its unique blend of art and science which cartographers have mastered over centuries to produce color maps which not only depict an area but can also convey an intuitive message about the area which can be very easily understood by the reader without really needing to know the techniques which produced it.

An expert system which would be capable of mastering the techniques of the cartographer and be able to replace him would be very ambitious indeed. This is not the scope of the present thesis. The goal of this study is rather to investigate the theories of color from the different points of view stated above and, from this investigation, lay the foundation of an expert-system which can help the designer with auspicious choices of color. If the system can, indeed, know some of the basic intuitions that people have about color and some of the universal rules used to manipulate them, it can be of immense help to any designer who, either wishes to have the machine do the work, either experiment with colors and get feedback from the system, or simply to do his own coloring with the help of an electronic medium. Interacting with the user, such a system could also help people who suffer from anomalies in color vision, by acting as a reliable second opinion and giving them suggestions during the design process. A prototype of this kind of system would be considered more useful if, built into the structures of data and program organization, are capabilities for future extensions and refinements in the various aspects of the system. Therefore, the scope of the system presently proposed will touch on most aspects, bringing them to a level of completion from which these extensions will emerge as a natural progression.

In order to determine this level, an overview study will be made of color in its various domains. The artistic view of color is usually the one that makes the first and most lasting impression on a human being. Good art is a matter of taste but there are some universally known artistic theories on color that artists may wish to follow or to purposely contradict to serve their own aims, as well as some good common intuition that people have about color and that are definitely worth studying. The scientific basis of color will then be surveyed from a physical (light mixing), biological (retinal perception mechanisms) and physiological (processes) point of view. These two main aspects can hardly be dealt with without mentioning the third and most important place where they meet, the psychological responses to color. Psychology has played a major role in the understanding of color and some of its findings will be described. Having examined what color is, how it is perceived and what it can do, it is then important to look at how it is done. This brings us to the messages conveyed by color through manipulation, the spatial effects of color in cartography and the design process with color to achieve these goals. We will see that many scientists and artists have expressed views which are of considerable interest in the understanding and use of communication with color.

Progressing more with the technical aspects of this work, the study will then attempt to focus the above mentioned theories into a knowledge base suitable for a computerized expert-system dealing with color design. What are expert-systems and how have they been used so far? What are the data structures, program level structures and heuristics necessary to a system dealing with cartographic information and color design? How much can be implemented to provide a sufficiently good basis for further study and extensions? These are some of the topics which will be covered as the development of the system is being discussed.

Finally, there is the prototype itself, the test of the theories. An expert-system which will be capable of coloring a two-dimensional map either by itself or with the user. A system having a knowledge base of a considerable number of colors, topological data structures, geometric processing procedures, and several basic rules of composition to manipulate colors during the design process. An interactive system which can communicate in a friendly way with the user, issue warnings, make suggestions and display choices. A system which leaves room for future extensions, such as new colors, shapes and textures, a completely natural language interface, as well as new rules taught to it by experts in cartographic design, geometry, graphics and psychology.

The final sections of this thesis will consist of the complete documentation for the system implemented together with some of the dialogues and pictorial results obtained.

It is hoped that this study in color will prove to be a fruitful meeting place for the various disciplines involved and that the computer system which will emerge from it will be a useful demonstration of their compatibility.

B. BACKGROUND

I. THEORIES OF COLOR, CARTOGRAPHY AND EXPERT-SYSTEMS

COLOR

Through the ages, color has been studied extensively from different points of view. Artists, psychologists, physicists, chemists, cartographers and experts in many disciplines have made important contributions to its understanding; more significantly, the hypotheses, theories and findings were generally directed towards a particular application or use of color. To design a computer system dealing with intelligent color choices, it is necessary to examine all these different aspects and have a basic understanding of the main points which have had an historical influence on the use of color graphic and cartographic design. Moreover, these approaches must be analysed, synthesized, grouped and harmonized in order to extract the essence which may form the base of a body of knowledge, which can then be engineered into the relatively new concept of a computer knowledge base.

Outlining only the important points dealing with color and cartography is a major task in itself and, because each person reading about such an interdisciplinary subject will have preferences as to which aspect of art, science, psychology, cartography, artificial intelligence, etc., is more important to his/her own use and may not have such a detailed interest in the other aspects, this chapter will attempt to summarize the important points in all areas which have had a direct influence on the design of the computer system discussed in the second part of this document. However, the research needed to arrive at the choices made is interesting in itself as it may help to explain how some of these choices have been arrived at and which other important points of these various disciplines may be of interest to the reader who is more concerned with a particular subject domain. Therefore, to complement this chapter, Appendices "B" through "G" expand on some of these background elements and, written as chapters in themselves, may be consulted individually by the reader with a particular interest in mind. Appropriate references and diagrams can also be found in these Appendices.

When color became a specific field of study, particular terminology applicable to this domain emerged in various areas. The three most common terms associated with color are hue, value and chroma, also called "color coordinates".(Billmeyer and Saltzman, 1981). Hue refers to the perception of shades such as blue, green, red, yellow etc., which, in physics, are measured in wavelengths. A pure color is one in which no white or black is present; adding black or white, the color appears darker or lighter; the pure color is called a saturated color or a color with a high level of chromaticity, while the value of a color refers to how close or far to black or white it appears.

Brightness, a combination of lightness and saturation, refers to the perceptual phenomenon of how bright or dull a color appears, and is usually associated with the physical concept of intensity, although a change in physical intensity does not necessarily correspond to a similar change in perceived brightness by the viewer.

Using these concepts, colors have been classified by artists and scientists under various schemes. Among the most well known are:

1) the Munsell system, an irregular solid primarily based on the saturation points of the various hues;

2) the Ostwald system, more regular and technological in nature, basing its designations on the effect produced by spinning discs composed of different percentages of white, black and the hue itself;

3) the Hicketier system, which uses the three art primaries, yellow, red and blue, to produce a three digit identification number based on the percentage of yellow (1st digit), red (2nd digit) and blue (3rd digit) component in each color;

4) the CIE color measurements, which are standards of wavelength composition;

5) the NCS or Natural Color System, based strictly on color perception; and

6) Goethe's color circle, which contains the six science primary colors (red, green, blue, yellow, magenta and cyan), and which is based on the physical phenomena of light addition and

subtraction.

These systems, as described by such authors as Libby (1974), Imhof (1982), and Agoston (1979), as well as the phenomena they make use of, whether perceptual or physical, are outlined in greater details in Appendix "B", entitled "Artistic and Scientific View of Color".

Some of these perceptual phenomena, such as simultaneous contrast and color mixture by averaging, have had a significant influence on the use of colors by both artists and scientists. Their views are not contradictory, but rather like two overlapping circles. Simultaneous contrast, for example, which can be described as a phenomenon which accentuates the contrast between adjacent colors by causing them to appear increasingly different in ways in which they are already different, may be considered by the physicist as the juxtaposition of two light beams whose mixture produces gray and by the artist as the juxtaposition of two pigments of complementary colors to produce a more striking effect. Similarly, the mixture of two or more colors by averaging, whether in time (quick light flickers) or in space (densely packed dots), when viewed at a far enough distance, results in giving the viewer of a printed page or painting (pigment) or of a television or computer screen (light), the impression of perceiving a new uniform color. Under the umbrella of human perception, colors then assume a whole new role, the role of conveying a message through the sophisticated mechanisms of the human eye.

Appendix "C", entitled "Biological, Physiological and Psychological Aspects of Color", describes the human visual system, how light which touches the retina of the eye gets transformed within the brain into electrical impulses which allow the perception of a whole range of colors. Rods and cones make up the photosensitive layer of the retina. (Goldstein, 1980). Generalizing their functions, we can basically say that the long wavelength cone of the eye perceives reddish hues, the medium wavelength cone, greenish hues, and the short wavelength cone, bluish hues. Rods, which are much more numerous and sensitive than cones perceive a world without colors, which means that at night we see mostly with our rods and during the day, mostly with our cones. Together, the rods and cones produce the red-green-blue of color vision, necessary for the perception of all other colors. Persons who have all three cone pigments are referred to as trichromats; those who have only two are called dichromats. Although the latter are often commonly referred to as color-blind, they are not really blind to all colors but only to those where the missing visual pigment would make an appreciable difference in the visual summation process. Monochromats, who only have rods and sometimes one type of cone are referred to as truly color-blind.

The main psychological theories on color vision, the trichromatic theory and the opponent-process theory, (Goldstein, 1980), are also discussed in Appendix "C", as well as the interesting phenomena of dark,light and selective adaptation and the influence of surrounding areas. These are particularly relevant to the design of the proposed system. Cartography is very concerned with topological relationships between neighboring areas and, therefore, the influence of the surroundings is a major element in the choice of colors on a map. Persons suffering from anomalies in color vision will use other "clues" to differentiate between colors making use of the natural adaptability of the visual system; persons who have complete color vision will use their adaptation mechanisms to see the same color under different illuminations. Some media, on the other hand, do not adapt themselves, which may account for the possible differences between color photography, color reproduction and the colored image on a computer screen seen under different illuminations. A computer system which could appropriately match and contrast colors could help people suffering from color perception deficiencies not only in producing good color designs but also in learning to appear to perceive colors as the average person sees them.

Color is one of the most obvious visual qualities experienced by human beings in their daily experiences; as one of several visual variables, the impact of its message cannot be ignored in the design process. Appendix "D" discusses how Jacques Bertin (Bertin, 1967), analyses and organizes the six visual variables (size, value, texture, color, orientation and shape) and their instantaneous perceptual spheres of influence (associative, selective, ordered and quantitative). Color is only seen as associative, which means that all signs of the same color and visibility can be perceived instantaneously, and selective, meaning that all colored signs can be grouped into families. To convey order, color must be combined with value, such as light blue to dark blue, and it cannot convey an intrinsic sense of measurable quantity unless it is linked with the size of the area colored. These principles are important in cartography, which makes use of gradual color values for altitudinal tinting, of groups of similarly colored elements to represent specific features of the terrain and of the visibility of a color to determine its appropriateness for small and large areas, important or irrelevant features. The advantages of using color are numerous: it helps keep the attention of the viewer, increases the number of interested readers, helps memorization and increases the scope of the message. Keeping in mind the above-mentioned principles of color diffusion or averaging mixture, contrasting effects and aesthetically pleasing color matches, color can then be a valuable asset to convey the appropriate message to any map reader.

CARTOGRAPHY

One of the leading researchers who had had a profound influence in cartography, especially when dealing with color, is Eduard Imhof (Imhof, 1982). Some of his contributions include hypsometric or altitudinal tinting of maps. This consists of following a color gradation scheme in which different altitudes or depths are colored lighter or darker depending on whether the progression is increasing or decreasing. Used primarily for contour maps, his proposed scale comes from a long line of attempts and experiments by such eminent cartographers as Karl Peucker. Appendix "E" outlines thirteen such schemes as well as other important graphic design rules used in cartography.

The goal of these rules is mainly to achieve good conceptual interplay as well as good graphic interplay to simplify map reading. The unbroken sequence of colors from the color circle is perceived as harmonious. Compounds of two or three complementary colors also have an harmonious effect. Subdued colors, lightened by white or darkened by black, are often more pleasing than pure colors. In addition, if the subtractive mixture of two colors produces white and their additive mixture black, harmony is also achieved, while bright colors placed in close proximity to each other may have the opposite effect. Pure and bright colors do better in small areas while large areas and background benefit from duller colors. Natural gradation of colors emerging from the same hue, for example light blue to dark blue through several other shades of blue, produces very pleasing and calming effects, imitating what in nature can be produced by distance viewing or different light intensities. Care should be taken when placing pure, bright and strong colors or light, bright colors mixed with white next to each other; they are best used intermingled sparingly throughout the picture. When dealing with a thematic map, the colors of the main theme also produce a better effect when scattered in a duller background color which allows brightly colored small areas to stand out more vividly. The overall composition should maintain a uniform color mood through a continual softening of area tones although this should not be considered opposed to a contrary requirement, that of contrasting effects.

The above points out that artistic freedom and intuition are still a necessary part of the cartographer's trade, just as much as scientific and logical rules of design. Going hand in hand with the graphic laws of composition, is the ever present symbolism of color. Respecting it ensures that the map will be a beautiful as well as an informative piece of graphic work. Additional details on cartographic principles dealing with color graphic composition can also be found in Appendix "E".

However, before coloring a map, one must produce it in accordance with other general cartographic principles. Mapping translates the real multidimensional world into a two-dimensional sheet of paper or computer screen. Its main elements are points, lines and areas. Organizing these elements to isolate some important spatial properties, to select, to generalize and to model the real world to facilitate understanding is important for the map to serve as a useful communication device. Measurements and spatial or topological relationships have always been considered significant elements in map design; occasionally, however, the designer may wish to sacrifice some theoretical elegance to achieve a special visual effect. In this context, color can play an important role.

During the First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems (Ed. Dutton, 1978), several ideas were presented to show the importance that topology has assumed in the field of cartographic representation. Special data structures used in systems such as DIME and POLYVRT, as well as proposed structures for three-dimensional surfaces, (Peucker, 1972, Peucker and Chrisman, 1975), are discussed together with other "Cartographic Principles" in Appendix "F".

Armed with a body of knowledge on color harmony and perception, graphic design and cartographic rules, it remains to organize or engineer them in a way that a computer can use to make the appropriate decision when faced with a map to color.

EXPERT-SYSTEMS

Computer systems capable of deduction through consultation of a group of rules and a body of knowledge called a knowledge base are generally referred to as expert-systems. Designed to simulate the behaviour of a human expert in a similar situation, expert-systems nonetheless are programs which are subject to certain limitations; this has given rise to several controversial opinions about their efficiency, structure and usefulness among adepts of artificial intelligence. How good an expert-system is can be measured in various ways, one of the best still being how well it responds to the particular user's needs.

From their early beginnings, Terry Winograd's "Blocks World" and Joseph Weizenbaum's "Eliza" programs, expert-systems have incorporated several common characteristics. Among them are heuristics for reasoning with judgemental knowledge as well as with formal knowledge of established theories; a user-friendly interface which hides its complex manipulations under an appearance of simplicity; a capacity to accept additional knowledge and other facts and rules without extensive modifications to its basic structure; a set of deduction mechanisms contained in what is commonly referred to as an "inference engine"; and finally, and maybe the most important, a well engineered knowledge base which contains the required facts and rules in an accessible and understandable representation scheme.

Referring to ideas presented by Artificial Intelligence experts (Buchanan and Duda, 1983, Special Issue on Knowledge Representation of the IEEE Computer Society magazine, 1983), Appendix "G", entitled "Expert-Systems", reviews the above-mentioned characteristics of expert-systems and presents some of the major knowledge representation schemes used in actual systems, such as semantic networks, frames and scripts, logic-based representation using Prolog-type computer languages, and the production-rules if-then-else representation scheme, which is used in the present Graphic Design with Color system to depict such rules as mentioned in the second paragraph under the heading "Cartography".

In addition to giving examples of these knowledge representation schemes, Appendix "G" reviews some of the systems or sub-systems which make use of some artificial intelligence to specifically address cartographic problems. Intelligent cartographic systems attempt to bridge the gap which exists between the traditional and the computer cartographer. Some of these include: a special language for dealing with maps (Youngmann, 1978); a sub-system named CARTE which interactively displays polygonal data (Wood, 1978); an image-based geocoding system for correlating satellite and topological data bases (Bryant, 1978); and the MAPSEE system which interprets freehand geographical sketch maps (Havens and Mackworth, 1983). Finally, it appears that with the progress of technological frameworks, experts are now more able to concentrate on the "knowledge engineering" aspects, helping to make future expert-systems useful and powerful computational tools.

The above is only a brief summary of the varied interdisciplinary approaches which had to be considered before attempting to incorporate color and cartograhic theories into the actual implementation of a computer system capable of making intelligent color choices. Many of these theories have strong supporters and staunch critics; within the confines of their specialized domain of application, they may even be the source of strong controversial disputes on particular hypotheses or conclusions. However, viewed from within an interdisciplinary context aimed towards a succinct integration of centuries-old knowledge, they present certain common characteristics that are generally accepted by the world at large in their applications. The goal of this study was not to debate the validity of or to take sides on any particular issue, but rather to present and integrate the best relevant parts of these theories and hypotheses into a compatible entity specifically oriented towards intelligent map coloring by a computer system. To this end, the second part of this document will explain in greater details which of these theories and rules have been retained and implemented, the data structures and algorithms used for their implementation, as well as some of the results obtained through experimentation.

C. A SPECIFIC COMPUTER APPLICATION

I. A SYSTEM FOR GRAPHIC DESIGN WITH COLOR

DOMAIN

The choice of the words "graphic design" or "cartographic design" in the title of this chapter has been a difficult one. It is true that the system described below is mostly of a cartographic nature in the sense that some of its specific knowledge and mostly the data with which it has been tested all relate to maps. As mentioned before, any working system must somehow be channeled in an area of application and, for reasons also described previously, cartography was chosen as the proving ground for this particular system. However, the title "graphic design" was chosen because what is at the heart of the system, its data structures and knowledge base, can be applied to color any type of graphics. Details in filling in these structures may change and certain specialized procedures may have to be replaced by other types of specialized routines. Nevertheless, the role played by visual variables and graphic rules of composition are universal and it is with this thought that the following description of the system should be considered.

One of the fundamental notions that should have emerged from the discussion on expert-systems, and that must be borne in mind, is that the system itself is not intelligent. It is made to simulate intelligence through its knowledge base and the rules which manipulate it. The knowledge base itself is input by the designer of the system in structures organized to access it in a specific way through procedural rules. The search is not different, procedurally, to that of any other program but the criteria which govern it are. The data is also contained in familiar, ordinary structures, but its organization within them is designed for easier and varied forms of access. So, if the designer modifies only the interior content of the structures, the system can be made to look as it has gone completely mad in its choice of colors or as it has suddenly acquired a whole new field of expertise dealing with, for example, colors appropriate for a play set in the seventeenth century under the reign of Baroque art or for the design of futuristic modes of transportation or habitation. The manipulation of these structures is achieved through interactive processes and procedures which blend colors and harmonize shades and hues within the context of geography and in accordance with the rules of graphic design.

COLORS

CREATION

The computer on which this system was implemented has eight built-in colors: black, blue, red, magenta, green, cyan, yellow and white. To create any other color, one immediately thinks of combining two or more colors through projection of their various beams of light on a single spot. Varying the intensity of the beams would produce different effects or colors. However, a computer with the capacity of varying intensities like analog waves is very expensive, with the result that most microcomputers have only discrete pixel lighting which means the pixel is either on or off. Furthermore, like in television sets, only three colors are actually present in the form of bit planes covering the screen within the machine, that is red, blue and green (RGB monitor). Therefore, all the other built-in colors have already made use of this capacity of combining two full beams of light on a single point: magenta results from the bit located at a certain point on the blue plane and the same bit on the red plane being on simultaneously. Cyan combines blue and green, yellow, red and green, while white requires the three planes to be simultaneously on and black, simultaneously off. To create more colors, one has therefore to resort to another process which is based on the phenomenon called color diffusion discussed previously. Briefly, this property says that if two contiguous areas of different colors are small enough, or are viewed at a far distance, the two colors will mix and average out spatially in the viewer's eyes. Black dots on a white page, seen from a far enough distance, will appear like a uniform grey square. Applying this to pixels on a computer screen, we can then create many combinations of alternating color dots and, since a pixel is almost too small for the eye to detect unless one is really looking for it, the resulting impression of a yellow and a red pixel alternating continuously will be orangy. This is the "dithering" process. With eight colors, dithering two pixels only produces thirty-six colors. This is because alternating a red pixel and a yellow pixel or alternating a yellow pixel and a red pixel is exactly the same. The result is therefore not eight to the power of two, but rather (7+6+5+4+3+2+1) or n(n-1)/2 combinations plus the 8 pure colors, giving 36 colors. Dithering three pixels would produce considerably more colors but we are again prey to the same phenomenon. A green, a blue and a red is no different than a green, a red and a blue on a large screen. The best way was then to emphasize certain colors by ensuring an inequality. The method chosen was two pixels of one color followed by one pixel of another color. Two reds and a yellow produce a darker orangy yellow than two yellows and a red. Sometimes even a slightly different hue is perceived. Two reds and a green appear yellowish while two greens and a red is definitely green. Proceeding this way for all colors, we get sixty-four combinations, most of which are different enough for the eye to differentiate, and which fall into seven broad categories of green, blue, red, yellow, magenta, cyan and neutral. There are certainly other dithering methods in widespread use. Some packaged graphic systems boast ninety-two colors, which may be very nice to have, but for a system such as this, the capacity of the human eye to differentiate between every combination was considered important. With sixty-four colors, this is sometimes put to the test. With ninety-two or more, certain combinations would certainly be so worthless as not to be worth considering. It would also require more storage and its usefulness in other applications would have to be thouroughly assessed before implementation.

These sixty-four colors, although easy to create and produce to color a single polygon, did however create some problems during the implementation of the coloring process. More details will be given later during the discussion of the actual coloring process algorithm, being sufficient to say for now that black and white posed special problems because of their dual roles in representing respectively the uncolored screen and the polygon borders. While a method has been found to use white as a color as well as a border in most cases of map coloring, it was however necessary to remove black altogether as a color, mostly because it was impossible to differentiate in all cases between an empty or uncolored pixel from one colored black. This finally resulted in a possible forty-nine combinations of colors.

On expensive computer systems which have almost infinite possibilities of producing colors directly at the level of each pixel, as the I-square-S, the above dithering process would not, of course, be necessary.

	BLUE	RED	MAGENTA	GREEN	CYAN	YELLOW	WHITE
BLUE	1 5 BL 111 C	2 2 MA 112	3 5 MA 113	4 4 BL 114 C	5 5 BL 115 C	6 1 NE 116 D	7 3 BL 117 C
RED	8 2 MA 221	9 5 RE 222	10 5 MA 223	11 2 YE 224 C	12 1 NE 225	13 5 YE 226 C	14 3 RE 227 D
MAGENTA	15 5 NE 331	16 5 MA 332	17 5 MA 333	18 1 NE 334	19 4 MA 335	20 2 RE 336	21 3 MA 337 D
GREEN	22 4 NE 441	23 2 GR 442	24 1 NE 443 D	25 5 GR 444	26 5 GR 445 C	27 5 GR 446	28 3 GR 447 D
CYAN	29 5 BL 551 C	30 1 NE 552 D	31 4 NE 553	32 5 GR 554 D	33 5 CY 555 C	34 4 CY 556 D	35 3 CY 557 D
YELLOW	36 1 NE 661 D	37 5 YE 662 C	38 2 NE 663	39 5 YE 664 C	40 4 YE 665 D	41 5 YE 666	42 3 YE 667 D
WHITE	43 3 BL 771 C/D	44 3 NE 772 D	45 3 NE 773 D	46 3 NE 774 D	47 3 NE 775 D	48 3 YE 776 D	49 5 NE 777 C

Figure 2.1 Color Chart.

CLASSIFICATION AND ANALYSIS

For usage in the knowledge base, the forty-nine dithered colors produced on the implementation computer were classified using different schemes. Out of each color result some five or six pieces of information which the different structures exploit interchangeably in the search of the best coloring scheme. The previous table illustrates the information obtained.

First, each color is given a three digit identification number by the system as well as a two digit number. The two digit number is only the order in which the colors appear on the chart (from 1 to 49), and it serves as the subscript of the color structure array. It appears on the top left hand corner of each color on the chart. The three digit identification number, referred to by the user when choosing a color, appears at the bottom left hand corner. This number also identifies how the color was dithered. Each digit is the color code of the pixel according to the particular computer and language used. Therefore, on the Zenith-100 with Pascal, black is "0", followed by blue, red, magenta, green, cyan, yellow and white (7). A name could have been used, but since the purpose was not to make a catalog of decorator paint and names can be ambiguous, a number was found more efficient and easy to use. On computers and with languages using other absolute color numbers for the pixels, the order would be different. With this classification scheme, color 114 refers to the color composed of two blue pixels (1) followed by one green pixel (4). This becomes particularly useful in order to decompose the color number during procedures without having to store the combinations separately. It must be remembered that the two digit number is not stored, but only used as a subscript. The third item of information, which appears on the top right hand corner, is a digit from 1 to 5. These are the match/contrast values of the pure color identified on the top horizontal line of the chart with the pure color identified by the left vertical line of the chart. The match and contrast value is really the brain which guides the system in deciding on the harmony of a color with other colors. The method used to determine it was a combination of the scientific and artistic principles previously described. The base was Goethe's color chart and the premise that adjacent colors are a good match and the opposite color a good contrast. For example, blue is adjacent to cyan and magenta and opposite to yellow; so the first two were given the weight 5 and yellow, 1. White is a neutral color, so it was given the weight 3 across the board. At the pixel level, a color with itself is considered a good match and this combination is worth five. One can argue that a color with itself is the perfect match when choosing a wardrobe but a terrible match when we wish to delineate areas on a map. The key here are the words "pixel level". Since a dithered color is composed of three pixels, the match/contrast function described later on will deal differently with totally identical dithered colors. This still leaves us with two colors of the circle unaccounted for and this is where some problems surface. The six colors mentioned in art had perfect symmetry, i.e. orange made of yellow and red, green made of yellow and blue and purple made of blue and red. In Goehte's circle, however, and this is how science orders colors, this symmetry does not exist visually as well. Red is opposite cyan and adjacent to yellow and magenta. Green and blue are left. Blue contrasts well with red and so does green, which is a cool color made with paint by combining yellow and blue. On the other hand, in considering blue, green and red are left. Green is a cool color and red a warm color. In this case, red was given weight 2 nearer to contrast and green 4 nearer to match. A table was made up using these principles and a function written to compare each pixel of one color with each pixel of another and add the weights. The results are values between 9 (all ones) and 45 (all fives), with the exception of the comparison of two identical colors with all three pixels in the same order. This combination is worth zero. A good match has a high value, a good contrast a low value (except zero).

This match/contrast function is the virtual brain of the knowledge base. It gets called every time two colors must be compared to compute their match/contrast value. Some of its main advantages are the savings in storage space, the reduced need for user input and its consistency in applying graphic design rules. To store the match/contrast value of each color with each other color would require in the order of N to the power of two storage space, where N is the number of colors. With 49 colors this would mean storing some 2401 integers or some 4K bytes. This could be reduced approximately by half by storing every combination only once, but at the expense of additional reference pointers (storage) or complex search procedures (time/complexity). The method used only requires storing N integers in the knowledge base and a simple function which takes advantage of the composition of the identification number and which can rapidly compute the value as needed. This also entails much less effort on the part of the user. If every match/contrast value was stored, it would also have to be determined and typed in as input. Easy to do by looking at a few colors, it becomes a nightmare to try to evaluate correctly all nuances of every color of a reasonable size knowledge base with every other color. Less input also means less chance of typing errors and a time savings at the input stage. In addition, any change made to the weight of two pure colors is automatically propagated to dithered colors when the function is invoked again, without the need for any further user intervention. As such, it can be considered as a virtual storage structure as well as a procedure. But what about "correctness"? As mentioned under the sub-heading "Creation", the basis for the assignment of the match/contrast values at the pixel level is Goethe's color circle, a universally accepted standard, supplemented, in ambiguous cases, by the "warm" and "cool" color classifications, also well proven in producing color harmony. Since the identification number of each color is based on pixel level composition and the same function is applied to any comparison, the results are bound to be consistent and in accordance with graphic composition rules. The algorithm of the match/contrast function (MC) is described in more details in Appendix "A" and results are discussed more thoroughly later on in this chapter. In anticipation, it can be said that, all other factors considered, this function performed very well and proved that the storage of all match/contrast values was unnecessary.

Such a system could also be used with computers having an infinite number of colors. One could make a color chart for saturated colors and, as colors go away from that point, use a function to compute an intermediate interpolated value. Values would then most likely not be integers but would include a fractional part such as 4.25. Care would also have to be exercised in setting a difference based on perception in order to consider two colors as different; if not, two colors could be computationally different but perceptually the same, defeating the purpose of the system.

Another piece of information which is essential for the knowledge base to have, is a "feeling" for the appearance of that color with regards to hue. Here, scientific principles and functions do not apply very well. One reason is that color intensities produced by the computer vary according to the hue. Some colors "show up" more than others. If we identify the six main hues of the color chart, plus a neutral hue, we find that perceived colors do not really follow straight lines as should be the case if the objective criterion of the two first pixels determining the color was used. Considering blue as an example, as observed during the implementation, two blue pixels and one blue or one green or one cyan or even one white will give a blue tint. With a red, the predominant impression is a violet blue, closer to the conception one has of magenta. With a yellow, the shade is rather grayish or undetermined. The central line of each color on the chart identifies within which hue this color fits best. It is subjective, based on perception as well as using it in coloring examples. A single square of this color may appear a certain shade, i.e. a kind of blue, but when used to color water, for example, it does not look right and looks more violet. One observation that can be made is that neutral shades seem to predominate. These are colors which do not fit well anywhere within one of the six hues mentioned. If one wishes to use yellow and gets color 661 (two yellows and a blue), it could be disappointing. These "neutral" colors are good fill-in colors, but do not represent the hue well. As this is a subjective criterion, it can be changed easily in the knowledge base by modifying some of the data read by the program as input, which will be discussed later. Since the match/contrast values for the colors are determined objectively by a function, it is interesting to note how these two criteria fare together. The important use of the hue is mainly for the system to determine if water is blue, vegetation is green etc.. and to respect color themes. A variable called "blue" is set to point to the pure blue in the color array which in turn, contains in the variable "next", the subscript of the next shade of blue etc.. The same is valid for other hues. Therefore, when the system is asked to check if a chosen color falls in a pre-determined hue, or to pick a certain hue, it only has to follow the "pointers" for that hue, proceeding along a color as it were, avoiding a lenghty search through all the colors.

Lastly, some colors on the color chart have none, one or two letters in the bottom right hand corner. This again has to do with classifying the colors for input into the knowledge base. In the rules of graphic design contained in the previous chapter and in the appendices, it was mentioned that large areas should have a subdued or dull color, and so should the background, that small areas should always have a more vivid tint etc.. To apply these rules, some colors were also classified as "dull"; these are the ones with the letter "D". Again this classification is a perceptual one. At least one shade of each hue is "dull", usually the lightest one. This is necessary for proper color selection in some parts of the program while respecting hue choices. Neutral and dull are different criteria. Neutral relates to how well a color fits within a particular hue, while dull relates to the subdued appearance of any color. The letter "C", on the other hand, does not have anything to do with the perception or the composition of the color itself, but rather relates to the use of that color for a special option called "contour map coloring" based on

Imhof's choices of color scales. This will be covered in more details as coloring options and processes are reviewed.

ACQUIRING AND STORING THE KNOWLEDGE: INPUT AND MAIN DATA STRUCTURES

COLORS

The previous analysis and classification of colors must somehow be made known to the computer. Is is an essential part of the knowledge base on which this system is dependent. There are four data structures dealing with color. The main one, the "color" structure, is an array of forty-nine records which contains the three digit identification number and a "pointer" to the next color in a particular hue. The number is computed by a procedure and does not need to be read in. Since the classification in a particular hue is not based on a mathematical function, it forms part of the data. For each of the forty-nine colors the number (1 to 49) of the color which follows it in the hue is read in. The order is irrelevant. These subscripts then serve as pointers and a default value is used for the last color in the hue. The first color in the hue is read into specific variables identifying the hue by name. These are usually the pure colors since they are the first to be looked at and stand more chance to be chosen but again, it is not necessary to do so. One reason it is desirable, is that saturated or pure colors offer the best perceptual delineation. The other two structures are an array containing the dull color and one containing the contour color subscripts (1-49). Finally, the first letter of each hue is read into a character array. This is used for cross-referencing and interfacing. As the previously mentioned color chart is based on relationships between saturated colors available to each pixel, it is "constant" regardless of the dithering pattern(s) used; therefore, with storing only some 2N+C elements, (in this case: color chart (49 integers), id. and next in hue (130 integers), cross-reference letters (7 char.)), the color part of the knowledge base is complete, within the confines of the present system.

POLYGON AND FEATURE CODE

After having acquired a knowledge of colors, it is then necessary to find out about the map to which they will be applied.

The second main structure concerns the input data itself and the pre-processing of it into a form necessary for the system to intelligently manipulate it. As entered, the data consists of a series of X and Y coordinates given in clockwise or anticlockwise order and forming polygon shapes. The end of a polygon is detected when the last pair of coordinates is equal to zero. Each polygon is an entity in the coloring process. Its borders separate one color from another. The system will not itself color two adjacent polygons the same color but it is possible through the manual option or an allocation of similar feature codes. A map is composed of a series of polygons. Each polygon structure contains the following information: an identification number, a feature code, the size, the minimum and maximum X and Y coordinates, an anchor point in the polygon, the color, the immediate neighbors and the larger neighborhood. If the function of the polygon identification number is self-explanatory, and depends only on the order the polygons are read in, the others may need further details.

The feature code is a code given by the user and by the system to identify certain geographic or thematic entities. The system "knows" of two features usually present in a map and which particularly affect the choice of colors: water and vegetation. It automatically prompts the user for the identification of these areas and assigns polygons identified in this manner the appropriate feature code. However, the user may wish to add to that. He can define other important features by name, identify polygons belonging to that feature and the system will in turn assign a number to it and keep a cross-reference of feature codes names and numbers. Feature codes can be used for many useful purposes such as: to group polygons, i.e. all industrial zones; to "single out" a certain special polygon, i.e. highlight the state of Maine, by reserving a special color for it; or to group polygons which form a single geographical unit but are separated by other features, i.e. Vancouver Island and the mainland under British Columbia. The feature code structure is used for this cross-referencing between the name preferred by the user to identify a feature and the number used by the computer in internal manipulations. It contains a feature code identification number, a name, an optional theme hue and a specific color. As any necessary information for this structure is either computed or stated in the program or acquired interactively, no previous input data is required. Any polygon not identified as anything special is given a default feature code.

Size is also self-explanatory and its main reason for being in the structure is its effect on color harmony. Many of Imhof's rules and Bertin's comments on the effects of visual variables relate to size. (Imhof, 1982),(Bertin, 1967). Large areas should not be too vivid unless on purpose, small areas cannot be too dull or they get lost in the picture. The size is computed using a formula outlined in Appendix "A" under SETUPPOL. The color of a polygon is its actual color on the screen; this part of the structure is filled in as polygon colors are assigned and allows comparisons to take place as to what has already been done in order to determine future colors. On the other hand, if a color is written in without the drawing process being activated, backtracking can take place more easily since the actual area is still uncolored. Whether the color is to be considered final or not is built into the procedure which activates the actual coloring process.

Next come the minimum and maximum X and Y coordinates. These are very important since they are the only point information kept on a polygon after its borders have been drawn. They are used both for determining the immediate and larger neighborhood of the polygon as well as in several other procedures. These are in fact four pairs of coordinates since the minimum X has its corresponding Y and so on. Their use in the determination of the neighborhood resulted from the following reasoning. In the POLYVRT system, (POLYVRT User's Manual, 1974), we have seen that the immediate neighbors were the polygons adjacent to the polygon in question. The information as to what was on the other side of the border required the user to say so at the input stage. Part of the information required about a segment was the left and right polygons. Although this information is extremely valuable, it seemed that an "intelligent" system should not have to require it as input but should be able to deduce it itself. This seems easier than it is. When a person looks at a map, adjacency is very obvious because one sees the whole picture. Computers, on the other hand, only have point information. A line is nothing else but a series of points. If the points are all positioned vertically at column i, for example, it is easy to conclude that column i-1 is left and i+1 is right. But what if the line is a diagonal? How does one determine what i+1 is? Point in polygon algorithms are complex, require keeping information on all vertices and, in image processing, a specific and unique border criterion must exist. A better question may then be: why are we doing this? What is the purpose of finding the adjacent polygons to include them in the immediate neighborhood? In this case, the answer is obviously the influence that a choice of color for a neighbor of a polygon has over the possible choice of color for the polygon itself. If a neighbor is blue, and the user wants a "match", do not pick yellow. But we also know, because of the diffusion of color, and of human perception, that a few pixels, important geometrically in the X and Y coordinate scheme, are really not important to the human eye. Or rather, their importance comes from the fact that they tend to be ignored. Who has not seen some slight overlapping in the printing process of colors? When are borders that sharp that they approximate a computer's view of a border? Anyone studying computer vision, an area of artificial intelligence which deals with vision of images by the computer, will have noticed how sharp borders to a human eye become fuzzy when translated into a computer array. Moreover, even on a large map, the eye tends to focus on an area and notice the environment in a sort of horizontal and vertical "corridor" fashion. That is, when looking at a polygon, if something comes into our horizontal and vertical line of vision, like surrounding the polygon on two or more sides, it will have as much an influence on our perception than something that just touches it for a small length. This perceptual phenomenon can be illustrated as follows:

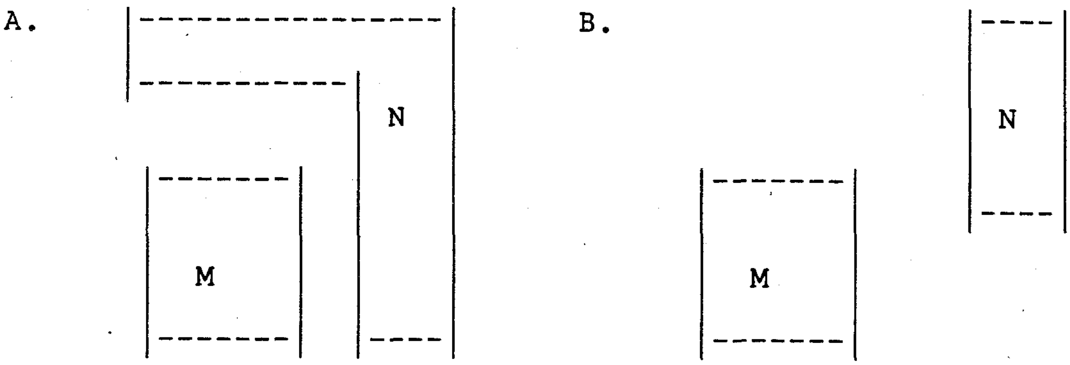


Figure 2.2 Example of a perceptual immediate neighborhood.

In looking at polygon M in "A", it is difficult not to immediately notice polygon N because it is present in both our horizontal and vertical perception "corridors", or line of sight. However, looking at polygon M in "B", it is easy to see that polygon N has a smaller influence on our perception since it appears only in the "corner of our eye", or lateral vision. If these polygons were colored, the difference would only be accentuated. An intelligent system must see that. Proceeding in a purely geometric way would give accurate results as to adjacency, but would be too long and complex and would ignore this important perception property that humans have when dealing with a color design, which, in turn, means that it would have to be built-in somewhere else as a heuristic in the system. It appeared to be much better to redefine the meaning of immediate neighborhood. This new definition, as applicable to this particular system, when expressed geometically, is that two polygons are considered "immediate" neighbors if there exists an overlap in their X and in their Y coordinates. Let us take, for example, a polygon with a minimum X coordinate (10,25), a maximum X coordinate (20,32), a minimum Y coordinate (20,24), and a maximum Y coordinate (15,35). That would mean that any polygon whose minimum X coordinate is less than 20 OR whose maximum X coordinate is greater than 10 AND whose minimum Y coordinate is less than 35 OR whose maximum Y coordinate is greater than 24, is a neighbor of the polygon in question. Explained another way, if we were to draw an horizontal corridor taking as borders the minimum and maximum Y coordinates (rows) and a vertical corridor taking as borders the minimum and maximum X coordinates (columns), anything which touches or is in the intersection is a neighbor. In fact, all the actual geometrically adjacent polygons are picked up this way. Even with two squares touching horizontally, the minimum X of one will be equal to the maximum X of the other, and since all rows or Y's are the same, they will qualify as neighbors. There is one geometric flaw in this algorithm which turns out to be the bonus in the color perception case. A polygon which would not touch the above-mentioned polygon but which would surround it like an inverted "L" shape, with, for example, a minimum X coordinate of (15,40), a maximum X coordinate of (25,40), a minimum Y coordinate of (25,30), and a maximum Y coordinate of (15,40), would also qualify as a neighbor. There is nothing to say that these four coordinates must be joined by straight lines. The above illustration depicts a similar situation. Such a case is precisely what should influence the system and make it appear to behave intelligently. Any human cartographer would see, if only subconsciously, that an area like that is either very large, elongated or somehow enters in the field of vision when looking at the polygon and should be considered as an important influence. The possible objection that a map may have dimensions so vast that if that surrounding shape is far enough it will not matter, is rapidly obliterated when we consider the medium of the system, a normal size computer screen. Any polygon overlapping in two dimensions will have an influence comparable to that of any adjacent polygon. Equally interesting is the case of a river or road which may run diagonally across the map, making every other polygon fall in its immediate neighborhood. A river is no problem in this system, since it is given the "water" feature code, which is treated differently in the various checking procedures (cf. Appendix "A"). As for the road, if the user assigns a neutral grayish color to it through feature code selection, it should not considerably affect other possible choices. If, however, a bright color is chosen, it it is quite correct that it should affect the choice of color for every other polygon, since the factors that affect our perception are the same as for any other polygon. If this is found to be undesirable in a particular case, breaking the road up in a series of smaller polygons and giving them the same feature code for coloring purposes would be a possible alternative to alleviate the situation. It is then that algorithm which is used to establish the list of immediate neighbors, the next item in the data structure.

A number of ways were also considered to establish the list of the further neighbors, the larger neighborhood. It first seemed possible to consider any polygon adjacent to the nearest neighbors as part of the larger neighborhood. This approach had similar problems to the one mentioned above. Any polygon adjacent to another may be so large that another small polygon adjacent to it would have no influence on the color of the first polygon from a perception point of view. A better approach was deemed to establish this relationship through an arbitrarily chosen radius. The centre of the polygon was considered as a starting point for this radius. This approach unfortunately suffered from non-reciprocity. Polygon A could be a larger neighbor of polygon B without the reverse being true. A set distance was then used from the borders of the polygon, another good use for the minimum and maximum X and Y coordinates. This also allowed to take into account the difference in the number of pixels on the X scale as opposed to the Y scale and the set distances could be adjusted accordingly. This approach ensured reciprocity and proved to be better for the proper selection of colors.

The anchor point is basically a point guaranteed to be in the polygon and not on a border. It is used for positioning the polygon identification number on the screen as well as a starting point for the coloring procedure based on the propagation method. This point is computed during the set-up phase. The algorithm used is based on the theorem which states that: Except for triangles, every polygon has at least two "ears". (Meisters, 1975). Simply explained, a "ear" occurs when, taking three consecutive vertices, the line linking the first vertex and the third vertex lies entirely inside the polygon. As long as the anchor point lies on this line, its position inside the polygon is guaranteed. Details on the implementation of this algorithm can be found in Appendix "A" under SETUPPOL.

Other data structures, less complex in comparison, also play an important role. The thresholds, for example, actually govern to a large part the behaviour of the system. When considering whether to use a color or not, the system will use the match/contrast value together with thresholds set by the designer or the user. The higher the value, the better the match is, and the lower the value, the better the contrast. Four threshold values are required as initial input data. The immediate match threshold, the larger match threshold, the immediate contrast threshold and the larger contrast threshold. Recalling what has been said previously about the match/contrast values of the different colors at the pixel level, it is obvious that these threshold values must fall between nine and forty-five, preferably between eighteen and thirty-six. The reason for using two match and contrast thresholds is the relative importance of the influence of the immediate and larger neighborhoods. As the immediate neighborhood should have more influence on the color chosen for a polygon, a higher immediate match threshold (or lower immediate contrast threshold) should be the governing criterion for the selection. Common sense and the type of map should also govern threshold selection. A map with a large number of small polygons and a high adjacency level should have a lower (or higher) threshold than a map with a few polygons since more colors must be chosen under more restrictions and having a wider selection is better. For example, a match threshold set at trirty-six would give only a few perfect match colors; set at twenty-seven, it excludes most neutral or "don't care" colors; twenty-six offers a good selection while preserving the essential criteria. The same value can be used for both thresholds, i.e. the mid-point 27, or different thresholds set. One can argue that if the decision on the threshold is left up to the user, and even worse, if the user has access to the knowledge base itself, there is no purpose to the system because it will be easy to mess up and will give its blessing to terrible combinations. On the other hand, if both are inaccessible, set by the designer, how can one experiment? The system can be seen as inflexible, of limited use. This is a philosophical argument over how expert-systems should be used. In this case, a procedure has been written to change the values of the knowledge base or of the thresholds interactively; these system access procedures could be removed from the appropriate "menus", if need be. Decisions of access to an expert-system are best left to the experts.

The above heuristics, built-in the data itself, constitute part of the intelligence of the knowledge base. Others will be in the form of minor data structures, applicable to parts of the program, or in the computing procedures themselves.

ORGANIZING THE KNOWLEDGE: THE APPLICATION OF GRAPHIC DESIGN RULES

It is obvious by the above discussion that, in a knowledge base, the major data structures are very closely linked to the algorithms used to manipulate them. They are at the core of the system and in their design embody a good part of the artificial intelligence manifested by the system.

But in addition to data structures, a knowledge base is also composed of rules to manipulate them. Rules form part of the processing algorithms, but some of these rules, which are more important in the context of the expert intelligence simulated by the system, can be extracted and brought together under the umbrella of the specific expertise shown by the system, in this case, graphic design with color. A dozen or more such rules of graphic composition and color harmony, explained in the previous chapter and the following appendices, have been implemented in this system.

First and foremost, the rules discussed in part one of this document such as: the rules related to the harmony of colors;

which colors are the best match for other colors; the affinity of warm colors and cool colors among themselves; the best contrast produced by complementary colors in the color circle and the contrast property of colors with a predominance of a hue whose complement is to be found in a lesser extent in another color; the effects produced by the adjacency of certain colors; Imhof's second rule on the placement of light, bright colors augmented with white and colors which tend to split the picture, etc.. All these and many more have been taken into account in determining the match and contrast values which form part of the main color data structure and are implicit in the choices made.

Secondly, size has been mentioned frequently as a visual variable which has a strong influence on the choice of color. The size of a polygon has been taken into account in the following way: the average of all the sizes of all the polygons is computed. If a polygon is larger than twice the average, then a more subdued color is chosen. This is consistent with the harmony of the global map neighborhood. Size is also considered for very small areas. A rule of good cartographic design states that if points, lines, and small areas are colored too light, they will disappear in the global picture. Bertin also stated that size is not associative. Small circles will not be as visible as big circles. It is thus important to make them stand out through the use of another visual variable, in this case, color. Therefore, when the size of a polygon falls below a certain pre-determined limit, the subdued colors are in turn avoided.

The rules about the influence of the immediate neighborhood on the color assigned to a polygon are present throughout the program. Whether the color assignment is automatic, done by the computer, or manual, dictated by the user, the table on match and contrast values is consulted. If the computer itself is making the choice and the color falls below the acceptable match threshold, then another color is chosen. If the color is selected by the user, the same procedure takes place, but, in this case, a warning is issued to the user that the colors clash, are not in accordance with his wish to match or to contrast the theme, or that other restrictions apply. The computer generally does not forbid the specific choice of a color by the user, but it gives him the alternative to change his mind before proceeding. This freedom of choice was built into the system on purpose as an expert-system should rather be seen as a help in times of need.

The rule of continuity in the image for cartographic terrain representation in contour maps has been implemented through the use of the special array containing contour colors similar to those of Imhof's example. This established scale can also be modified to deal with hypsometric tints used in other contour coloring methods simply by changing the numbers in the array. Associating certain colors to certain natural elements, which may be considered another cartographic rule, but is also plain common sense in its basic form, has been implemented for two basic elements: water and vegetation. Anything identified as water will be a shade of blue and anything identified with vegetation will picked from within the green hue range of colors. This is accomplished by identifying these areas through the use of feature codes and then following the pointers of the blue and green scales and checking if the chosen colors are within this scale. If an inappropriate color is chosen by the user, a warning is issued, similar to the one given for color clashes, that "water is usually blue" or "vegetation is usually green", and offering the user the chance to change his selection. There are many other cartographic color conventions that could have been included. For example, volcanic rock is red, Jurassic sediments are blue, coal is black etc.. These, however, deal with more specialized areas, such as geology, meteorology, arctic and alpine studies, and would require the user's knowledge and participation in identifying these areas. By setting down the guidelines for such procedures through the two basic elements, water and vegetation, the system demonstrates that these kinds of refinements are possible. The use of user-defined feature codes is an example of how the system can be given a color requirement for some kind of feature and be responsible for adhering to this requirement throughout the map coloring process.

The rules related to color harmony, theme colors, maintaining a basic color mood, mixing of color temperatures, keeping colors in one map to a manageable level, graphic interplay and color composition are evidenced by the dialogue which takes place with the user. In semi-automatic or interactive coloring mode, the program gives the user the choice of two color themes, one primary and one secondary. Having made that choice and indicated which features these colors apply to, the user is then asked if the rest of the map should emphasize matching or contrasting colors. The program then takes it upon itself to follow these directions, adapting itself to the user's choices. The manageable level is partially achieved through the use of feature codes. All elements belonging to the same feature code are the same color. Also, the program begins again at the start of the picking order for each new polygon. If that color is appropriate, it will be chosen. This reduces the number of colors used in the map. If for visual effect, the user prefers to use all different colors, as in, for example, a map of the United States where each state is of a different color, this can be easily achieved through the manual option.

The preference of a subdued color for the background is explicitly implemented by having the user identify the background polygon and the system choosing a subdued color for it. Finally, the threshold used for determining if a color is appropriate or not is another rule which the computer uses constantly. These thresholds can be changed if not enough colors exist to properly color the actual map or to see how far one can go towards perfection. Its pre-set level ensures an adequate choice of colors while adhering to the composition and graphic interplay rules previously described.

Evidently, if one starts dissecting all the specific rules mentioned in color graphic composition guides, certain of these rules will not be found explicitly in the system. In many cases, they are "preference" rules, the little extra touch that makes a graphic composition particularly striking. This system covers the basic color graphic composition rules; the interesting fact is that as long as there are artists, cartographers, designers and other experts striving for perfection, it can be built upon and learn new tricks, as any being with intelligence would.

USING THE KNOWLEDGE: PROGRAM PHASES

SET-UP PHASE

Having covered the knowledge base aspect of the system and looked at some of its heuristics, it would be useful to cover briefly the sequence in which it proceeds and some of its more basic procedures. More details on the type of variables used and on the actual algorithms can be found in the documentation contained in Appendix "A" of this document.

The first action of the program belongs to what can be referred to as the set-up phase. This is where the color structures are filled in followed by the polygon structure. The filling in of the color structures has already been mostly described under the color data structures section. It mainly consists of reading in the values required for the match/contrast table, computing the color identification number, reading in the next color in hue, dull colors, contour colors, thresholds, hue identification characters etc.. The polygon structure, however, requires a number of pre-processing steps. The polygon identification number and the minimum and maximum X and Y coordinates are established when the points making up each polygon are read in. Once a polygon has been read, its size is computed and its outline is drawn on the screen with its identification number. To read in the polygon vertices, a temporary array is used over and over again, destroying the previous information on a single polygon so that all that remains at that point is the information kept in the polygon structure. After all the polygons have been read in, the immediate and larger neighborhoods of each polygon are computed and stored in the appropriate record. Size "limits", over which a polygon is considered large or under which its is regarded as small, are also computed at this time. The rest of the information is then obtained by the first set of interactions with the user. This is where the water, vegetation, and other feature codes are identified, the background information is obtained, and the user's selection of a coloring procedure is made. Six choices are offered on the menu: automatic (the computer colors the map without any help from the user), manual (the user dictates individual polygon colors), semi-automatic or shared (the user can select a primary theme or color as well as a secondary theme to apply to specific features), and the special case of contour map coloring in accordance with Imhof's cartographic conventions. The fifth choice is the "system" choice, or knowledge base access, and the sixth is to quit the program.

Specific to this implementation, the three main routines which make up this phase are named: SETUPFC, SETUPCOL and SETUPPOL.

INTERACTIVE OPTIONS PHASE

The system option is a series of procedures through which one can change the parameters governing the intelligence displayed by the system: the threshold values, the contents of the match/contrast value table, the dull colors and the contour colors. This option also allows the user to get a hard copy on the printer or store in a disk file the values contained in the above tables which form the knowledge base, as well as the content of all the polygon records. Finally, it can display all the forty-nine colors in the space available on the top of the screen without disturbing the map. This is also the only option which offers a return to the main menu after each choice, so it is ideally suited for experimentation. Useful for debugging, it remains so after the system is properly working to make any changes, experiment with different schemes, add information, monitor performance and generally find out the state of the system after the set-up phase. A question as to whether part of this option could have been left out of the menu is based on the concern for accessibility of the knowledge base itself expressed by many adepts of artificial intelligence. In this way, the procedures which affect the behaviour of the system, could have been kept removed from its working procedures. However, in an experimental system, knowledge base access is good for verification and experimentation. As a compromise, the design of this option is modular and menu-driven, so that any choice could be removed from the appropriate menu should the need arise.

The automatic procedure is basically system driven. When one person makes all the choices it is, in a way, easier; the same applies to a computer. The system must ensure that water is blue, vegetation is green, anything else identified as belonging to a particular feature code is the same color and that a color identified for a specific feature code becomes "reserved". It then proceeds to color polygons considering the impact of the immediate and of the larger neighborhoods. A subdued color is chosen for the background and large polygons unless these polygons have already been identified as having to be the same as a particular feature code. Subject to the above criteria, small areas are not assigned a subdued color and a match criterion is applied to neighborhoods. To avoid always repeating the same colors every time the program is run, the user is given the choice of picking a starting hue. A starting hue is different from a thematic color as will be seen in the semi-automatic option. What it means is that the program, when looking through all the available colors, starts at a particular hue. Obviously, if a color meets all the criteria, it will be chosen, giving the map a definite predominance of colors in that hue. However, there is no guarantee that a certain area will be any specific color except, of course, that water will be blue, vegetation, green, and that areas identified as belonging to a specific feature code will be colored the same with no other polygons having that color.

The semi-automatic procedure is more complex but interesting, as it allows more exchanges with the user. It applies only if the user has identified at least one special feature. Otherwise, it is the same as automatic. The system first lists the color themes available. The user is then given the choice of identifying a primary theme or a specific primary color. The difference is that the system has a choice of any color within the primary theme hue but that it must adhere to the user's choice of a specific primary color. If the user has identified at least two features, he is then given the choice of identifying a secondary theme from the ones listed. The system then lists the user-defined features and asks to which specific feature the primary theme or color should be applied as well as for the identification of the secondary theme feature, if applicable. The next action of the system is to ask the user if he wishes the rest of the map to emphasize matching colors or contrasting colors. If a specific primary color was chosen, all the polygons bearing the appropriate feature code are assigned that color; if a primary theme was chosen, the system selects a color within that thematic hue. Since this is the first action performed by the system in terms of color assignment and no other polygon has therefore already been colored, the selection of an appropriate primary theme color is guaranteed. The secondary theme is the subject of more constraints as the match/contrast values and other appropriate criteria, such as size, are now considered with respect to the primary theme or color, should the polygons identified fall into each other's neighborhoods. Usually, this is not much of a problem since a theme has several color choices, polygons may not be adjacent, or a good choice of themes by the user ensures that the selection will be appropriate. However, it is possible, in some cases, that an appropriate color within the secondary theme hue may not be found. For example, let us say that the user chose bright yellow as a primary color and blue as a secondary theme, that polygons identified as belonging to the primary theme are adjacent to polygons belonging to the secondary theme, and that the user identifies that he prefers matching colors. As evidenced by the color circle, blue and yellow are and excellent contrast but a poor match. Then it is unlikely that a blue will match the yellow and the system cannot select a color meeting all the required criteria within the blue hue. This, of course, also depends on the established thresholds and the proximity of the polygons in question. Should such a case arise, the system then returns to the user offering him suggestions before starting the process again. These suggestions are basically the foundation of the knowledge base, i.e. listing the best match for all colors in the color chart as well as the best contrast. The user can then see that selecting yellow and blue as well as match is contradictory. The appropriate response would then be to select other matching theme colors or use contrast as the base criterion for the map. Once the primary theme/color and the secondary theme have been taken care of, the system then proceeds in much the same way as the automatic procedure, looking at all the other feature codes, then at polygons with no feature code. The different results mentioned previously that can be produced on the same map using the automatic procedure with a certain starting hue and using the semi-automatic procedure with the same hue as a primary theme and a matching criterion, can now be more easily explained. The primary theme is looked at first, with no other polygon having yet been assigned a color and the specific feature code to which it applies is identified by the user; the starting hue is considered after water and vegetation have already been taken care of and it is not reserved for anything in particular. This can create a map with a similar color mood, different in the actual coloring details.

For these two procedures, the colors are first assigned internally, then are put on the screen at the end. This allows some form of backtracking to take place, as all polygons with a certain feature code are tested before a color is assigned. However, once a color has been assigned by a certain procedure, no more backtracking takes place.

The algorithm for backtracking can be summarized in the following way. The program first considers polygons which have a special feature code. Subject to the appropriate hue (H) restrictions imposed either by nature (blue, green) or by the user (theme), a color (HC) is chosen for the polygons having feature code "X". If acceptable, this color is then tested for all other polygons having feature code "X". If at one point the test fails, color HC is then abandoned and the search starts again with color HC+1, until all possible colors in that hue have been tested or a color is chosen. For polygons without feature codes, all colors are considered for each polygon and the first acceptable color, or first fit, is chosen. Obviously, the search must follow some kind of order. For example, under the shared option, the order is: primary theme, secondary theme (if any), water, vegetation, other user-defined feature codes in arithmetic order, polygons without feature code in arithmetic order (1,2,...N). It does mean that the last polygon to be looked at will be subject to a greater amount of constraints than the first, which can also lead to the conclusion that presenting polygons as data in a different order may result in a different coloring scheme. While this may be true in some cases, is it really important that polygon P1 be colored 334 and polygon P2, 443, or the reverse? This possible variety may add to the versatility of the system. Since all available colors are considered, the experimentation has not shown that the order makes a significant change in the overall map in all cases. However, as an extension, the system could be made to change the order of the search if any scheme fails, resulting in more time being spent on the search itself. The internal backtracking mechanism makes this option a real possibility, since no color would be put on the screen until the final answer is obtained.

Assigning the colors first also served another purpose. The complete internal selection and assignment process is very fast, while the actual setting of pixels on the screen is very slow in comparison. For a thirty polygon map, all colors can be chosen in a split second and take some twenty minutes to appear completely on the screen. In the experimental stage, having the program print the colors it would choose instead of actually coloring the polygons, speeded up the process considerably. Even in the final analysis, this process seems to be preferable as it leaves open the possibility of increased exhaustive searching.

With all the imposed restrictions, is it possible for the program not to find an appropriate color? Sometimes. So many factors can influence the choices that some combinations may not be aesthetically possible to achieve. It is less likely to occur in the automatic procedure, where the program makes all the choices, than in the semi-automatic procedure, where constraints are imposed by the user and the program must do the best it can with what it is given to work with. Some of these factors are the shape of the map, the number of polygons, their size and their relationship with each other, the type of choices made by the user in selecting the themes, and, most likely, the thresholds that have been set. This is why, when the program goes through all the available colors and cannot find a suitable one, it simply leaves that (those) polygon(s) uncolored and proceeds with the rest, the secondary theme being the exception to the rule where the program returns immediately to the user without continuing any further. At the end, the main program takes over and, if some polygons remain uncolored, it gives a message to the user of the general form: " I could not appropriately color certain polygons. Next time try changing the thresholds. They are probably too high (for match) or too low (for contrast)".

Thresholds are user controlled and should constitute a sufficient basis to effect a change in the colors chosen. Although changing the order of the data has been suggested, it does not seem to be a good way of proceeding as it would involve more work on the part of the user and a very large number of permutations are possible. Implementing this approach would best be done internally, as suggested in the preceeding paragraphs, bearing in mind the increased time needed to reach a conclusion. For a practical system, threshold modification will, in most cases, result in a good colored map unless the user has made purposely contradictory choices, in which case the order would not improve matters either. However, again at the expense of increased time spent on the original computation, the threshold change option could be done by the system. Instead of the above message, the system could start again, decreasing its match thresholds by 1 to 3 and increasing its contrast thresholds accordingly. When a solution is finally found, it could then tell the user that it is the best it could do under the circumstances, and had to adjust the thresholds to the values finally chosen. The implementation of such an option depends greatly on the basic philosophy used. Should the expert-system work in conjunction with the user and be a truly interactive system providing a learning experience as well as user involvement, or should it do things on its own without "bothering" to ask the opinion of the user as much as possible? This system was based on the first approach, although the second certainly has its merits and may be considered as a possible extension.

With appropriate thresholds and choices, the above-mentioned situation, where some polygons remain uncolored, will not occur often; surprisingly, as discussed under results, the program often performed better, i.e. gave a more attractive product, under automatic than under shared or even sometimes manual which is the next option to be discussed.

The manual option is user controlled. This is where the user dictates the color of individual polygons to the program, and it is well suited for a learning or independent choices environment. The user is first given a set of commands that can be used: Help, to see the commands; Show, to show a sample of a particular color on the top right hand corner of the screen; Dither, to see a sample of all available colors without interfering with the map; Opinion, to get the same suggestions as mentioned under the semi-automatic procedure; and Color, to actually color a polygon. With the Color command, the user is asked for the id. of the polygon to color and the id. of the color he wishes to use. The following checks are however performed: if a polygon has been identified as being water or vegetation and the color chosen does not belong to the blue and green hue respectively, the user is warned. A warning is also issued if a polygon has been identified a belonging to a particular feature code, a polygon with that feature code has already been colored, and the color chosen is not the same as the color assigned to that previous polygon; in this case, the new color can be identified as the color to use for that feature code and it will then serve as the basis for the check from then on. Similarly, if a polygon does not belong to a particular feature code and the color chosen has already been used for polygons with that specific feature code, another warning is issued. Finally, the match/contrast criterion check is applied as follows: since the procedure is mostly designed for experimentation and to agree as much as possible with the wishes of the user, only the immediate neighborhood is checked for either a good match or a good contrast. Which means that if all other tests have passed, the only colors which will fail this last test and produce a warning are those for which the computed match/contrast value with the immediate neighbors falls under the immediate match threshold and over the immediate contrast threshold. It is possible to bypass this check by setting the immediate match threshold low enough and the immediate contrast threshold high enough for most colors to be acceptable.

The warnings take several forms. Sometimes a color is shown to the user with the mention: "Water is usually blue. You have chosen (sample of color shown). Do you wish to change it (y/n)?", or, "You have already chosen (color id.) for (feature code name). Do you wish to change it (y/n)?", or even, "Would you like to use (color id.) from now on for (feature code name) (y/n/)?". In all cases, except the last, the procedure starts again for that polygon if a "yes" answer is given. If the user does not wish to change his mind, than the color is accepted regardless of the results of the tests. It is a user controlled procedure. This is particularly useful when one literally "paints oneself into a corner". Amazingly, when considering that under the automatic mode, the program can choose some thirty perfectly matching colors in a split second with all the other criteria being respected, it is surprisingly easy to do so in the manual mode when coloring polygons one by one. Forgetting to consider the little polygon in the corner belonging to the same feature code than the large ones on top, choosing colors to obtain a good variety without considering the consequences for the one polygon that has six immediate neighbors with differently colored hues, simply not paying too much attention to the global picture when making choices, can all result in a catch-22 situation. Sometimes, this way of coloring can be done on purpose, like coloring the 48 continental states each with a different color, which is unlikely to occur using the automatic or the semi-automatic options as these are based on the principle of using a reasonable number of colors, repeating the same color when appropriate. With this trial an error option, anything is possible and yet control is available from the system when desired. Polygons are colored when the color is chosen.

The final coloring option, contour map coloring, follows an example established by Imhof and proceeds in accordance with the given scale of contour colors. The procedure first lists the height of the thirteen steps available. As each polygon is listed without interfering with the scale which stays on the screen, the user is asked to identify the step to which it belongs. The only rule applied is that water overrides elevation and a special color is reserved for it, which means that a lake located at a high elevation will still be turquoise rather than brownish. The system then colors all the polygons with the color identified as belonging to the identified step. Changing the color used for a specific step only requires modifying the input to the knowledge base or proceeding interactively through the "system" option.

This phase is, in the implemented program, mainly composed of the routines named: MANUAL, SHARED, AUTO, CONTOUR and SYSTEM.

THE COLORING PROCESS

Because of the color dithering process and the minimal information kept on each polygon in the knowledge base, actually coloring the polygons offered a special challenge. The main coloring algorithm used is based on the propagation method, augmented by other tests applicable in certain circumstances.

The propagation method consists basically of the following algorithm: 1) pick a point inside the polygon and color that point; 2) set x-start to the x coordinate of that point minus one, x-end to the x coordinate of that point plus one, y-start to the y coordinate of that point minus one and y-end to the y coordinate of that point plus one; 3) do a double loop, x-start to x-end, y-start to y-end, and for each point[x,y]; 4) test if that point is empty; 5) if so, test if it is next to a colored pixel; 6) if it is, color that point and otherwise skip it; 7) reset x-start to x-start minus one, x-end to x-end plus one, y-start to y-start minus one and y-end to y-end plus one; 8) repeat steps 3-7 until no change occured during the loops.

This process works well for any type of polygon with a single pure color, testing ever increasing rectangles around the starting point. Points on the top left hand corner or other points that do not meet the criteria the first time eventually do so because each point is tested over and over again.

The first problem to consider in this case was the dithering of the colors. Somehow, the computer must be able to re-create the dithering pattern each time a point is colored. Left to its own device, changing the color in a circle of the three colors used to create the dithered color each time a point is colored, the result obtained was fascinating as abstract art but did not look very good on a map because the pattern of points being colored changed according to the shape of the polygon. The solution was to design a function which would respect the dithering pattern regardless of the order in which the points were colored. Taking the total number of X coordinates (columns) available for the map we could say that, if it were one big polygon colored with color id. 334, the points of the first column would be colored with color 3, the points of the second column would also be set to color 3, the third column to 4 and the pattern would repeat itself up to the last column. This led naturally to devising a function based on modulo 3 plus 1. The three pure colors used to make a dithered color are separated and kept into a three element array at the beginning of the procedure. For each point which passes the tests and is to be colored, its X coordinate is then divided by three, the remainder plus one being the subscript of the location of the array which indicates the color to be used. Polygons of any shape can then be colored knowing that the dithered color will look as it should.

Another big problem consisted of the dual role played by the the background and the border colors during the design of the map. In this particular implementation, the background was black and white was used for borders; other computers may have a different screen color. Therefore, during the following discussion, one can substitute "background color" for any reference to black and "border color" for any reference to white. In this system, if black and white were to be used as colors available for dithering, as was originally intended, and also as representing respectively empty/uncolored pixels and borders and identification numbers, problems arose in the coloring process. The propagation method of testing individual pixels in ever increasing rectangles means that pixels located outside the polygon will be tested as well as pixels inside the polygon. If a polygon borders an uncolored polygon, the test for a pixel away from the border being empty will be true, the adjacency test to a colored pixel will be false and there is no problem; if, in addition, we add as condition for the test for adjacency, adjacency to a pixel colored with one of the colors we are looking for, i.e. a color to be used inside the polygon, bordering a colored polygon colored with other pure colors (except white) is also no problem. However, what if a polygon was colored with a color composed of the black, black, blue dithering pattern and the next polygon was to be blue, blue, green. It is conceivable to find an empty/black pixel next to a blue pixel and give it the color green. This would be be all right if the point tested was inside the polygon but a disaster otherwise. Testing to see if the point is in the polygon has another major flaw. Most point-in-polygon algorithms rely on the "plumb" line method or something similar. This consists of "dropping a line" from the point in question to the lower border of the polygon and testing for the number of intersections with a border. An odd number of intersections means that the point is inside the border while an even number means that it is outside. This is very well if white only serves to identify the borders, but if it is also a color, or used for other purposes such as identification numbers, intersecting a white pixel does not mean a thing. There is a 50/50 chance of being right which is, of course, unacceptable. A version of this point-in-polygon method was implemented for coloring polygons dithered with white but it also presented some problems, as discussed below. Since after long due consideration and experimentation it was found that there is no sure way of making the difference between an empty pixel and a pixel colored black in every case, even for a person looking at the map at the pixel level, it was decided to remove black as a possible dithering color and reserve it for empty/uncolored pixels, which accounts for the forty-nine colors available instead of the original sixty-four possibilities.

White presented similar problems but "dropping" white would only give thirty-six colors and colors dithered with white serve very good purposes as dull and neutral colors, in addition to being aesthetically pleasing. One difference with black is that white is a proper color, lessening the problem somewhat. A solution was found by fooling the program into believing that white is an "illegal" color while the testing process takes place. This ensures that any empty point adjacent to a border and surrounded otherwise by only black or white pixels will not pass the second test, keeping the propagation of colors outside the polygon borders from happening. The dithering method used proved useful. When a color dithered with white is chosen and the program separates the color digits into the array, it replaces any white digit (7) with the other color, if applicable, or with blue, if pure white. It also keeps track of the applicable subscript(s) and keeps a list of the affected columns, using the dithering function. It then proceeds through the algorithm as usual, but with one additional job. If the column is one which should have been colored white, it keeps track of the minimum and maximum Y coordinates for that column. The result is that if a polygon should be colored with the red, red, white dithering pattern, it will appear pure red on the screen at first. At the end, the program goes through the list and resets the appropriate pixels to white. To help with polygons which are laterally concave and for which proceeding directly from the minimum to the maximum Y coordinates would not work, a version of the "plumb line" point-in-polygon algorithm was used. Any non white pixel is reset to white. For any white pixel, the pixel below is tested. If it is not white, the number of intersections with white pixels to the bottom of the polygon are counted. According to the point-in-polygon algorithm described above when discussing black pixels, an odd number of intersections means that the point is "in", and an even number of intersections means that it is "out". This method generally works well except in a few cases such as when another area borders in between the protruding sections, or if the identification number of the neighboring polygon, drawn in white, falls in the concave part, as indicated by the illustration which follows. These problems are unpredictable and depend on the shape of the map. The "alligator tooth" polygon, used in computational geometry test cases, could also cause some confusion. But as it is not an usual case in maps and can easily be taken care of by changing the affected coordinates by one pixel with no visible effect on perception or on the general shape of the land, it does not cause a major problem.

This was not a problem originally, when the program kept track of the coordinates of each affected pixel individually. Unfortunately, pixels being what they are, small and numerous, it tended to run out of stack space to keep the list, resulting in the above modified method. Contour colors have no white in them, avoiding this situation. Another alternative for contours would have been to always color the polygons by nesting levels, the most nested first. This would however require the user to identify the appropriate level of nesting information for each polygon and keeping this information in the polygon structure. Then white could be used in contour colors but at the expense of

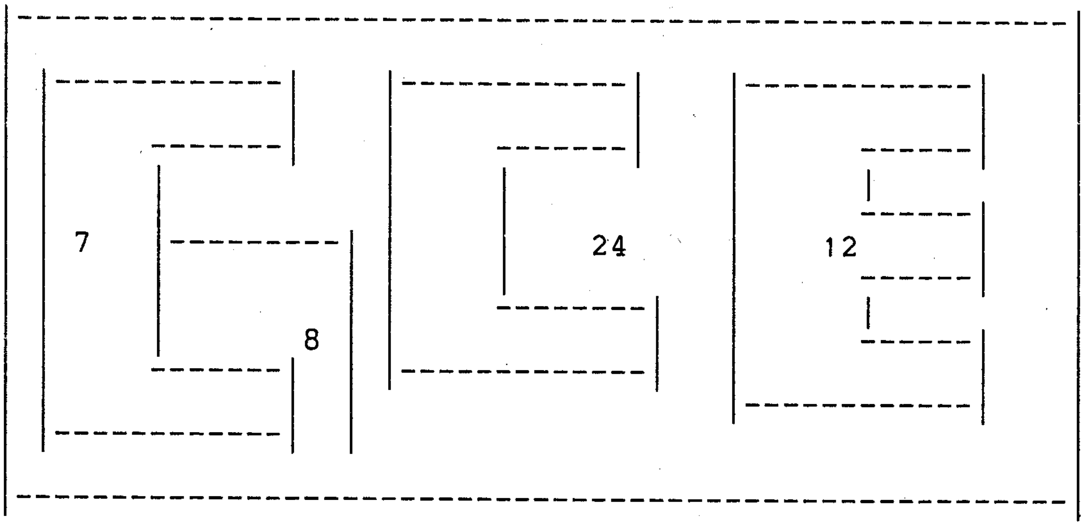


Figure 2.3 Exceptions due to the use of the border color.

a little more storage, some diminished independence resulting from the need for additional user input and an extra level of time and complexity during the coloring process itself.

The next problem, the white identification number itself, was easily taken care of by resetting the number to black before proceeding. White in other polygons being "illegal" anyway, it does not interfere with the algorithm. Shouldn't this method work equally well with black? It does inside the polygon. However, when a pixel outside the polygon is tested, the column in which the black pixel is may not be the column in which the black pixel would be in the new dithered color.

Black is then "out" but white is "in", with no problem at all if the polygons are of usual shapes with no nesting or if enough memory space is available to keep track of the individual affected pixels, and with a few exceptions if the algorithm grouping pixels by columns and using the point-in-polygon checking feature is used.

The coloring problems solved, another one remained. The method is very slow as it tests each pixel over and over again. It is however, accurate with any shape of polygon. A way was devised to speed it up somewhat for certain polygons using the information kept in the polygon structure and some boolean variables. When the propagation of a polygon with no nested polygon inside it encounters the limits of the polygon in either direction, i.e. the minimum and maximum X and Y coordinates, except for the upper left hand corner, there is no need to retest pixels in that direction. It can then skip directly to the last row or column ihe other direction. "Blocking" boolean variables were used to achieve this effect, speeding up the process. A problem occurs when some nested polygon hinders the normal propagation in all columns, creating a shadow effect where some pixels will remain uncolored after the loop starts "jumping". With the normal method, these pixels would be tested over and over again, resulting in proper propagation. For testing purposes, this modification was used when no situation risked of producing this effect. With large polygons, the first method can be frustrating to wait for but it works for any type or shape. It could be suggested that, at a later date, it may be possible to add another level of intelligence to the program, making it decide by itself which modifications to apply to the basic algorithm according to the circumstances. Would this decision making slow down the process obliterating any gain in speed? Are the criteria obvious? Where should it fit? These are questions to be explored under the umbrella of graphics design and image processing tied in to proper artificial intelligence knowledge engineering before any further attempt takes place.

Shortly before the publication of this document, a third method for speeding up the coloring process was suggested. This method, however, would require a change in the pre-processing process. It is understood that keeping a record of all vertices and edges would allow for faster coloring algorithms but, as one of the goals of this program was to use the minimum storage required for implementation and use on available microcomputers, this approach was not used. This third suggested modification would basically consist of finding and storing local maxima and possibly single border pixels which do not separate two different polygons but are at the bottom of a "V" shape. Then, by giving the program a list of all the polygons which are on the top line with their respective color and blanking out all identification numbers at the start, feasible under the automatic and the shared options, the coloring algorithm could proceed line by line from top to bottom. On encountering a white pixel, it would check if it is the maximum Y of a polygon, a local maxima, or something else. If it is the maximum of a new polygon, the linked list would be updated and the new polygon inserted; a local maxima would indicate a split in the following lines; on reaching the minimum Y of a polygon, it would be removed from the list, while the same effect would be achieved on encountering the single pixel at the bottom of the "V", indicating the end of a split in a polygon; an ordinary border would mean going to the next color on the list. The modulo 3 plus 1 function would still have to be used to maintain consistency in the dithering pattern. As this algorithm has not yet been implemented and tested, the additional requirements in storage and computational complexity, as well as correctness in all cases, have not been determined. At first glance, however, it seems that it could speed up the coloring part of the process as well as allowing black and white as colors since it would never test the same pixel twice.

As in any worthwhile intelligent program or expert-system, there can always be refinements. Achieving perfection on all counts takes time and someone will always come up with another idea to add. This attempt to design an expert-system dealing with such a subjective subject as color proved that it can be done and many refinements could naturally follow without much change in the main structures. Others would necessitate a new direction of thought. Still others may benefit from assembly language implementation or other heuristics or tricks of the trade. As it is, the "expert" is quite knowledgeable.

EXPERIMENTATION AND RESULTS

An obvious question remains to be answered. Once implemented and tested on various maps, what did the results show? Did they, in fact, live up to the expectations and produce attractively colored maps within the framework of good graphic composition? Some photographs of the design process and the completed colored maps obtained are included for reference during the following discussion.

The program was tested on various types of maps using different parameters. Among the maps used were geographical maps, such as a map of part of the United States yielding thirty polygons with some reserved as "water" areas, thematic maps such as an interior map consisting of offices and corridors of the type found at various places in a building which was ideal to test feature codes and themes as well as how polygons could be grouped differently without any natural color restriction, and maps emphasizing the contour colors such as a gradual scale of the thirteen steps available.

For each of the maps, excluding contours, the automatic, semi-automatic and manual procedures were tried, the contour maps being, of course, limited to the contour map coloring option. Apart from choosing different starting hues and themes (primary and secondary), and trying different combinations with the manual option, changes were made in the knowledge base until dull colors and contour colors appeared to be the best, in the match/contrast values, the order in hue, and mostly on thresholds. The match/contrast values proved to be satisfactory as first designed. The membership in hue was adjusted to fit daylight and nighttime viewing as well as near and far viewing with as much accuracy as possible, this being a perceptual criterion. The thresholds proved to be the most versatile testing criterion, capable of changing considerably the appearance of a map, given the same parameters. The obvious conclusion is that the more restrictions one puts on the program in terms of features, choices and type of map, the larger the margin that should be given to it in terms of threshold setting for it to do a good job. This could be equated to the degree of freedom and flexibility given to a human designer in the same circumstances. Other adjustments were made within the program itself concerning, for example, the radius used to determine the larger neighborhood and the size boundaries function until a satisfactory formula was found to apply universally to the maps tested. Details were given during the discussion of the appropriate procedures.

As previously indicated, in the section dealing with the program phases, the automatic procedure worked very well given middle of the line immediate thresholds (around 27). The semi-automatic also did well but was dependent on the number of feature codes identified, the number of polygons and their topological relationships, and sensible theme colors and match/contrast criterion being chosen. The example given previously of the blue, yellow , match combination did, of course, create some problems, but these were not due to the computer but rather to poor artistic judgement on the user's part. Thresholds had to be adjusted with regards to the above constraints. The manual procedure performed as could be expected with appropriate warnings being issued at the right time. The contour map coloring scale is attractive and approximates as much as possible the scale used by Imhof in one of his examples. Pure match could not be achieved as few dithered colors give brown, slanting towards an orangy-yellow combination instead. Also the removal of black as a color forced the use of some turquoise shades for the underwater steps, as not enough blues were available. As explained in the coloring process, the necessity to avoid colors dithered with white for this purpose also contributed to the choices made. The system procedure also performed as required and proved itself very useful.

To go any further, it is now necessary to refer to the maps themselves, as a simple description does not convey the best possible message and nuances obtained. The following will then refer to pictures taken of selected maps, together with the parameters imposed to the program to arrive at this choice and some of the dialogues exchanged.

The pictures which follow represent some of the results of the set-up phase and the work of the various coloring modes.

The first picture is the map as it appears on the screen after the data has been read in and SETUPPOL is completed.

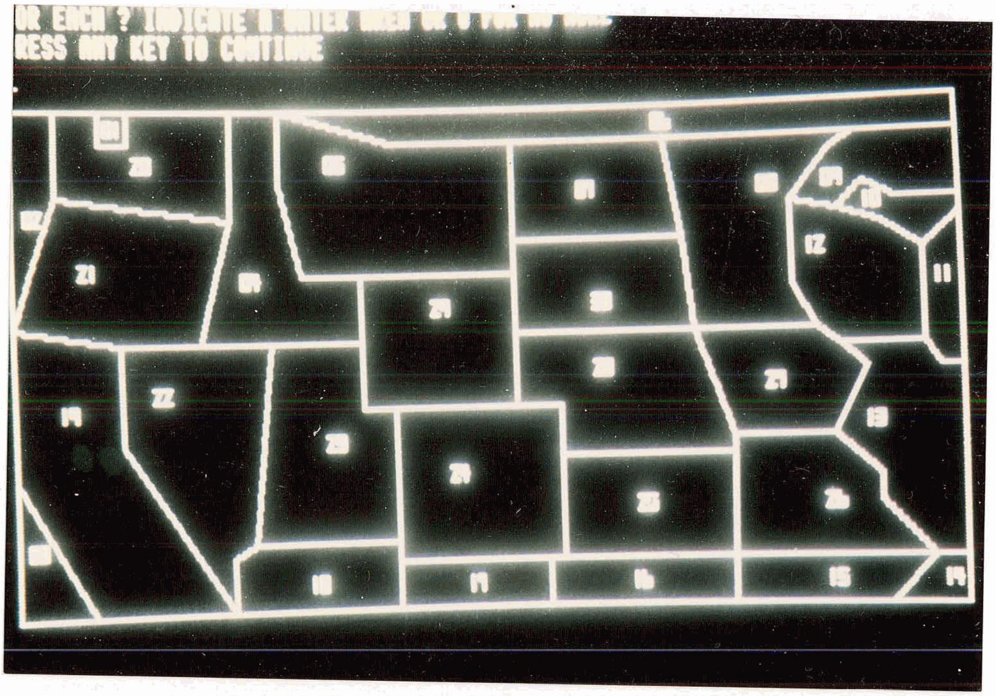
(Figure 2.4). The data consists of a series of (x,y) coordinates in clockwise or anticlockwise order, for each polygon. The program reads those in and draws joining lines forming the borders. As each polygon is read, a number is assigned to it and this number is then also drawn on the screen for future identification reference during the dialogue with the user. Note that in this implementation, the background screen is black and the borders and identification numbers are drawn in white. By this time, all the necessary information about the polygon, i.e. size, neighborhood, minimum and maximum X and Y coordinates, etc., has been computed and stored in the polygon structure. This is where the interactive phase begins and the user is asked to identify the water areas, vegetation areas, any other special feature code he wishes to use and the background, if any. The set-up information being acquired, the program is then ready to move into the coloring phases.

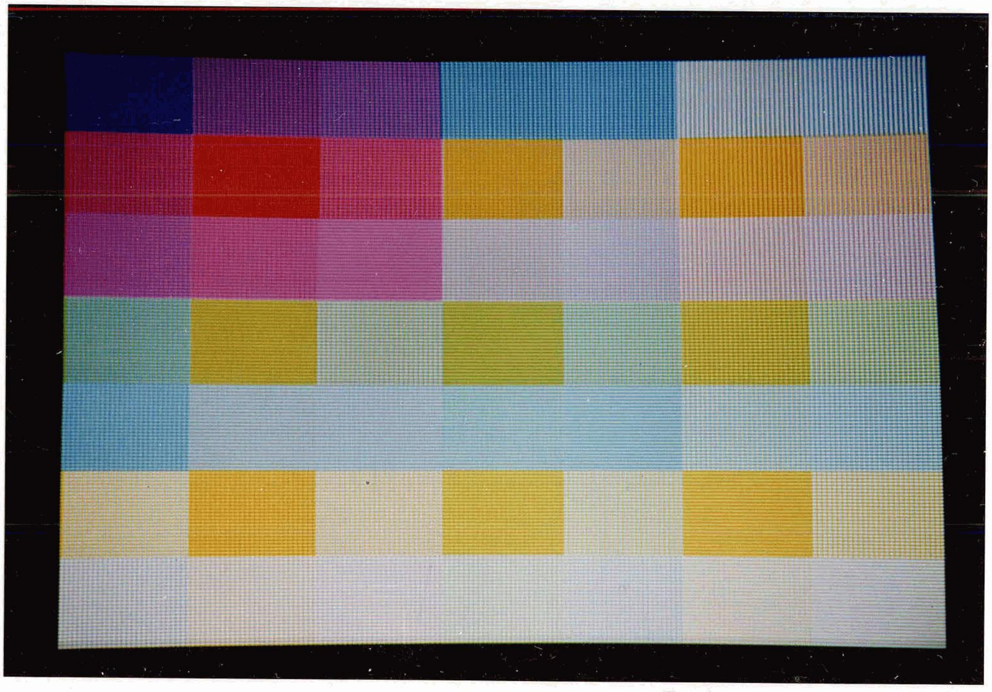
The second picture (Figure 2.5), represents the forty-nine colors available with the dithering process (excluding black). These are the colors which will be used to produce the following photographs. This entire chart is available in a reduced format (not to interfere with the map on the screen), during the program execution. It can be called from the "system" option, or from the "manual" mode. It then occupies the six top lines of the screen.

In the automatic mode, only one question is asked of the user:

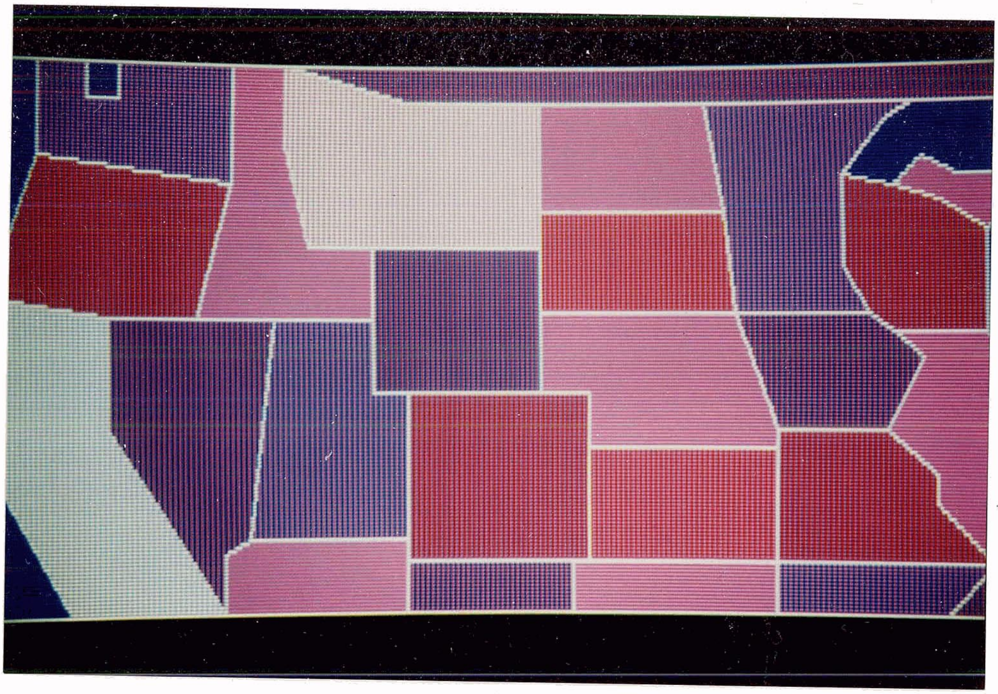
"Do you wish to pick a starting hue for me?
1.Blue 2.Cyan 3.Green 4.Yellow 5.Red 6.Magenta
2.Anything ... Enter choice number:"

The program makes all the choices. In the two illustrations which follow, magenta (Figure 2.6) and yellow (Figure 2.7) were chosen as starting hues. Notice that in both cases, the areas identified as "water" are colored blue. Choice number 7 (anything), is a pre-set default value which can be any of the above.





Figures 2.4 and 2.5



Figures 2.6 and 2.7

The shared mode is more complex. This is where the user can pick a primary theme or color and a secondary theme which is directly applicable to user identified selected features. The constraints imposed on the program are more numerous. The following five pictures were produced using the shared mode.

A typical dialogue follows, showing how this next picture of Europe (Figure 2.8) was obtained. It also shows the dialogue which precedes the coloring phase and obtains feature codes. In this map, three special features were identified in addition to water. France, Russia and Greece were grouped and used for the main theme color. Italy by itself was identified as a feature and used for the secondary theme. The two polygons which make up Turkey were identified as one feature in order for them to be colored with the same color.

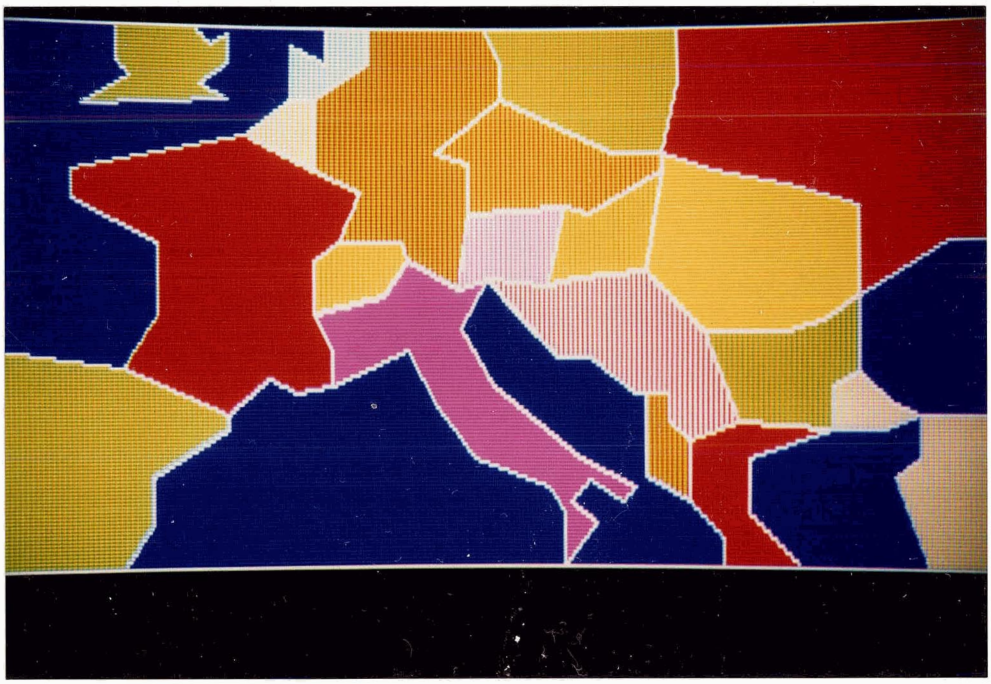


Figure 2.8

```
---------------------------------------------------------

welcome to the map coloring program
before starting, I need to set up your map
setting up ... please wait ...

this is the map as read
we will now identify special features
we will start with water areas.
for each ? indicate a water area or 0 for no more
press any key to continue

? (user lists water areas,terminating with 0)
  (in ex. map, 5 areas were identified       )

continuing now with vegetation areas.
for each ? indicate a vegetation area or 0 for no
more
press any key to continue

? 0

how many other groups would you like (max 8)?  3

enter the group name followed by a period: FRG.
for each ? indicate an area or 0 for no more.
press any key to continue

(France, Russia and Greece were identified for )
(this group. The above question repeated itself)
(another two times. Italy was identified second)
(and the two polygons forming Turkey third.    )

enter background or 0 for none:  0

which of the following options would you like?
1. I color the map       2. You color the map
3. We color the map      4. Contour map coloring
5. Quit program          6. System
Enter choice number: 3

the available color themes are:
1:Blue  2:Cyan  3:Green  4:Yellow  5:Red  6:Magenta
as a 1st choice, you can select a theme or a
specific chart color. Select 1:theme or 2:specific
color... enter choice number:  2
enter specific color number: 222

if you wish a secondary theme, enter theme no.
or 0 for none: 6

the features you have identified are:
 3  FRG      4  ITALY     5  TURKEY
```

```
press any key to continue

(replacing first line only)
for which feature do you wish to use your
primary theme? Enter the feature number: 3
for which feature do you wish to use your
secondary theme? Enter the feature number: 4

if you wish the rest of the map to emphasize
matching colors enter "1" or contrasting colors
enter "2" : 1

working....

colors chosen   (At this point, the color chosen
                 for each polygon, is written on
                 the device mapped to file 01 and
                 the actual coloring of polygons
                 on the screen begins)

(when all polygons have been colored)
program ends... bye for now.
```

Figure 2.9 Typical dialogue in shared mode.

This picture is a "zone" map, where the "blue" polygons were identified as water, the "red" as a feature code and the yellow areas as another. (Figure 2.10). The background was also identified. The areas not covered by a special code were colored as follows: another shade of yellow (centre), pale cyan (centre), magenta (top right), and white (top centre). The grayish/blue chosen for the background is a dull color which matches its surroundings while giving enough contrast to differentiate between areas.

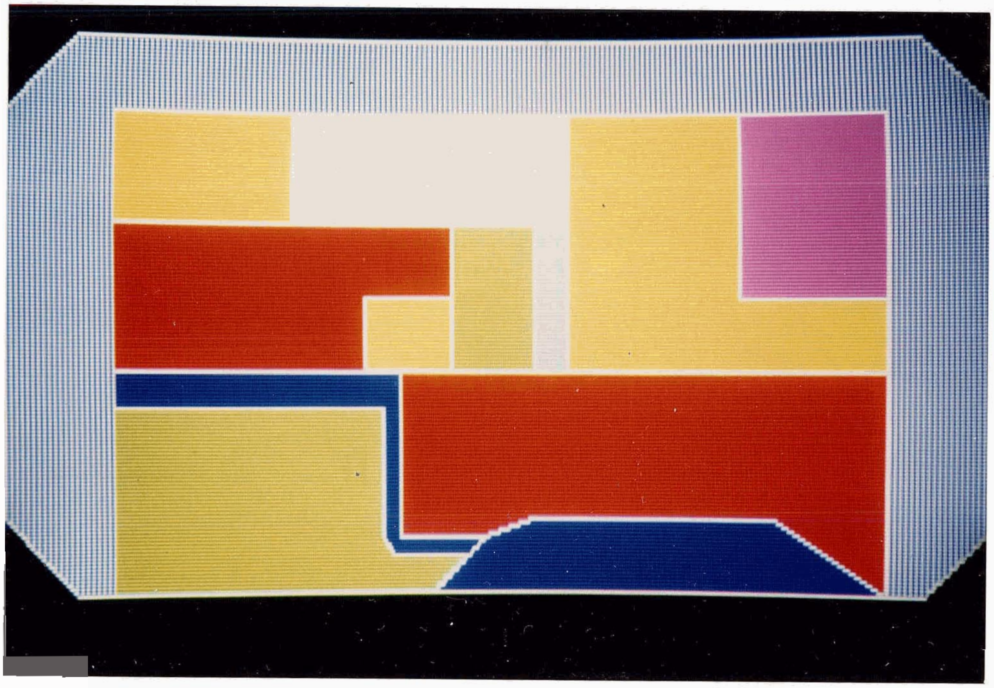


Figure 2.10

This picture represents the interior of a building. (Figure 2.11). Polygons colored the same were identified as belonging to the same feature code. A color mood of similar yet different tones has been preserved while the blue chosen for the centre polygons was respected. This highlights certain areas while ensuring that the rest of the map is clear enough to differentiate between areas. The corridor being a large polygon, it is a dull color.

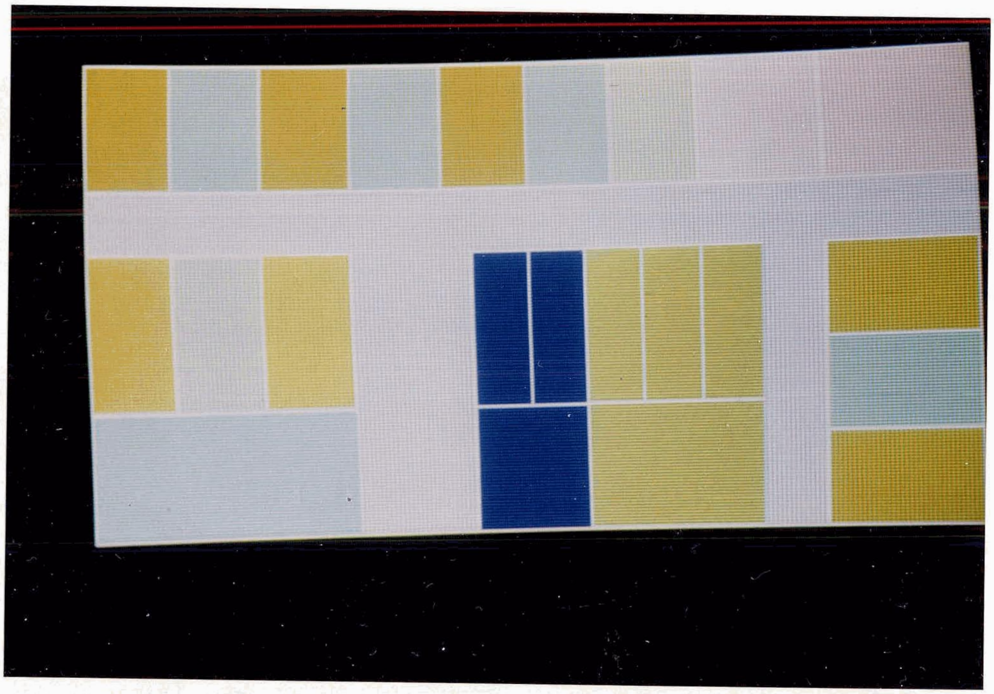


Figure 2.11

This is a picture of the same part of the United States which was used to illustrate the automatic mode. This one was produced under the shared or semi-automatic mode with a blue primary theme and a match criterion. (Figure 2.12). In order to use the shared option, a polygon was identified as a special feature. When comparing which states have the same color, it is interesting to note the difference with the automatic mode which also uses a specific starting hue for the search and a match criterion. This different distribution is due to the order in which the features are considered and the polygons are looked at, resulting in maintaining a certain color mood while making the program appear "unpredictable" and therefore more "creative".

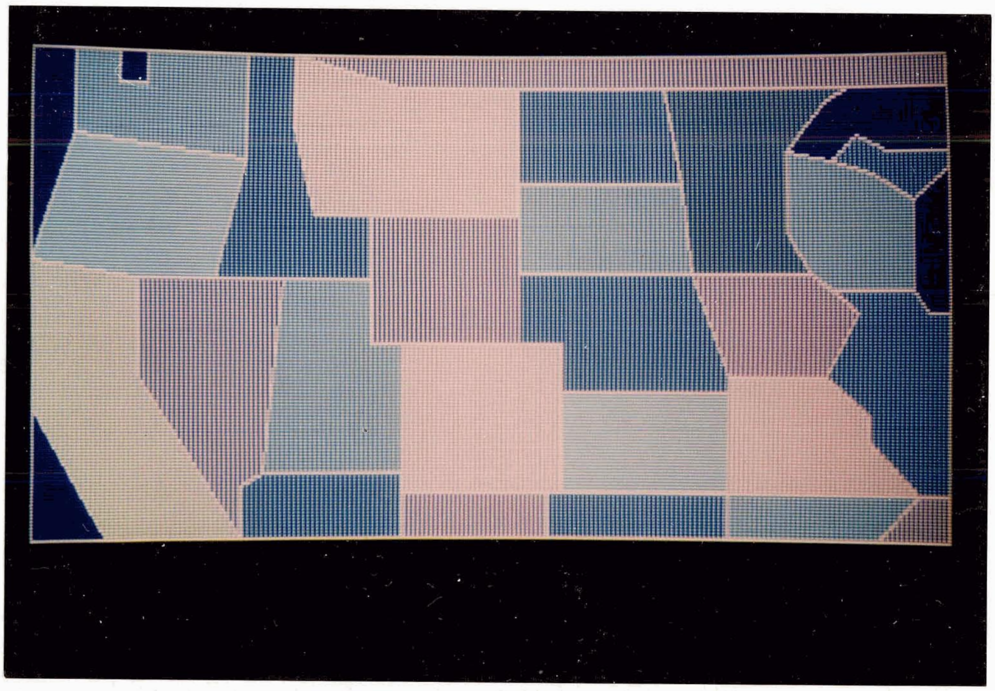


Figure 2.12

This last picture in shared mode illustrates what can happen when the contrast threshold is set too high for the user's choices. (Figure 2.13). In this picture, the coast was identified as a feature. Areas were also identified as green. In accordance with the color circle, (Figure 2.14), which appears below the map, magenta was chosen as an excellent contrast for green and a reasonable one for yellow. The middle of of the range of values, pure white, was considered acceptable either as a match or as a contrast. However, once some polygons have been colored, the program is "stuck". Polygons twenty-two and twenty-nine for example, border both magenta and green; no color can properly constrast both colors, which are opposite in the color circle, simultaneously. In addition, for polygon twenty-two, the color chosen should also be a reasonable contrast (under threshold) to yellow, and for polygon twenty-nine, to white. Polygon six borders many areas and contrasting each one with the latter contrasting each other is difficult. It is easier to match colors since, in accordance with the color circle, the two adjacent colors to a color are a perfect match but only one opposite color is a perfect contrast. As the other remaining two colors can go either way, in accordance with the "warm" and "cool" color schemes, a mid-line threshold tends to favor matching colors. The contrast threshold may need to be set higher. For further details, consult the "Color classification and analysis" section in this chapter. Obviously, the shape of the map, the topological relationships of the polygons and the order in which they are considered, also have a large role to play in the appropriateness of the choices made. Increased backtracking may eliminate some "deadlock" choices but at the expense of increased time and complexity and possibly reduced usefulness; as it is, the polygons are left uncolored and the program returns to the user with a message to try changing the thresholds. The user can then try again with other color choices or other thresholds.

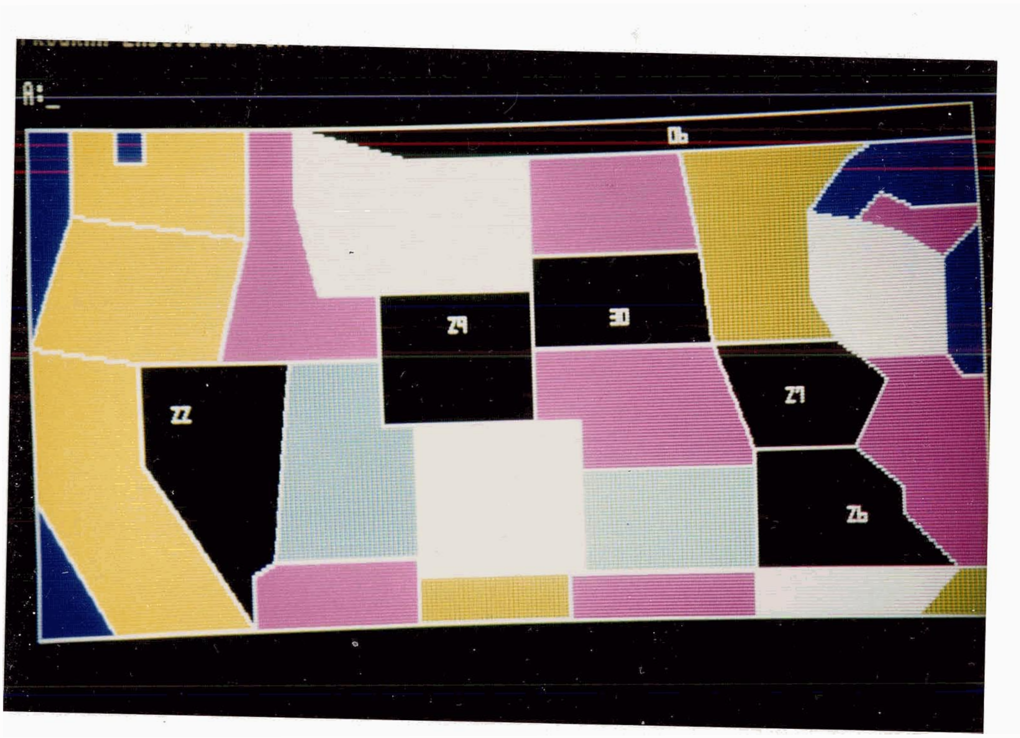
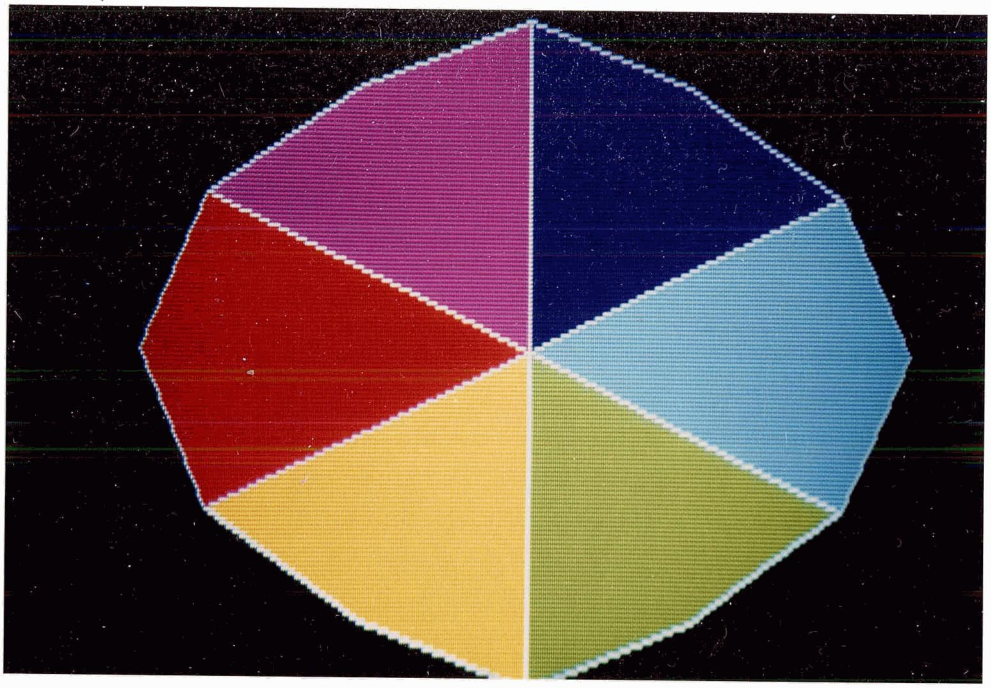
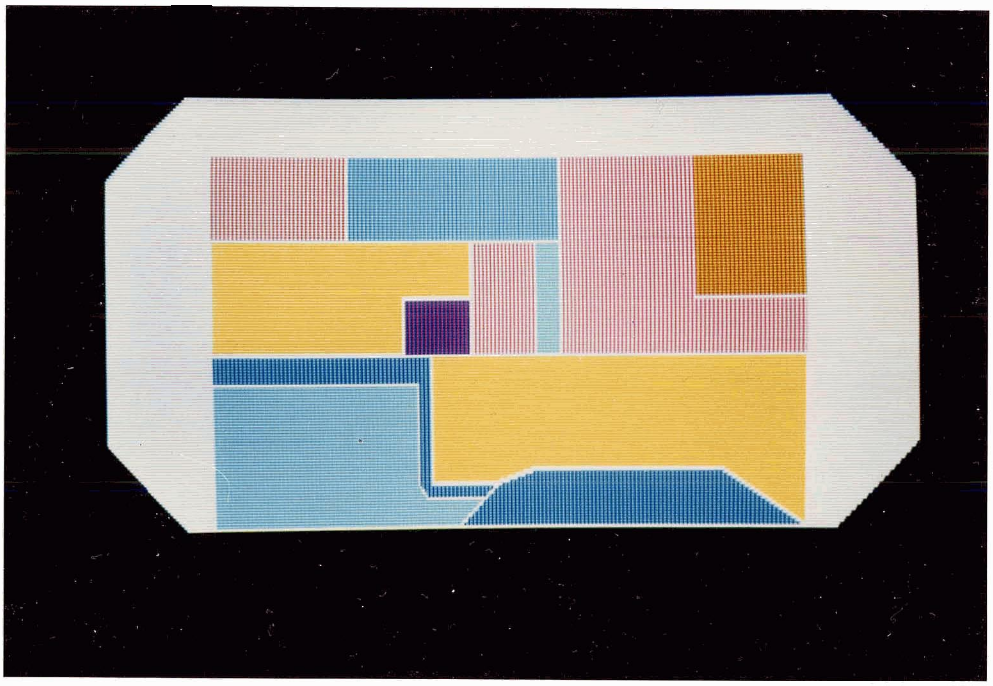


Figure 2.13

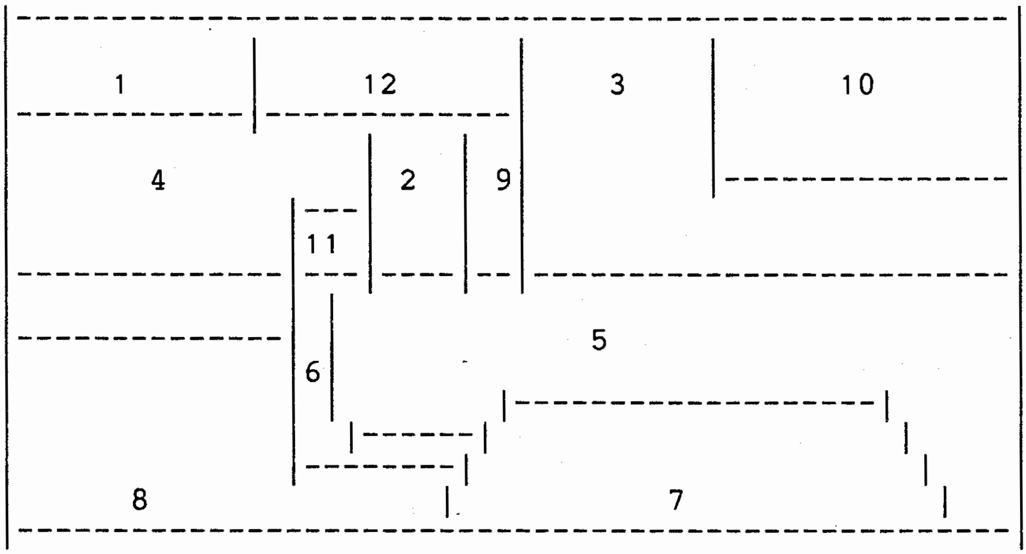




Figures 2.14 and 2.17

In the manual mode, the user dictates the colors to be used subject to several verifications made by the system. The following excerpts of the dialogue used to produce the photograph of the zone map (Figure 2.17), illustrate some of those system checks. "Errors" in judgement were introduced for this purpose.

At its original state the polygons were numbered as follows:



The surrounding background polygon is 13

Figure 2.15 Graph of uncolored zone map.

```
Dialogue excerpts:

areas 6-7 were identified as: water
areas 1-2-3 were identified as: industrial
areas 4-5 were identified as: residential
areas 8-9 were identified as: undeveloped
areas 10-11-12 were not assigned a code
area 13 was identified as the background

attempting to color 6 with 666(yellow) produced:
 water is usually blue. you have chosen   sample
 do you wish to change it(y/n)?           of 666

after having colored 3 with 227, attempting to
color 10 with 227 produced:
 you have already chosen this color for industrial
 do you wish to change it(y/n)?

attempting to color 1 with 222 produced:
 you have already chosen color 227 for industrial
 reenter the color you wish to use for area 1: (227)
 do you wish to use this color from now on
 for residential (y/n)?

choosing a blue (such as 113) for 11 produced:
 this color does not go well with its neighbors
 do you wish to change it(y/n)?

the background had to meet the dull color criterion
```

Figure 2.16 Excerpts of dialogue under manual mode.

The following design was also produced on manual mode. It gives an example of the variety of possible choices and can be used to illustrate the simultaneous contrast phenomenon described in the first chapter. (Figure 2.18). It is interesting to note how the centre squares (6 left and 4 right) appear when made adjacent to various colors. This design also shows that the program can be used to color not only maps but any other shape. The main difference is that some of the checking mechanisms for natural features would not serve any useful purpose and could be replaced by other rules more pertinent to the area of application.

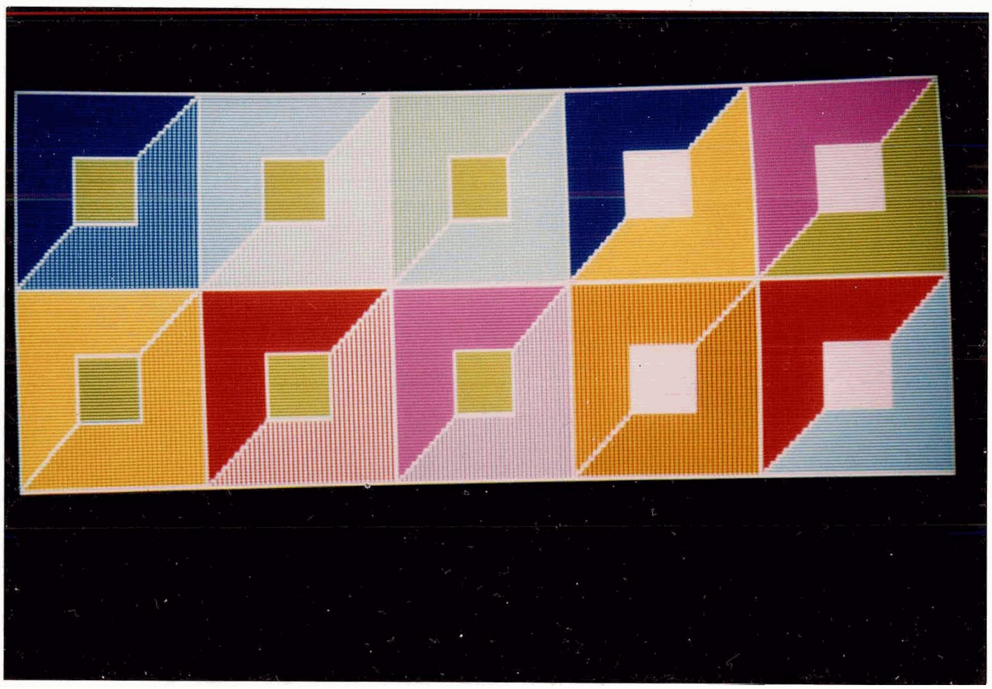
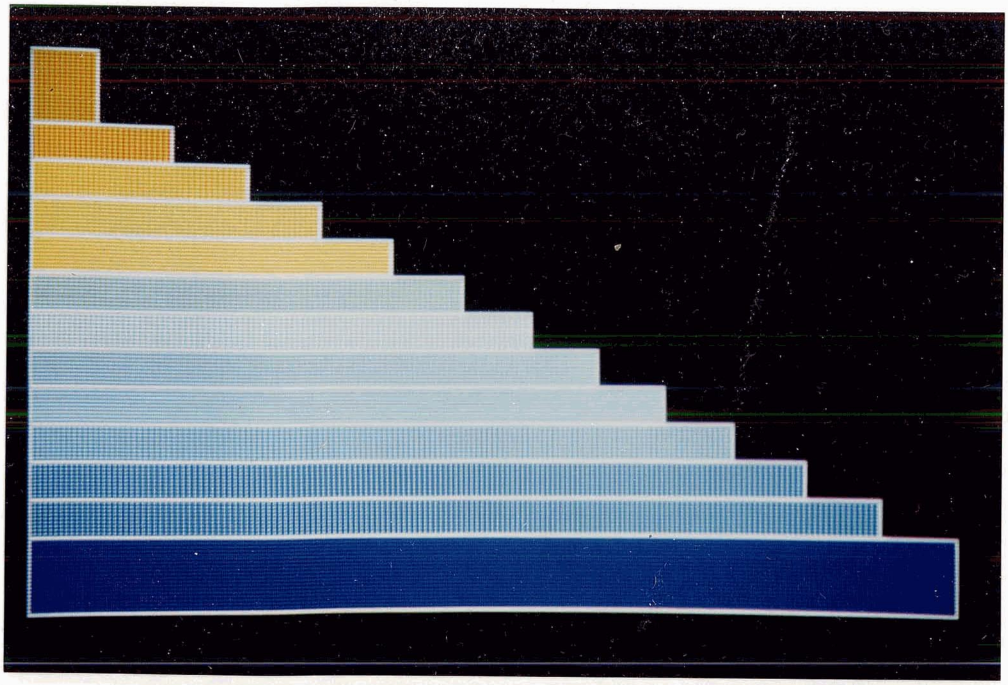


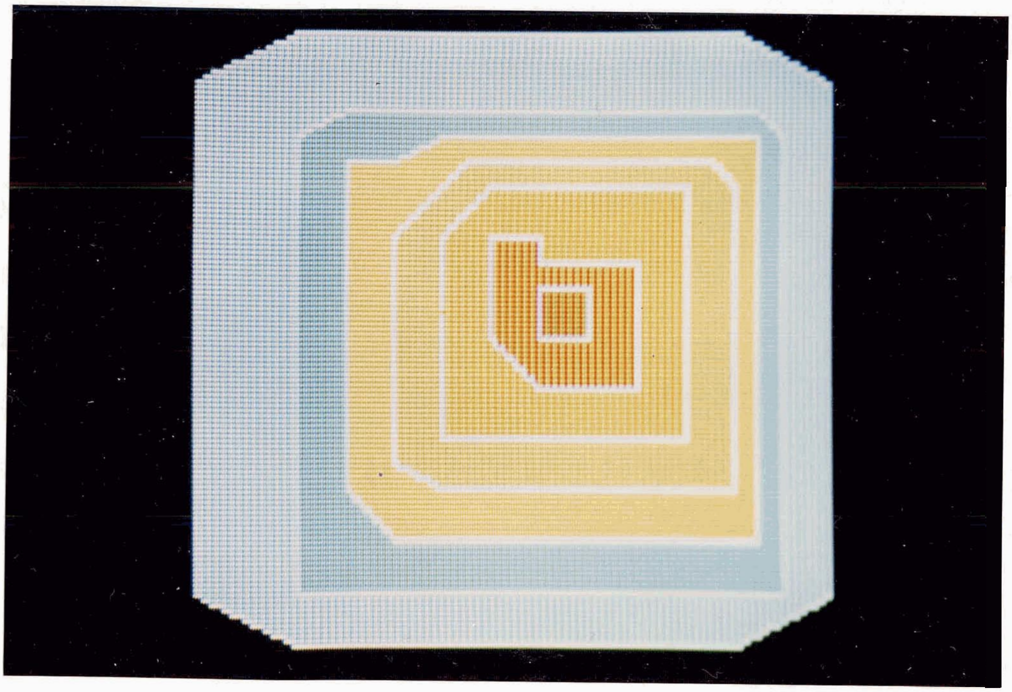
Figure 2.18

The last option, contour map coloring, uses a pre-defined set of colors read in as data. Thirteen elevation/depth steps are available using this option, of which the first seven are underwater steps and the last six, land steps.

The first picture of the following page (Figure 2.19) represents the thirteen steps from bottom to top, giving a series of blues and cyans getting progressively lighter for underwater steps, the first land step being light green; the land scale then becomes yellow and brownish as the altitude increases. This scale is based on the Imhof example mentioned in the appendix on Color and Cartography. The limited amount of brown colors produced by the color dithering process on the implementation computer forced this shift to yellowish/orangy tones to compensate. Also, because of the particular requirements for the use of white discussed with the coloring algorithm, no colors dithered with white have been included, giving one additional level of turquoise to Imhof's blue. Having chosen this option, the user/computer interaction consists of the program listing the depth/altitude values for each step, listing each polygon and asking the user to identify which step it belongs to.

The next two pictures represent the same contour map using first land steps and the first water step (7-13), then underwater steps (1-7). The centre polygon is the highest/lowest and gradation is progressive. (Figures 2.20 and 2.21).





Figures 2.19 and 2.20

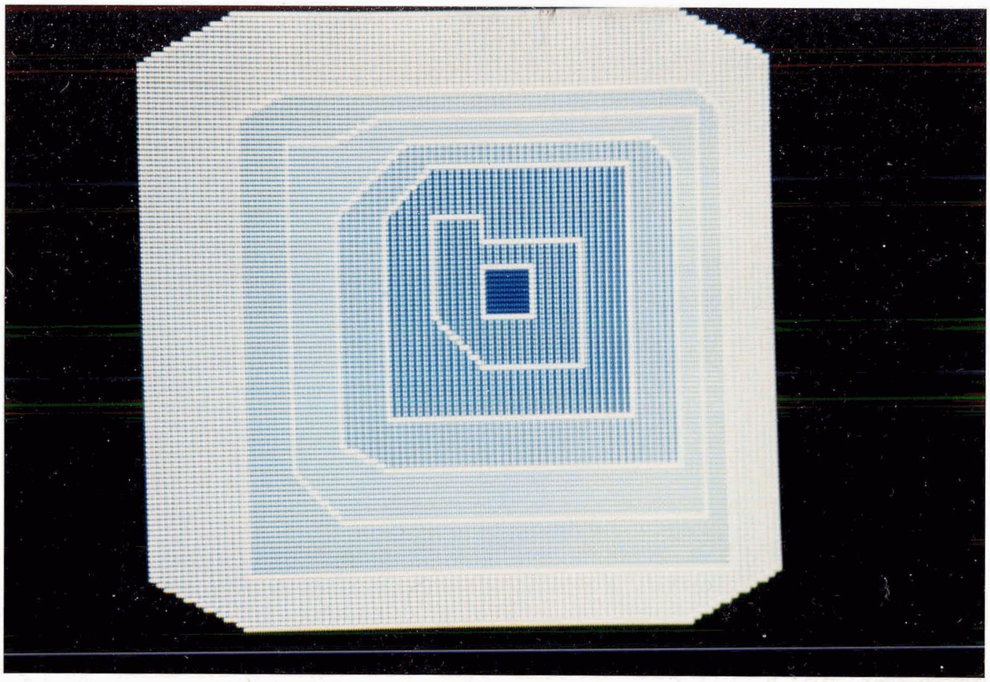


Figure 2.21

Many more tests were made with this program and it is impossible to show all the possible results in this document. The above is a representative sample of what can be accomplished. Given a digitizer, a program to smooth out lines, and someone to type in the coordinate data, many other more complex maps could be colored with the same rules. In such a case, however, the long form of the coloring algorithm should be used (i.e. without the speed-up modification) to accommodate every design and every type of object.

D. CONCLUSION

To the artist, color is more than a tool of the trade. Color is art. Vivid and subdued, dark and brillant, matching or contrasting, all is harmony, effect, message, sensation and aesthetics. To the scientist, color is one of the most fundamental elements of the physical laws of nature. Are not all colors derived from the primaries which are, in turn, the colors of the spectrum which no prism can decompose any further? The miracle of the light reflected in the retina of the eye and transmitted to the brain to produce a sensation of the colorful world which surround us, the extremely complicated interaction of the functions of the organs of vision, still amaze the biologists and physiologists. As for psychologists, the symbolism and message of color, the subjective visual response, the conscious or subconscious direction of interest, memory effect, perception variations and reactions produced by the use of color, are all areas of extreme interest which are still the subject of further studies. The cartographer does not need a detailed knowledge of all these disciplines. To him, what is important are the spatial properties of color, the effects produced by the use of color on a two-dimensional picture of the world, how a message is conveyed and how well it is received. However, to produce an accurate, attractive and meaningful map, some part of all the above disciplines come into play, and the cartographer, consciously or not, must coordinate them. In addition, he must emphasize spatial and topological relationships between the data he wishes to portray. In computer terms, the representation of color must be made to relate to the cartographic data structures which it will support.

In designing the system described above, Graphic Design with Color, the author has attempted to convey some of the ordinary intuition that people have about color to a computer system. This was done through the use of a knowledge base, a tool of artificial intelligence, which allows the computer to assess situations, compute several alternatives, adapt itself to the wishes of its user, make decisions, double check the validity of these decisions according to a variety of criteria, change its mind if required and come up with a product which, in the eyes of most people, would have required some intelligence on the part of the system. A product which is not only attractive, but also conveys some of the messages that a cartographer would have wished to convey, if he had colored the map by hand. This knowledge base was composed of data structures, the main ones being the color structures and the polygon structure, which, when filled in, contained knowledge on the harmony of colors, on the spatial and topological relationships of each polygon to other polygons on the map, as well as on the map data given to it by the user. Interactive dialogue with the user allowed the system to incorporate further information about relationships between cartographic features and colors to be used for them, and to find out desired color themes, feature designations, and other preferences of the user. Through the use of procedures designed to make and validate choices through the adherence to many criteria related to color graphic composition and cartographic conventions, the system was able to warn the user of possible bad choices, suggest alternatives, and by itself color a map which meets the cartographic and artistic requirements of a good map as well as the wishes of the user.

By bringing all of the above-mentioned disciplines under the same roof, relating color to topological cartographic data structures, and displaying intelligence through its knowledge base, this system is a pioneer. It is a seed which, with proper care and feeding, could blossom into a full-grown fully-fledged expert-system, extremely useful in countless situations. What are the present trends, in computing science, psychology and cartography, from which this system could benefit?

First of all, the field of graphics is a rapidly advancing branch of computing science. Already, many algorithms have been found, if not all implemented, in advanced computational geometry; graphic packages, now on the market, seem to be able to do beautiful designs and most of the functions a graphic designer would wish. The problem is, as with most packaged programs, that they are designed for one specific purpose and are not, in algorithms and coding details, available for perusal. The present system, although it makes use of graphics, emphasizes the color and knowledge base aspects of the system. It could benefit from further graphic capabilities, such as, for example, the ability to color ,easily and rapidly, a polygon of any shape and any level of nesting with all the colors it is possible to create. Also, if dithering did not have to be used to create colors, but all colors were available directly to every pixel, some of the requirements of the coloring procedure could be simplified since reserving black and one other color for special purposes, i.e. empty and borders, would only remove two colors from the chart. Increased speed in the actual coloring process is also a consideration which assembly language implementation, parallel processing or other criteria could help to bring about. Finally, another element which has been implemented internally only through some bactracking is the ability to color again a polygon which has already been colored on the screen. The "intelligence" required for such a procedure is simple: what has been done once can be done again. However, this would require reversing the present coloring and number drawing procedures using black as a color and anything else as empty. White would still be a special case since borders should not be erased if we are not to increase considerably the storage required for the knowledge base. Increased backtracking capababilities would however also introduce the problem of using a larger time span and appropriately selecting stopping criteria in accordance with the problem given.

Another branch of artificial intelligence which is at the forefront of exploration right now is natural language understanding. From Winston's cube world to Wilk's translating machine, parsers have been developed to understand a limited vocabulary in medicine, geology, business etc.. This system already allows the user to choose his own names for feature identification; the rest of the dialogue, mostly prompted by questions, looks for a certain structure suggested by the promptings. The requirement to enter data in specific columns has certainly been obliterated and the dialogue displayed by the computer can be termed "user friendly". As a computing language, Pascal is accurate, fast and proved well suited for the organization of the data into the required structures, the modular block organization of the procedures, the computations in general and this particular application; it also has the added benefit of being a language commonly used on microcomputers; however, lacking string processing facilities, it hindered a certain flexibility and higher degree of "naturalness" of interface possible with other languages which are more string oriented. To allow a conversation to take place at the level of every day life, a natural language parser, complete with a dictionary of definitions and synonyms, could have much to add to any expert-system.

Another emerging field of artificial intelligence, computer vision, is concerned with the vision and recognition of objects by the computer itself. Although much research has already taken place in the basic questions of edge detection, shadows and light, and relationships of an object to another, object recognition, color vision, and color recognition have not yet progressed to a widespread practical state. What progress would be achieved if the system could actually look at colored maps and comment on them, recognize and understand its own choices and advise the user as to the appearance of something just by "looking at it"? Seeing the whole picture and the relationships would also help it in establishing other possible topological relationships.

Numbers could also be increased: more colors, more rules, more cartographic conventions. What has been implemented so far are the basic rules of graphic and cartographic composition, applicable to most types of maps. Cartographers often produce specialized maps related to geology, meteorology, botany, all of which have specific color conventions. Adding these rules to the system as it is would not be difficult, just time consuming. Rules could also be added to broaden the sphere of influence of the system to deal with other areas: interior decoration, clothes, vehicles, etc.; in short, anything which requires harmony of colors, could benefit from this system. However, to do so, the knowledge base would have to be expanded, relationships and weight values revised and new rules and new procedures added. The system would have to learn the new fields through the eyes of various experts and designers.

Using Bertin's terminology, color could be combined with all the other retinal variables to produce effects only possible through such combinations. Combined with size for quantitative information or with texture for emphasis, the extended capabilities of the system would be applicable in many fields.

Last but not least, the greatest value of color lies in its psychological attraction and some of its greatest effectiveness in a learning environment. The possibilities of using such a system as a teaching tool are limitless. People suffering from color vision deficiencies would benefit from a system capable of teaching them, in full privacy, correct or acceptable color combinations. Many such persons have been unable to aspire to positions that require a good eye for color harmony; with an intelligent system to help them, mistakes could be found early, experimentation done without wasting materials, choices assessed and, most of all, the task could become a friendly and beneficial learning experience. The system could even be programmed to simulate their usual perception of the world, to see as they see as it were, and show them the difference between it and what is considered "normal" perception.

There is so much that can be done with computers and with color in any field of human endeavour. It seems that an intelligent computer system dedicated to graphic design with color has a very bright future indeed.

APPENDIX A

PROGRAM DOCUMENTATION

REMARKS

- Some variables declared in the main program, particularly for large "typed" structures, are used globally as inherited variables in procedures and functions.
- For this program, the valid x coordinates are 1..300 and the valid y coordinates are 60..210. The program modifies the x distortion by multiplying the value by two.
- The features which may be particular to the type of computer used are:
  - The functions PIXEL (used to set a pixel a certain color on the screen) and COLOUR (used to test and return the color of a pixel on the screen), were written in assembly language for the Zenith-100 computer and are Copyright (C) KEA Systems Ltd.,1984. Their usage has been authorized for educational use but the code is unavailable.
  - To keep the map frozen on the bottom of the screen and the dialogue on the six top lines, homing the cursor and blanking lines are necessary. This is done by: WRITE(CHR(27),'H'); BLANKS(N); WRITE(CHR(27),'H'); where CHR(27) is the escape code (i.e. ESC H) for homing the cursor and BLANKS is a procedure which blanks N lines.

DATA STRUCTURES, TYPES AND VARIABLES, DECLARED IN THE MAIN PROGRAM

CCT = Type containing the match/contrast values of each built-in color with every other built-in color (array [1..7,1..7] of integer).

COLORT = Record type containing : id ... 3 digits color id., nhue ... subscript of the next color within the hue or 99 for end or hue.

PT = Pointer to type POLY.

PAT = Array [1..30] of PT.

INTYPE = Pointer to type IMM.

LNTYPE = Pointer to type LARGE.

POLY = Record type containing : id ... polygon id., feat ... feature code, color ... color assigned to the polygon, size ...polygon area size, minx,maxx,miny,maxy ... array of two elements containing the value stated and its corresponding counterpart i.e. minimum x coordinate and corresponding y coordinate, anchpt ...coordinates of a point known to be inside the polygon, in ... pointer to immediate neighborhood list, ln ... pointer to larger neighborhood list.

IMM = Record type containing : id ... id of immediate neighbor, nexti ... pointer to next record.

LARGE = Record type containing : id ... id of larger neighbor, nextl ... pointer to next record.

FCT = Record type containing : id ... feature code id., name ... feature code name (up to 20 characters), thcol ... theme color wished for this feature (char), spcol ... specific color assigned to this feature.

SHORT = String of 20 characters.

MINI = One character.

I1,I2,O1 : Text files used for input of knowledge base data (I1), input of map data (I2), and available for output (O1).

CC : Color chart of type CCT.

COLOR : Array[1..49] of type COLORT.

CT : Array[1..7] that contains the first letter of each theme color.

PA : Array of pointers (type PAT) used to access polygons.

FC : Array[1..10] of feature code records (FCT).

DULL : Array[1..18] containing the subscripts of dull colors.

CONT : Array[0..13] containing the subscripts of contour colors.

MTRESH,CTRESH : Arrays of two elements containing the match and contrast thresholds for immediate and large neighborhoods.

NO_OF_POLY : Number of polygons read.

OVERS,UNDERS : Size limits for procedures to consider a polygon "large" or "small".

BACK : Contains the id. of the background polygon, if applicable.

SREPLY : String of type SHORT.

ANSWER,KEY : Characters.

NEUTRAL,BLUE,RED,MAGENTA,GREEN,YELLOW,CYAN : Starting "pointers" to other colors in this hue.

I,J,K,ANS,COUNT,DISPLAY : Integers.

FUNCTIONS AND PROCEDURES

FUNCTION PIXEL (VAR X,Y,Z:INTEGER):INTEGER;EXTERN;

FUNCTION COLOUR (VAR X,Y,Z:INTEGER):INTEGER;EXTERN;

As explained in the remarks, these two function are assembly language functions which set and test individual pixels. PIXEL sets the pixel at X and Y to color Z, and COLOUR returns the color of the pixel at X and Y by modifying the Z value passed to it. The dummy assignment variable used to call these functions is said to contain invalid data.

PIXEL ... CALLS: None.

IS CALLED BY: Draw, Drawfig, Dith, Show, Colorp.

COLOUR ... CALLS: None.

IS CALLED BY: Colorp.

PROCEDURE BLANKS(N:INTEGER);

This procedure sets N blanks lines on the screen. It is used to keep the map at the bottom of the screen while the dialogue occurs on top. The maximum value of N should be 6 or it may interfere with the map.

CALLS: None.

IS CALLED BY: Main, System, Dith, Tresh, Colval, Struct, Contour, Manual, Help, Show, Startcol, Sugg, Auto, Shared, Level1.

MAIN PROGRAM.

The main program calls on appropriate procedures to do the set up and the coloring. It directly asks some input from the user and evaluates the coloring process at the end.

- Initialize input and output files for reading and writing.
- Set up the color, feature code and polygon structures and draw the map on the screen (SETUCOL; SETUPFC; SETUPPOL).
- Ask the user to identify water areas, vegetation areas and any other special areas. Update the feature code structure accordingly.
- Ask the user to identify the background if any.
- Offer coloring options to the user and call the appropriate procedure.
  - 1. I color the map (AUTO).
  - 2. You color the map (MANUAL).
  - 3. We color the map (SHARED).
  - 4. Contour map coloring (CONTOUR).
  - 5. Quit.
  - 6. System procedures (SYSTEM). This option allows a return to the main options menu after each call.
- After the map has been colored, check if any uncolored polygon remains. If so, suggest to the user to change the threshold values since they were probably too high (match) or too low (contrast).

_ Display end of program message.

CALLS: Blanks, Setupfc, Setupcol, Setuppol, System, Contour, Manual, Auto, Shared.

IS CALLED BY: None.

PROCEDURE SETUPFC;

This procedure initializes the feature code structure. Default values are: id.:0, name:'none', thcol:'N', spcol:999.

CALLS: None.

IS CALLED BY: Main.

PROCEDURE SETUPCOL;

This procedure reads in match/contrast weight values, dull colors and contour colors. It also sets the chain of hues, the match and contrast thresholds, and the letters identifying the color themes. This data should be contained in the file mapped to I1.

_ Read in match/contrast values (CC).

_ Read in dull color subscripts (DULL).

_ Read in contour color subscripts (CONT).

_ For each element of the color structure read in the subscript value (from 1 to 49) of the next color in the applicable hue.

_ Read the subscripts of the top hue "pointers": neutral, blue, red, magenta, green, cyan and yellow.

_ Read the match and contrast thresholds for immediate and larger neighborhoods.

_ Read the first letter of each color theme (CT).

CALLS: None.

IS CALLED BY: Main.

PROCEDURE SETUPPOL;

This procedure sets up the overall polygon structure. Presently, the program allows up to 30 polygons with up to 28 points each. The coordinates are contained in the file mapped to I2; they should be given in either clockwise or anticlockwise order and the end of the polygon is determined by the coordinates 0 0. These same values also determine the end of the series of polygons. Valid coordinates for X are 1..300 and for Y 60..210.

_ Read an X and a Y coordinate from the input file and while not end of file do:

- . While not end of polygon do:
  - - Step up the X coordinate by two to account for the larger number of horizontal pixels.
  - - Allocate a polygon structure.
  - - Initialize all variables in the structure:
    - * id: the order in which the polygon is read in.
    - * feat: 0
    - * minx[1],minx[2],miny[1],miny[2]: 999 ... large number.
    - * maxx[1],maxx[2],maxy[1],maxy[2]: 0 ... small number.

* anchpt[1],anchpt[2]: 0
* color: 999
* pointers to immediate and larger neighborhoods: NIL

- Establish minimum and maximum X and Y coordinates, i.e. if the value read is less than the present minimum or greater than the present maximum, use this value.
- Read the next X and Y coordinates.

. Wrap around the temporary POINTS array, i.e. copy last value to the first line and the first coordinates read to the last line, for subsequent sequential manipulations.

. Compute the size of the polygon using the formula: size = half of the absolute sum (i = 1 to number of points) of Yi * (X coordinate of i+1 minus the X coordinate of i-1); or, put another way:

```
where i=1 to number of points
size= size + (Yi * (X(i+1) - X(i-1)));
size= ABS(size) * 0.5;
```

. Draw the borders of the polygon on the screen by calling on the DRAW procedure with every pair of points to be linked.

. Draw the polygon's id. number by separating the digits and calling the DRAWFIG procedure with each digit to be drawn and the location of a point inside the polygon.

This point is obtained by the following method:

a) take three consecutive vertices

b) link the first and the third and take the middle of the linking edge

c) test if this point is in the polygon

d) if not, take the last two vertices and the following vertex and repeat b) and c) until the point is in the polygon.

In the case of a triangle, the point is pushed back inside the triangle by a few pixels in order to superimpose the raster matrix for the identification number.

- Read the first X and Y coordinates for the next polygon.

_ When all the polygons have been read and drawn, compute the neighborhoods.

- An immediate neighbor to polygon X is defined as one that has one or more overlapping X coordinate(s) and Y coordinate(s).
- A larger neighbor is one that falls within a given radius from the borders of polygon X.
- For additional details on the reasons for these algorithms, refer to chapter on Graphic Design with Color.

_ Compute the size boundaries, i.e. the definition of "large" and "small" polygons by calling on the procedure SBOUND.

CALLS: Sbound, Draw, Drawfig.

IS CALLED BY: Main.

PROCEDURE SBOUND(OVER,UNDER:REAL);

This procedure computes the average size of all the polygons and then sets the oversize bound to twice the average and the undersize bound to a quarter of the average size.

CALLS: None.

IS CALLED BY: Setuppol.

PROCEDURE DRAW(X1,Y1,X2,Y2:INTEGER);

This procedure draws a line between two points. If the X or Y coordinates of the two points are the same, the line is straight; otherwise, either the X (angle less than 45 degrees) or the Y (angle more than 45 degrees) is stepped up to produce a clearer picture.

- Set DX to X2-X1 and DY to Y2-Y1.
- If DX = 0 then for I:=0 to DX by the sign of DX set pixel (X1+I,Y1) to white.
- If DY = 0 then for I:=0 to DY by the sign of DY set pixel (X1,Y1+I) to white.
- If DY <= DX then compute M=DY/DX and B=Y1-(M*X1) and for I:=0 to DX by the sign of DX set pixel (X1+I,M*(X1+I)+B) to white.
- If DY > DX then compute M=DX/DY and B=X1-(M*Y1) and for I:=0 to DY by the sign of DY set pixel (M*(Y1+I)+B,Y1+I) to white.

CALLS: Pixel.

IS CALLED BY: Setuppol.

PROCEDURE DRAWFIG(F,X,Y,C:INTEGER);

This procedure draws the identification numbers inside the polygons.

- Set up the mask raster array (5 x 5 elements) and initialize it to zero.
- According to the number required (F), set the appropriate elements to ones.
- Superimpose the mask array to the location given (X and Y) and draw the figure by setting the '1' pixels to the color (C).

CALLS: Pixel.

IS CALLED BY: Setuppol, Colorp.

PROCEDURE SYSTEM;

This is the procedure which accesses the knowledge base and other program structures. Four choices are offered to the user: 1) Change thresholds (calls TRESH); 2) Change color values (calls COLVAL); 3) Print structures (calls STRUCT); 4) Show all available colors (calls DITH). The description of these procedures follow.

CALLS: Blanks, Dith, Tresh, Colval, Struct.

IS CALLED BY: Main.

PROCEDURE TRESH;

This procedure is used to reset immediate and large match and contrast thresholds interactively. It gives the present threshold values to the user, asks for new values and does the replacement.

CALLS: Blanks.

IS CALLED BY: System.

PROCEDURE COLVAL;

This procedure changes the color values interactively. The user has three choices: 1) Change dull colors 2) Change contour colors 3) Change match/contrast values between pure colors.

For options 1 and 2, the present color subscripts are displayed one by one and the user is asked for a replacement. Zero indicates no change. For option 3, the user is asked for the two "pure" (1 to 7) colors relation which should be changed and the new value (0 to 5). The change is made in the color chart table of match/contrast values. The function MC, which calculates actual values for dithered colors will use this changed relationship when called, ensuring appropriate propagation.

CALLS: Blanks.

IS CALLED BY: System.

PROCEDURE STRUCT;

This is an output procedure. It prints in an output file or on the printer (i.e. device mapped to O1), the contents of the following structures as requested by the user:

1. Color chart ... match/contrast values.
2. Color structure ... id. and next in hue.
3. Dull colors.
4. Contour colors.
5. Match and contrast thresholds.
6. Feature code structure ... id, name, theme color, specific color.
7. Polygon structure. For each polygon: id., feature code, size, color, minimum and maximum X and Y coordinates and their respective counterparts, anchor point coordinates, list of polygons in the immediate and the larger neighborhood.
8. Size limits over which or under which a polygon is considered "large" or "small".
9. The match/contrast value of each color with each dithered color with all other dithered colors.

Option nine is not an actual storage structure as the other options are but rather a virtual storage structure. It is the result of the match/contrast (MC) function being called in a double loop to generate those values. However, since the values are used as if they were stored, it was made possible to print them in this procedure.

CALLS: Blanks.

IS CALLED BY: System.

FUNCTION MC (CID1,CID2:INTEGER):INTEGER;

This function computes the match/contrast value of two given colors. If the two colors are identical, the function returns zero. Otherwise, the numbers are separated to get individual built-in color numbers (1 to 7), three for each dithered color, i.e. 334 is 3-3-4. It then computes the actual total value by summing up the weights, in accordance with the color chart (CC), for each digit of one color with each digit of the other color. For example, if the weight between 2 and 4 is 2, and between 2 and 3 is 5, then the match/contrast value between colors 222 and 334 will be 36.

CALLS: None.

IS CALLED BY: Chmatchi, Chmatchl, Chconi, Chconl.

FUNCTION CHDULL (C:INTEGER):BOOLEAN;

This function checks if the color passed to it is in the dull colors array.

CALLS: None.

IS CALLED BY: Level1, Level3.

FUNCTION CHHUE (C:INTEGER;H:MINI):BOOLEAN;

This function checks if a color is part of a certain hue. A variable FIRST is set to the first subscript of the hue specified by H. While not end of hue and not equal to C (i.e. not found), FIRST follows the "pointers" along the hue. If FIRST reaches the end of the hue, then the color has not been found and it returns false, otherwise it returns true.

CALLS: None.

IS CALLED BY: Startcol.

PROCEDURE PKKUE (H:MINI;VAR PLACE,COL:INTEGER);

This procedure picks a color within a hue. A variable FIRST is set to the first subscript of the hue specified by H. PLACE is where the procedure picked the color along the hue the last time it was called. It gets changed by reference during this procedure. The procedure returns to the last "place" set. If this is not the end of the hue, it then picks the color, increases the place by one and returns the color COL; otherwise, it sets PLACE to the default end-of-hue value and returns.

CALLS: None.

IS CALLED BY: Level1, Level3.

FUNCTION(S) CHMATCHI/CHMATCHL/CHCONI/CHCONL (P,PC:INTEGER):BOOLEAN;

These four functions are similar and are used to check the appropriateness of the proposed color passed to them (PC) with the neighborhoods of polygon (P). CHMATCHI checks for match with the immediate neighborhood. CHMATCHL checks for match with the larger neighborhood. CHCONI checks for contrast with the immediate neighborhood. CHCONL checks for contrast with the larger neighborhood.

- The functions first set a pointer to the appropriate neighborhood. For each polygon in that neighborhood do:
  - If the feature code of that polygon is either water, vegetation, or the same feature code as the one of the polygon passed to it, it returns true.
  - If the polygon is uncolored, it returns true.
  - Otherwise, it computes the match/contrast value of the two colors by calling on the function MC. The following criteria are used to set the function to true:
    - CHMATCHI .. value is greater than the immediate match threshold.
    - CHMATCHL .. value is greater than the larger match threshold.
    - CHCONI .. value is not zero and is less than the immediate contrast threshold.
    - CHCONL .. value is not zero and is less than the larger contrast threshold.
  - If these criteria are not met, the function returns false.

CHMATCHI, CHCONI ... CALLS: MC.

IS CALLED BY: Startcol, Level1, Level3.

CHCONI, CHCONL ..... CALLS: MC.

IS CALLED BY: Level1, Level3.

PROCEDURE ASSIGNC (CODE,COL:INTEGER);

This procedure goes through the polygons and checks if the feature code matches the code passed to it. If so, it assigns internally the color COL to that polygon. It also finds the feature code in question in the feature code structure and assigns the specific color to it.

CALLS: None.

IS CALLED BY: Shared, Level1, Level2.

PROCEDURE CONTOUR;

This is one of the coloring options. It uses only the contour colors.

- The program identifies the 13 steps (6 land and 7 water) used.
- For each polygon listed by the program, the user is asked to identify the appropriate step. The steps remain displayed while this takes place. The step identified by the user is kept in an array (C), the subscript of which (J) is the polygon number.
- When all areas have been identified, the program colors the polygons in accordance with the step indicated and the appropriate contour color (COLOR[CONT[C[J]]].ID).

CALLS: Blanks, Colorp.

IS CALLED BY: Main.

PROCEDURE MANUAL;

This is one of the coloring options. The user dictates the color to be used for each polygon. The procedure first gives a list of the commands available, reads the user's command, and calls the appropriate procedure. It repeats the process until the command is "Q"(quit).

CALLS: Blanks, Dith, Help, Show, Startcol, Sugg.

IS CALLED BY: Main.

PROCEDURE HELP;

This procedure is called from the MANUAL procedure. It lists the commands available: H : To see these commands. D : To see all available colors. Q : To quit the program. S : To see a sample of a particular color. C : To color a polygon. O : To get my opinion.

CALLS: Blanks.

IS CALLED BY: Manual.

PROCEDURE SHOW (REP:INTEGER);

This procedure shows a sample of a color on the top right hand side of the screen. When called from MANUAL, it asks the user for the color id. number (it first receives a dummy value as a parameter). When called from the automatic component of other procedures, it skips this part and uses the color passed to it by the calling procedure.

CALLS: Pixel, Blanks.

IS CALLED BY: Manual, Startcol.

PROCEDURE STARTCOL;

This procedure starts the coloring process from the MANUAL procedure and calls on COLORP to do the actual coloring on the screen.

- Ask user for the id. of the polygon to be colored and the color to be used.
- If this polygon has a special feature code (other than default) do:
  - . If the feature code is 1, i.e. water, check if the color is part of the blue hue. If not, warn the user, and ask him if he wishes to change the color. If yes, start again.
  - . Do the same for vegetation.
  - . If the feature code structure contains a specific color for this code, it means that a polygon with this feature code has already been colored. If the new color is not the same, then tell the user which color has been used for that feature and ask the user if he wishes to keep the same color for this polygon and other polygons with this feature code from then on.
- Check if the proposed color has already been used for a special feature different from the polygon's feature code. If so, point it out to the user and ask if he wishes to change it.
- Check if the proposed color would conflict (not a good match or a good contrast) with its immediate neighbors. If so, ask

the user if he wishes to change it.

- At this point the proposed color has passed every test or the user has decided to use it anyway. If the polygon has a special feature code and the specific color of the feature code structure has not been set or has been changed, then set it to the color used.
- Set the color variable of the polygon structure to the color finally decided upon.
- Color the polygon by calling COLORP.

CALLS: Blanks, Chhue, Chmatchi, Chconi, Show, Colorp.

IS CALLED BY: Manual.

PROCEDURE SUGG;

This procedure is used to give coloring suggestions to the user. For each hue it gives the best match and contrast hues. The scheme presented is similar to the one used in the knowledge base, i.e. adjacent colors in the color circle are a good match and opposite colors are a good contrast. It simply displays the combinations on the screen. It is called from the MANUAL procedure and from LEVEL1 (out of the SHARED option).

CALLS: Blanks.

IS CALLED BY: Manual, Level1.

PROCEDURE AUTO;

This is one of the coloring options. The program chooses all the colors by itself but it does give the user the chance to pick a starting hue for the search. It then calls on other procedures (LEVEL2, LEVEL3) to proceed with tests and assignment of colors. Finally, it goes through the polygon structure,writes the color chosen for each polygon in an output file and calls on COLORP to color the polygons with the assigned colors.

CALLS: Blanks, Level2, Level3, Colorp.

IS CALLED BY: Main.

PROCEDURE SHARED;

This is one of the coloring options.

- If no special feature code has been identified by the user, switch to the automatic procedure (AUTO) , otherwise do the following.
- Display the six color themes available (blue, cyan, green, yellow, red, magenta) and ask the user to choose a primary theme or a specific color from the chart.
- If there were at least two user identified special features, then ask the user if he wishes a secondary theme or not, and if so, to identify it.
- List the user identified features and ask the user for which feature he wishes to use the primary theme (or color) and for which feature he wishes to use the secondary theme (if applicable).
- Ask the user if he wishes the rest of the map to emphasize

matching colors or contrasting colors.

- If a specific color was chosen as the primary theme choice, assign that color to the polygons having the selected feature code and update the feature code structure accordingly (ASSIGNC).
- If a primary and/or secondary theme(s) was(were) chosen, call LEVEL1 to handle it(them) with some pertinent information: theme color, feature code number, whether match or contrast is requested etc..
- Pick a starting color to pass to other procedures: if a primary theme was selected and a match is desired, then pick this hue; for contrast, start with the third hue from the one chosen. This ensures a better chance of success from the start.
- Call LEVEL2 to handle polygons with feature codes for which the hue is picked by the system.
- Call LEVEL3 to handle polygons with no feature code.
- Write the color chosen for each polygon in an output file and call on COLORP to color all the polygons.

CALLS: Blanks, Assignc, Level1, Level2, Level3, Colorp.

IS CALLED BY: Main, Level1.

PROCEDURE LEVEL1 (F,F1,F2,MT:INTEGER; N:MINI; FLAG:BOOLEAN);

This procedure takes care of the polygons with feature codes for which a theme color has already been identified interactively by the user (when called from SHARED) or it can be called from

LEVEL2 for polygons with feature codes for which the system determines the color. The last part of the procedure behaves differently in these two cases i.e. if the theme was identified by the user it returns to the user otherwise the calling procedure takes care of making any necessary subsequent changes.

- While not end of hue and not found, pick a color ensuring that the color has not already been used for other feature codes. If such a color is found then OK is set to true else it is set to false.
- While OK and there are still polygons to test do:
  - If the polygon has the appropriate feature code then do:
    - Test if the color picked matches or contrasts (according to the value of MT) the immediate and larger neighborhood.
    - If the polygon is oversize or is the background then test if the color picked is a dull color and if it is undersize ensure that the color is not dull.
  - If all tests show that the color is ok then go to the next polygon.
- If still OK, it means that the loop has ended because all the appropriate polygons have been checked. Set FOUND to true else FOUND is false.
- If FOUND, then go through all the polygons again and assign the color picked to those which have the appropriate feature code. Update the feature code structure by including the specific color. Set the FLAG to true. If NOT FOUND, set the

FLAG to false.

- Check if the procedure was called with choices made by the user. If so, tell the user that it could not find a suitable color within that hue and ask if he wishes suggestions before starting again. In any case, the control is passed back to the beginning of the SHARED procedure to start again. If, on the other hand, it was called from LEVEL2, it returns to it and the calling procedure uses the state of the FLAG to proceed.

CALLS: Blanks, Chdull, Pkhue, Chmatchi, Chmatchl, Chconi, Chconl, Assignc, Sugg, Shared.

IS CALLED BY: Shared, Level2.

PROCEDURE LEVEL2 (F1,F2,MATCHCON,START:INTEGER);

This procedure takes care of polygons which have a special feature code but for which no theme color has been identified by the user. It is called from SHARED after the user identified colors have been taken care of directly by LEVEL1 and by AUTO which is the program's automatic coloring procedure.

- Choose the water and vegetation colors. If this procedure is called by SHARED, ensure that the color chosen is not the primary color or the secondary color chosen by the user. Assign the color chosen to the appropriate polygons.
- Go through the feature code structure. If a feature code has been identified and no specific color given to it yet call LEVEL1 passing it the required information. If the procedure

returns without having chosen a color (FLAG is false), go to the next hue in the color table. If the program has gone full circle back to the starting hue then give a last try to the neutral colors. If still not chosen, the polygon(s) remain(s) uncolored and the main program will handle it.

CALLS: Assignc, Level1.

IS CALLED BY: Auto, Shared.

PROCEDURE LEVEL3 (START,M:INTEGER);

This procedure handles polygons with no feature code.

- Go through all the polygons and do:
  - If the polygon has no assigned color and some colors remain do:
    - While not end of hue and not found do:
      - Pick a color starting at the starting hue and ensure that this color has not already been used for a special feature code.
      - Check for match or contrast (M) of the color picked with the immediate and larger neighborhood of the polygon to color.
      - Check the size and background criteria, i.e. if the polygon is large or background the color should be dull and if it is small it should not be dull.
      - If all tests are ok then assign this color to the polygon and the color has been found.

- If the color has not been found within the hue then pick the next hue. When all hues of the color circle have been tried, neutral colors are tried. If no color is appropriate, the polygon remains uncolored and the main program handles it.

. Go to the next polygon.

CALLS: Chdull, Pkhue, Chmatchi, Chmatchl, Chconi, Chconl.

IS CALLED BY: Auto, Shared.

PROCEDURE COLORP (PN,CN:INTEGER);

This procedure actually colors the polygon (PN) on the screen with the dithered color (CN).

_ Separate the three digit color identification number into individual built-in colors and store the digits into an array (C) of three elements.

_ Reset the polygon identification number to black.

_ If one or more of the three digits of the color is 7 (white) do:

  . Reset it to the other color or to blue if all white.

  . Find out which columns of the polygon should be white, which is possible because the dithering occurs in a regular pattern. Create a linked list with the column number and two spaces for the minimum and maximum Y coordinates of the polygon for that particular column. The latter are originally given default values.

_ Set one pixel of the polygon, given by "anchpt", to its appropriate color.

The algorithm followed for the coloring is a modified propagation algorithm. The basic algorithm is as follows:

- Set XSTART and YSTART to the X and Y coordinates of the first point minus 1. Correspondingly, add 1 for XEND and YEND.

  Repeat

  For X = XSTART to XEND

  For Y = YSTART to YEND

  If the pixel(x,y) is empty (black), and if it is next to a filled pixel of the colors used in the polygon then color that pixel.

  end end;

  Decrease XSTART and YSTART by 1 and increase XEND and YEND by 1.

  Until no more changes.

- The first modification is due to the dithering. Dithering is computed with the formula: K:=(X-((X DIV 3)*3))+1 and C[K] is the color used.
- The second modification is when K equals a digit which should have been white but was reset at the beginning to avoid confusion with the borders. The Y coordinate is compared to the minimum and maximum Y coordinates in the linked list for the column in question and, if need be, these are adjusted.
- The third modification, which is present in one version

of the program, was done to increase the speed and avoid some useless testing for most regular or elongated polygons. The distance between the coordinates of the first pixel and the minimum X and Y coordinates of the polygon is computed at the beginning of the procedure. These distances are used when no more movement is needed in one direction. Then the normal testing loop carries on for that distance to ensure that the top left hand corner is colored and then the increment goes directly to the last row or column. Another version of this program does not have this modification, so it is slower, but it works for any kind of polygon.

- Finally, if there were some white pixels, go through the linked list and for all columns:
  - If pixel is not white, set it to white.
  - If it is, test if the next pixel below is white and if not, test the number of intersections with white pixels to the bottom of the polygon.
  - If odd, set the pixel to white and if even, skip to the next white pixel.

CALLS: Pixel, Colour, Drawfig.

IS CALLED BY: Contour, Startcol, Auto, Shared.

PROCEDURES AND FUNCTIONS: MAIN GROUPINGS

MAIN

- SETUPFC	- SYSTEM	- MANUAL	- AUTO	- SHARED	- CONTOUR
- SETUPCOL	dith	help	level2	level1	colorp
- SETUPPOL	tresh	show	level3	level2	
sbound	colval	startcol	colorp	level3	
draw	struct	sugg		colorp	
drawfig		dith		sugg	

Sub-Groups:

Level1	Level2	Level3	Startcol
.chdull	.assignc	.chdull	.chhue
.chhue	.level1	.chhue	.chmatchi
.pkhue		.pkhue	.chmatchl
.chmatchi		.chmatchi	.show
.chmatchl		.chmatchl	.colorp
.chconi		.chconi	
.chconl		.chconl	
.assignc		.assignc	
..sugg			
.shared			

Blanks is called by any interactive procedure.
Pixel is called by any procedure that set screen pixels.
Colour is called by colorp.
MC is called by Chmatchi/Chmatchl/Chconi/Chconl.

Figure A.1 Groupings of procedures and functions.

PROCEDURES AND FUNCTIONS: CROSS-REFERENCE

NO	PROCEDURE/FUNCTION	CALLS	IS CALLED BY
1	Pixel	-	9-10-28-36
2	Colour	-	36
3	Blanks	-	4-11-12-13 14-15-25-26 27-28-29 30-31-32-33
4	Main	3-5-6-7-11-25 26-31-32	
5	Setupfc	-	4
6	Setupcol	-	4
7	Setuppol	8-9-10	4
8	Sbound	-	7
9	Draw	1	7
10	Drawfig	1	7-36
11	System	3-12-13-14-15	4
12	Dith	1-3	11-26
13	Tresh	3	11
14	Colval	3	11
15	Struct	3	11
16	MC	-	20-21-22-23
17	Chdull	-	33-35
18	Chhue	-	29
19	Pkhue	-	33-35
20	Chmatchi	16	29-33-35
21	Chmatchl	16	33-35
22	Chconi	16	29-33-35
23	Chconl	16	33-35
24	Assignc	-	32-33-34-35
25	Contour	3-36	4
26	Manual	3-12-27-28-29 30	4
27	Help	3	26
28	Show	1-3	26-29
29	Startcol	3-18-20-22-28 36	26
30	Sugg	3	26-33
31	Auto	3-34-35-36	4
32	Shared	3-24-33-34-35 36	4-33
33	Level1	3-17-19-20-21-22 23-24-30-32	32-34
34	Level2	24-33	31-32
35	Level3	17-19-20-21-22 23-24	31-32
36	Colorp	1-2-10	25-29-31-32

Figure A.2 Calling list of procedures and functions.

	BLUE	RED	MAGENTA	GREEN	CYAN	YELLOW	WHITE
BLUE	1 5 BL 111 C	2 2 MA 112	3 5 MA 113	4 4 BL 114 C	5 5 BL 115 C	6 1 NE 116 D	7 3 BL 117 C
RED	8 2 MA 221	9 5 RE 222	10 5 MA 223	11 2 YE 224 C	12 1 NE 225	13 5 YE 226 C	14 3 RE 227 D
MAGENTA	15 5 NE 331	16 5 MA 332	17 5 MA 333	18 1 NE 334	19 4 MA 335	20 2 RE 336	21 3 MA 337 D
GREEN	22 4 NE 441	23 2 GR 442	24 1 NE 443 D	25 5 GR 444	26 5 GR 445 C	27 5 GR 446	28 3 GR 447 D
CYAN	29 5 BL 551 C	30 1 NE 552 D	31 4 NE 553	32 5 GR 554 D	33 5 CY 555 C	34 4 CY 556 D	35 3 CY 557 D
YELLOW	36 1 NE 661 D	37 5 YE 662 C	38 2 NE 663	39 5 YE 664 C	40 4 YE 665 D	41 5 YE 666	42 3 YE 667 D
WHITE	43 3 BL 771 C/D	44 3 NE 772 D	45 3 NE 773 D	46 3 NE 774 D	47 3 NE 775 D	48 3 YE 776 D	49 5 NE 777 C

Figure A.3 Color Chart.

APPENDIX B

ARTISTIC AND SCIENTIFIC VIEW OF COLOR

ART

To show that to the artist, color can be a fanciful flight of the imagination, one only has to look at the names given to some colors: Prussian Blue, Lucretia, Quick Silver etc.. Even by searching along the historical paths and concluding that Lucretia must be some kind of red, the name is at best imprecise. How does Prussian Blue compare to Royal Blue?

To help clarify matters for burgeoning artists, a number of naming systems have been developed, acting as a map into the labyrinth of colors. Some of the best known are the Munsell system and the Ostwald system, the latter not being very favored by artists because it is cumbersome to use.

Any discussion of these systems requires the use of terminology particular to the phenomenon of color. Hue, value and chroma are often referred to as the three "color coordinates". (Billmeyer and Saltzman, 1981). Color is often used as a synonym for hue, as the latter refers to the perception of red, blue, green, yellow, etc., which, in physics, is measured as wavelengths. Value or lightness refers to how close to white or black a color appears, differentiating between light and dark tints. Chroma refers to how much "color" a color contains, or how colorful it is; chromaticness, colorfulness, saturation or purity of tone are other names which bear only very fine distinctions with chroma and which are often used to described the properties of a color. Finally, brightness, a combination of lightness and saturation, is often considered as a color variable, differentiating between bright and dull colors. Intensity is the measure usually associated with the psychological concept of brightness, although a change in physical light intensity does not necessarily correspond to a similar change in perceived brightness by the viewer. Armed with these brief definitions, color classification systems can then be more readily understood.

The Munsell solid is an irregular form. W.C.Libby describes it best as follows:

> "Its structural core is a vertical value scale of nine regular gradations between black and white. At the level of each gradation, one must visualize a horizontal disc, and arranged about the center of this disc, ten hues in their natural order, regularly spaced. Further, each hue is radial and each, as it extends outward, is identically graded to express increments of intensity. When the radiuses - yellow for instance - lie in the same vertical plane, the solid is functional. In order to find the exact and unique position of any color, we simply take a "lift", as it were, to the correct value level, move laterally on the radial line of the correct hue, and continue until the correct intensity has been reached. It is apparent that if the final increment of intensity were the same for each hue at each value level, this solid would resemble an open cylinder or perhaps a revolving door, with white and black appearing as extensions of its axis at the top and bottom. But each hue reaches its greatest intensity at its natural value level, as we know, and all hues do not reach the same maximum intensity. This makes the solid highly irregular, particularly if we are talking about reflective surfaces rather than light, and the introduction of more intense colors into the solid would have the same effect." (Libby, 1974, p.19).

A stylized design patterned after the Munsell system is illustrated in color in Plate 1 (Libby, 1974). It looks basically as follows:

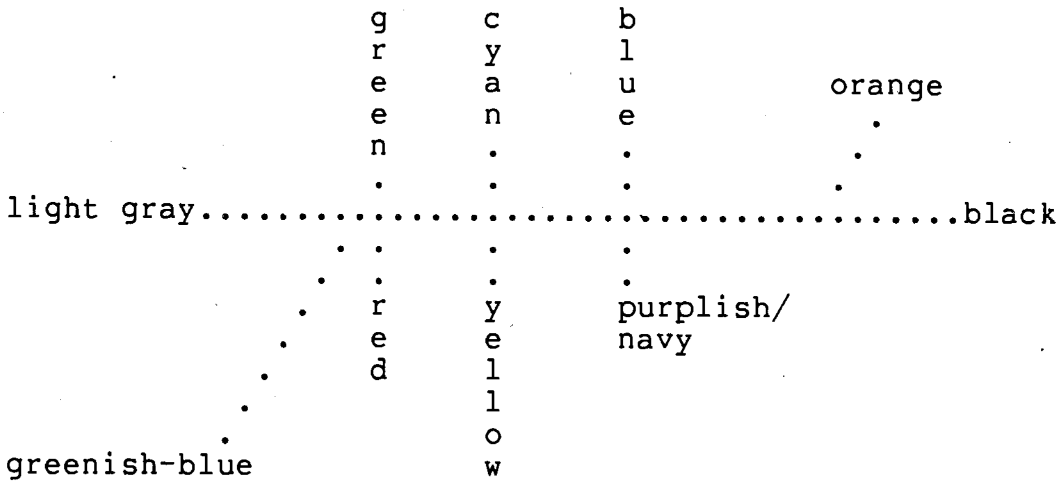


Figure B.1 Stylized Munsell graph.

The overlapping of the various colors demonstrates that every color is exclusively different from every other but related to it.

The Ostwald system, on the contrary, is regular and more technological in nature. (Libby, 1974). It is composed of twenty-four hues arranged symetrically about red, green, yellow and blue. Using double cones, six shades of gray connect the white apex of the upright cone to the black apex of the inverted cone. Since the height of each cone is half the diameter, the solid, when bisected vertically, reveals two equilateral triangles, each with a full color in its exterior angle, these full colors being complementary. Each triangle is then divided by lines parallel to the sides of the cones into twenty-eight rhombuses. Hues are named numerically, while values and scale are alphabetic. If we take, for example, the color 20-LE, we see that it is unique and reproducible. "L" means that this color can be obtained by spinning a disc that is 8.9% white and 91.1% black; for "E", we have 35% white and 65% black; "20" is the color green. So if two radial slices of color, a white one representing 8.9% of the surface and a black one representing 65% of the surface are added to the spinning disc of the particular hue which would comprise 26.1% of the surface, the color 20-LE is produced, which is in fact a green of median value and of moderate intensity. The entire solid contains 680 such unique colors. This is a far cry from the ambiguity of poetic names and could very well lead to computerization; it is certainly a difficult quick access to the names of colors (knowledge of history does not help) and to their dimensional relationships. The artist does not choose colors because they are dimensionally and systematically related, but rather because they convey his intent, are consistent and sympathetic, and are sensually satisfying as well. Any system which aims at helping artists must therefore be added to the artist's resources as a tool to understand relationships, and not as a carcan to make him abdicate the right to make aesthetic decisions.

Another system derived from Ostwald, but more recent and somewhat simpler, is the one developed by Hickethier from Becke's earlier attempts. (Imhof, 1982). Each chart is divided by ten horizontal blue strips running increasingly lighter from top to bottom and ten vertical red strips running increasingly lighter from right to left, resulting in a square of one hundred smaller squares. The lightening of color is done by percentage screening. Yellow is added from chart to chart resulting in a cube of one thousand small squares. These three primary colors, yellow, blue and red, combined with each other in various percentages, make all the possible colors ever required while maintaining a fairly simple designation process. Each color is designated by a number consisting of three digits. The first digit always refers to the yellow content, the second to the red and the third to the blue component. The numbers zero through nine correspond to the intensity of the three color components mixed. White is therefore "000", while numbers on the diagonal (222...888..etc.) are various shades of gray, "909" is a rich green and "999" is black.

On the more simple scale, we all learn very early that in art the primary colors are red, yellow and blue. These are the pigments from which all the other colors are made. Secondary colors are identified as orange (red-yellow), green (blue-yellow), and purple (blue-red). Colors are further classified by their "temperature", warm colors being yellow-orange-reddish and cool colors being of the blue-green variety. These are obviously not thermal facts but an emotional and associative response somewhat related to reality. "Somewhat" because, in many cases, logic does not extend to art and, psychologically, individual responses differ considerably. A flame goes through the stages of being first red, then orange, yellow, green, blue, violet and at its hottest, white. Yet the warmer colors are at the low scale of this heat progression. Maybe this is because most people experiencing the warmth of a fire will normally see the red-yellow glow, the hotter colors being reserved for something as unromantic as an acetylene torch. Warmer colors are more dominant, stimulating and spatially assertive. Some colors, such as green and purple may be considered to have an ambivalent temperature. It is usually determined by the other colors surrounding it, the influence of its neighbors.

There exists a visual phenomenon, called simultaneous contrast, which accentuates the contrast between adjacent colors by causing them to appear increasingly different in ways in which they are already different. If a light and intense yellow is placed next to a darker, less intense yellow-red, the first will become lighter, yellow-green and more intense. This phenomenon is continuous, inescapable. However, there is one exception which confirms the rule. For complementary colors (i.e. orange and blue), their intensities increase simultaneously when they are brought together. This can be used very artfully to create dramatic color effects, such as those symbolizing Christmas (red and green) and Easter (yellow and purple). Complementary colors, already as different as possible in hue, can become even more different by changing in value and intensity. For example, an intense red, placed next to a less intense blue-green, becomes more intense while causing the blue-green to do the same. An excellent example of simultaneous contrast can be found in Plate 4 (Libby, 1974).

To the professional artist and to the artist in all of us, all these phenomena become so much ingrained that they become intuitive. Only natural objects can be predictable in color. Yet in any work, color harmonies can be created to achieve marvellous effects. As Joseph-Emile Muller said, when commenting on the paintings of Paul Klee, a modern artist:

> "However, it is not only through the medium of form that this marvellous quality is created - it may be produced equally often by colour. And not only when the colour is subdued, with delicate variations of tone (as is most frequently the case), but also when it is laid on in a broad wash, when it is, if not brillant, at least extremely pure, fresh and insistent. Even then, it still has a kind of secret character, always some suggestion of a hidden world that lies beyond." (Muller, 1957, p.6).

So as long as the important structural principles of design with color are respected, that is unity, clarity and interest, the artist can make use of the full kaleidoscope of colors created by a judicious mix of the primary, secondary and complementary colors to impart a message understood by the heart of both his admirers and his staunchest critics.

<u>SCIENCE</u>

In science, color is studied in disciplines such as biology, chemistry and physics and certainly in psychology. As biology and psychology are very related and consider not only the objective but the subjective experience that people have of colors, they will be studied more extensively on their own in the next chapter. Chemistry is concerned with color as it affects dyes, molecular composition and reaction of materials under different conditions, and generally with aspects of color not directly relevant to the computer production of colored designs.

Physics, however, deals directly with light and how it can produce colors, which is more relevant as colors produced on a computer screen will result from these principles. Light is a form of radiant energy composed of electromagnetic waves between approximately 400 and 700 nanometers (nm) in length. These waves vibrate at right angles to the direction of propagation. The greater the amplitude of the light waves, the more intensive the light is and the brighter the illuminated object appears. If we separate daylight, which is also called "white" light, by a glass prism, we obtain a spatial separation of the colors of the spectrum. The resulting colors - red, orange, yellow, green, blue, indigo and violet - are called pure or spectral; they cannot be separated any further by a prism. A light containing only one wavelength or a very narrow spectral bandwidth is referred to as "monochromatic".

Wavelengths are usually measured in millionths of a millimeter, or nanometers. The visible spectrum is between 0.0004 millimeters which is the end of the short-wave ultra-violet end and 0.00075 millimeters which is the start of the long-wave infrared end. Combining all spectral colors gives white; the absence of all wavelengths of visible light is black. Two complementary colors mixed also result in white. Colorimetry, which is the science of the measurement of color, relates color to the stimulus light and to the typical eye's sensitivity. Color mixtures are further defined as additive or substractive. (Agoston, 1979).

An additive color mixture occurs when light from two or more sources is combined, added together, before it reaches the eye. If two beams of light, such as green and red, are projected on a white wall, the combined light, yellow, is the sum of the two beams. This is the principle used in the computer to produce this color. Computer monitors usually have only three bit planes: red, green and blue, which is comparable to television sets. Therefore, to produce yellow, the bit located at a certain x coordinate and y coordinate on the red plane is turned on and so is the bit located at the same coordinates on the green plane. The resulting effect on the monitor is a yellow pixel. For television sets, analog intensities also come into play creating more diverse color results than is possible with discrete technology only. After producing the yellow light, if it is then mixed with blue light, blue stimulating the short wavelength cone of the eye, and yellow doing the same for the long and medium wavelength cones, we then produce again the perception of white light.

Subtractive color mixture can be demonstrated by passing a beam of sunlight successively through two pieces of colored glass which act like light filters in photography. A green filter and a yellow filter produce a yellow-green color. This is because most of the shorter wavelength light (blue to green) is absorbed by the glass, most of the yellow, yellow-green, orange and red light is transmitted, green and red produce yellow, as seen above, and all other light is absorbed or subtracted out. In the same vein, if a sheet of pure red glass and a sheet of pure green glass are held together in front of a lamp emitting white light, no light is transmitted, the subtractive effect giving black. This principle is similar for the composition of paints. A red paint appears red because it reflects only the long wavelength or red portion of the white light which falls on it. In green light, it appears black. On the other hand, when white light falls on blue paint, yellow, orange, red and some green is absorbed, and the rest of the green and the blue is reflected. Yellow paint absorbs blue, orange, red and some green and reflects yellow and a little green. By mixing the two pigments, yellow and blue will be absorbed, as well as the other colors that each absorbs when alone, such as red and orange, leaving only green being reflected by both the yellow and the blue. Therefore, as we know, mixing yellow and blue paint produces green. Most objects are non luminous and their color is produced by the reflective scattering of the light that falls on them. The color of these opaque objects is called subtractive for the same reason that a paint appears a certain color. This part-absorption depends on the pigment, that is the molecular composition of the object's surface.

Another way of looking at color produced by light is called color mixture by averaging. One type is spatial averaging, a kind of retinal mixture or blending. Similarly, if two colored lights alternate too rapidly for the visual process to follow, retinal temporal averaging results, producing a mixed color. Black text on a white page has a gray appearance if the viewing distance is large enough. In the same way, two adjacent colored dots will fuse, appearing as though they were mixed, if they are viewed from a distance beyond the eye's ability to resolve them individually. The difference with additive color mixture is basically the following: in additive color mixture, the energy of two combined beams is the result of adding the energy of the two initial beams and the brightness is increased. In color mixture by averaging, the effective energies of the stimuli are area-averaged and time-averaged in the visual process and a kind of average brightness and hue are produced.

There are many formal theories, standards and measurements applicable to color. Many are discussed and illustrated in "Color Theory and its Application in Art and Design", (Agoston, 1979). Among the most well known are the CIE (Commission Internationale de l'Eclairage) color measurements. They are standards of wavelength composition. For example, "A" represents the light produced by a 500 watt tungsten filament lightbulb, "B", direct sunlight, "C", daylight, and "D" is a breakdown of "C" in the range of 380 to 830 nm. Chroma is the amount of hue in a color, its apparent intensity or brillance. Chromaticity can thus be defined as the quality aspect of psychophysical color, that is the composite representation of approximately equivalent amounts of hue and saturation. It does not include the effective amount or intensity of light. The psychophysical amount is then taken as the physical values weighted by the human visual system's efficiency. The CIE chromacity diagram and the Maxwell chromacity triangle, named after its inventor the Scottish physicist James Clerk Maxwell (1831-1879), are examples of this kind of system.

There is even a natural color system (NCS), (Agoston, 1979), which provides an effective means for everyone with normal color vision to make color evaluations without the use of color measuring instruments or color samples for comparison. A color determined in this way is an absolute measurement based on color perception. It differs from psychophysical determinations which rely on color matching. This system can be traced to the German physiologist Ewald Hering (1834-1918).

Finally, one cannot forget the color circle of the famous German scientist Goethe. This color circle, based on the physical phenomenon of light addition and subtraction, identifies six primary colors of which three are additive (red,green and blue) and three are subtractive (yellow, cyan and magenta), as the following array represents. These are the colors that every computer system with eight colors has in addition to black and white.

...	BLUE	CYAN	GREEN	YELLOW	RED	MAGENTA	...

Figure B.2 Goethe's Colors.

The division of colors based on light (physics) is therefore different than the division of colors based on pigment (art, especially painting). It is easy to see that yellow must be a primary color in art because no mix of red and green paint, nor any other combination for that matter, will ever produce it. Green, however, is easily produced by mixing blue and yellow. On the other hand, anyone who looks at color television, where screen phosphors are self-luminous rather than reflective, cannot argue that an appropriate mix of red, green and blue light cannot produce all other existing reproducible colors. All in all, if we compare the six primary colors of physics with the three primary and the three secondary colors of art, we see that only one differs. It is cyan (turquoise,blue-green) in physics and orange (yellow-red) in art. This basically stems from the ambiguous position of yellow and green, each a primary in their own field but a secondary mix in the other. The other two colors, red and blue (both primaries), and their combinations (magenta/purple/violet), will not change in either. The difference cannot be looked upon as right or wrong. It is simply a function of the medium used to produce the colors or, more precisely, the wavelengths that stimulate the eye. As colors are usually considered as a sensation produced in the brain in response to light perceived by the eye, (Agoston, 1979), the important thing to remember in using colors is really a matter of perception and of the psychological effects that they produce, whichever classification scheme is used.

APPENDIX C

BIOLOGICAL, PHYSIOLOGICAL AND PSYCHOLOGICAL ASPECTS OF COLOR

In the previous appendix, many of the phenomena usually found in a psychology textbook on color have already been discussed. Color classification systems, simultaneous contrast, color diffusion, additive and subtractive color mixtures are only a few of those items which span a variety of disciplines. Placing them, in this text, under a particular heading only emphasizes their primary characteristics or where they are most likely to assume greater importance. Sensory psychology is built on a sound scientific basis and as such, includes experiments relating to the measurements of the physical aspects of color composition to measure their subjective effects. It thus adds the human element to purely physical considerations, and can see and interpret the results through subjective human eyes. Color is a sensation difficult to define to another person without using examples. It is also a very private experience. If three persons agree that the sky is blue, they may not perceive the same blue. Going even further, one person may perceive green, due to some color deficiencies in his or her eyes, but because blue is the accepted color for the sky, that person very honestly considers that perceived hue as blue. To understand some of these perceptual differences and to discuss the psychological effects of color, it is therefore important to give a brief overview of how vision, and color vision in particular, is seen from a biological and physiological point of view. (Goldstein, 1980).

The nervous system transforms environmental energy into electrochemical energy and communicates this coded information from one part of the body to the brain or vice-versa through a network of neurons. For vision, the occipital lobe of the cerebral hemispheres is the receiving area, also called the primary visual cortex. The eye is where light first makes contact with the part of the body which changes its electromagnetic energy into neural impulses. The eye is composed of many parts. The optical elements include the cornea, lens, pupil, and intraocular fluid. The neural portion includes the multi-layered, light sensitive retina, with its densely packed fovea and output via the optic nerve. In the brain, the lateral geniculate nucleus and the visual cortex complete the chain. Light enters the front of the eye through the cornea, the complex receptor neurons in the retina located at the back of the eye change this light into neural activity, which then leaves for the brain through the optic nerve to go into the lateral geniculate nucleus in the thalamus and then into the visual cortex itself. A visual pigment molecule is composed of a protein called opsin and a light sensitive group called retinal. On the average, the isomerization of about seven visual pigment molecules results in the perception of light, though, under optimal conditions, interaction with a single photon by one pigment molecule is perceptable. The retina is not a straight-through system. A number of cells send their signals to one cell resulting in convergence; in addition, lateral interaction between cells takes place, resulting in a phenomenon called lateral inhibition. Simply explained, this lateral inhibition means that if one cell is considerably excited, it sends messages to its neighbors inhibiting them to send their own messages to the brain so that this cell with the message having the highest content has a clear path to the brain, which can then focus its attention on it. A special case of visual perception called Mach bands illustrates the fact that there is a close connection between physiology and perception and that what we perceive is not necessarily an exact copy of the energy distribution in the environment. Mach bands, a result of lateral inhibition, sharpen contours and edges by increasing and decreasing the contrast between the signals which occur during a transition from lighter to darker. Therefore, if our nervous system were different, with no lateral inhibition, we would see colors and brightness differently; as evidence, the problem of enhancing edges to make them sharp is a difficult one in the field of computing vision. The optic nerve is made up of "on" fibers, "off" fibers, and "on and off" fibers which interact with each other to analyse the stimulus applied to the receptors. Similarly, we find cells in the cortex which respond only to specific stimulus features, such as orientation, corners, angles, length etc.. Sensory coding of the properties of the environment is necessary to perception; different theories suggest that this is done either through specific nerve fibers or through a pattern of activity across a large number of nerve fibers. Experiments have proven that both are correct and connected.

Specific to the visual system and to color perception, are the rods and the cones which make up the photosensitive layer of the retina. The 120,000,000 rods in the human retina operate under dim illuminations and the 6,000,000 cones operate under moderate and high illuminations, which means that at night we see with our rods and during the day we see mostly with our cones. Rods are, in fact, one hundred times more sensitive to light than cones. There is one type of rod and three types of cones (long wave, medium wave and short wave). Measuring the wavelengths of the rods and of the three types of cones, we see that rods function at wavelengths approximately located between 400 nm and 610 nm, short wavelengths cones between 400 nm and 525 nm with maximum absorption at 445 nm which is about where the blue hue is located, medium wavelengths cones between 440 nm and 625 nm with maximum absorption around 540 nm in the green region, and long wavelengths cones between 450 nm and 690 nm, with maximum absorption at 570 nm closest to the wavelength produced by red. Together, they produce the red-green-blue of color vision, necessary for the perception of all other colors. Rods and cones are distributed differently on the retina; the fovea, a small area located directly in the line of sight, contains only cones. Rods and cones also differ by the composition of their visual pigments and by the fact that they are connected differently to other neurons in the retina.

The phenomenon of dark adaptation and light adaptation is interesting in terms of color perception. Dark adapted sensitivity is about 100,000 times greater than light adapted sensitivity. This means that going into a dark area, our eyes and neural part of the visual system adjust and we see better as we spend more time in the dark. It is mainly due to the process called pigment bleaching and regeneration, where the retina first becomes transparent and then regenerates itself. The two molecules previously mentioned, retinal and opsin, separate and rejoin to produce this phenomenon. Applied to color, this process can account for a different perception of brightness, remembering that color is composed of the three dimensions hue, brightness and saturation. An experiment made showed that two flowers, one red and one blue, which appeared equally bright in the light, when seen later in the dark changed first to gray, then the blue appeared brighter as time went on. (Goldstein, 1980). Red and blue reach their saturation point at the same level and both are pure hues. This change of brightness with dark adaptation is called the Purkinje shift and is due to a combination of the following factors: only cones perceive colors, rods see in black, gray and white, rods are more sensitive to short wavelengths than cones and blue is a short wavelength color in the spectral sensitivity curves for rod and cone vision. Green occurs in the medium range and red in longer wavelengths. Another experiment conducted with a dog is also interesting with regards to perceived brightness. Trained to differentiate between a blue ball and a green ball, when the light intensity or brightness of the object was increased to compensate for the change in hue, the dog was fooled. The fact that hues reach their respective saturation point at different levels of brightness will be discussed further in the analysis of color as a visual variable. Getting back to the eye, the summation of the different responses to a given wavelength in each of the three cone types results in our perceived value for this color. Since there are more rods than cones, more rods converge into a single ganglion cell, accounting for more of the spatial summation process. For the same reason, it is then more difficult for the brain to pick out which rod was actually more sensitized. Therefore, the visual acuity of rods and spatial localizability of stimuli is less than that of cones.

Perception of colors is also affected by other factors. One of them is the reflectance of the object observed. The region of the spectrum reflected by an opaque object determines the object's color. For a transparent object, it is the selective transmission of certain wavelengths which determines its color. The area surrounding the object is another important factor in the perception of a particular color. Simultaneous contrast has already been discussed under "art" and the influence of neighboring areas is an important concept which will be covered in more details under "cartography". It is easy to see why when we realize that the basis for the influence of the surroundings is a physiological fact caused by the observer's light, dark and selective adaptation mechanisms. Selective adaptation is where the eye sees the same color under different illuminations, adapting itself to changing environments. The best and most simple proof of it is using the same color film in daylight and artificial lighting conditions. Since the film does not adapt itself, colors on the pictures may look different while to our eyes they do not. This is a useful principle to remember as it may account for the possible differences between color photography, color reproduction, and the color image on a computer screen seen under different illuminations. What we take for granted physiologically may lead to differences in the results produced by different technical processes.

Two interesting psychological theories are the trichromatic theory and the opponent-process theory. (Goldstein, 1980). The trichromatic theory basically states that any three wavelengths can be used to match or to produce any color as long as we cannot match one of them by mixing the other two. This theory, advanced by Young and Helmholtz in the early 1800's, is based on the psychophysical result which states that because a combination of three wavelengths can appear identical to a single different wavelength, the visual system cannot tell the difference between two physically different stimuli. Therefore, two lights with different wavelength distributions appear the same color if they stimulate the three visual receptors, i.e. the short, the long and the medium wavelength receptors, in the same ratio. The opponent-process theory, proposed in 1878 by Ewald Hering, the same scientist who devised the previously mentioned color classification system, explains color vision by hypothesizing blue-yellow, red-green and black-white mechanisms in which a color in each of these pairs has opposite properties to the other. This opposition is supported by the fact that a bluish-yellow or a greenish-red has never been found to exist and by other phenomena such as simultaneous contrast, after-images and color blindness. A color-blind or color-deficient person who cannot see red usually cannot see green either and the same applies for the yellow and blue pair. Both theories are again, of course, correct, with trichromatic theory explaining the operation of the receptors and opponent-process theory explaining the operation of higher order neurons.

A trichromat is a person with normal color vision. There are anomalous trichromats but as the reasons for this type of different color vision are numerous and individual, one cannot summarize the hypotheses in a few sentences. A dichromat is a person who, usually for genetic reasons, possesses only two cone pigments and is missing the third. There are, of course, three types of dichromats: those who can see long and medium wavelengths, i.e. missing the short wavelength cone pigment, those with short and long wavelengths vision and those having only short and medium wavelength cone pigments. Dichromats are considered to have color deficiencies as they can only perceive some colors as people with normal vision do but not other colors where the missing wavelength would make an appreciable difference in the summation process. A controversial theory was proposed by Land to the effect that only rods and one cone are necessary for color vision. Without debating the merits of this particular theory, it is only necessary to mention that his principal claim that light is independent of flux, the color constancy principle, has had considerable influence in the development of artificial intelligence procedures dealing with computer vision. In general, monochromats who have only rods and sometimes one type of cone, are referred to as truly color-blind.

Color-blindness or rather anomalies in color vision affect a greater percentage of the male than of the female population. These people usually adapt well to the deficiencies in normal life circumstances, i.e. the top traffic light is red and the bottom one green. Brightness cues also help where cones fail. However, the greatest difficulty comes when they are required to match a number of colors especially in surrounding areas. The handicap is then more visible and sometimes embarrassing. A computer system which could remove some of this fear by doing appropriate matching and contrasting could help people suffering from color perception deficiencies not only in producing color designs but also in learning to appear to perceive colors as the average person sees them. Color is one of the most obvious visual qualities experienced by human beings and one with which a person is most actively involved in daily experiences. Color can make a profound impact on people's reactions; as one of six visual variables, its message cannot be ignored in the design process.

APPENDIX D

THE MESSAGE OF COLOR AND OTHER VISUAL VARIABLES

Having seen some of the more objective and scientific aspects of vision and color, we can now turn to the message that retinal impressions can convey. Jacques Bertin, in an excellent study entitled "Semiologie Graphique" (The Science of Signs), organizes retinal variables in various categories and explains the meanings that each can convey. (Bertin, 1967). The notion of level of organization is considered fundamental. Perception of retinal variables can be associative, selective, ordered and quantitative. The variables themselves are classified as size, value, texture, color, orientation and shape.

Size relates only to the dimensions of the object (i.e. small circle, big circle); value is the different shadings that the object can take like, for example, from white to black through various tones of gray, or light blue to dark blue etc.; texture, in two-dimensions, is viewed mostly as the thickness of lines or patterns covering a certain area, the best example being "hachuring" on a map going from very thick to very thin lines; color is seen as pure or saturated, not to be confused with value, so that only one shade of each color is considered (i.e. one blue, one yellow, one red etc.); with orientation, the shape of the object itself, especially lines, assumes a spatial orientation with respect to the page (horizontal, diagonal, vertical etc.); finally, shape is the most diverse and self-explanatory retinal variable since all shapes, geometric or free-form, fall under this category.

Returning to levels or organization, let us examine how each of the retinal variables can fit into each level and thereby convey an instantaneous message to the viewer. The word "instantaneous" is very important here; relating it to a map, it means that one does not have to study the legend or spend a lot of time looking at it to get the visual impression which the variables attempt to convey. The level of associative perception can be defined in terms of equality, such as a regrouping of the correspondence of all the categories of this variation fused with one another. All signs have the same visibility. An associative variable does not vary the visibility of signs. On the other hand, selective perception does entail a variation in visibility. It is used to classify signs into categories, isolating all the elements of a particular category and rejecting all other elements to perceive the image formed by the category in question. Ordered perception attempts to compare two or more types of signs, giving a spontaneous universal ordering from small to large, low to high etc.. For quantitative perception, a ratio or rapport is defined as well as an ordering, producing natural steps in the progression.

The table which follows indicates which of the six retinal variables can spontaneously convey any of the four levels of visual organization:

	Associative	Selective	Ordered	Quantitative
Size	no	yes	yes	yes
Value	no	yes	yes	no
Texture	yes	yes	yes	no
Color	yes	yes	no	no
Orientation	yes	yes/no *	no	no
Shape	yes	no	no	no

* yes for points and lines, no for surfaces.

Figure D.1 Properties of retinal variables.

It is interesting to note that none of the retinal variables has all the dimensional properties. Only size can convey all three of selection, order and quantity and, as a corollary, a perception of universally instantaneously measurable quantity can only be conveyed by size. Size and value are not considered associative; this may be a little difficult to comprehend at first and to do so, we must recall the property of spontaneous perception and the similar level of visibility necessary to qualify under associativity. For example, in a map representing the density of the population by occupation, if farmers are in black, blue collar workers in gray and white collar workers in white, the eye will spontaneously perceive only the density of the farmers. Similarly, large black circles would be perceived before medium or small black circles. Texture is associative, selective and can be ordered, while orientation is associative but can only convey selectivity when it is used with points and lines; a surface is not perceived as being oriented. Shape, with its billion possibilities, is only in itself associative. Similar shapes with no variation of visibility can be grouped but if the visibility varies, the shapes are perceived as different and the general "shape" made by these shapes cannot be spontaneously reconstructed. Finally, we see that color, which is of primary importance here, can only convey spontaneously associativity and selectivity. The usefulness of dwelling shortly upon the other variables in the previous lines becomes evident here, as it will be necessary to combine these variables with color to make it convey any other dimension or message, such as order and quantity.

Color is considered here as the noticeable difference between two rectangles of the same value. When combined with the notion of value, it produces a two-dimensional array which can be traversed horizontally (all colors of the same value) or vertically (all values of the same color). Each element of this array is considered as a tone which has two parameters: the color, blue, green, yellow etc., and the value, which is the percentage of black of the corresponding gray. A saturated color, from a psychophysics point of view, is one to which no white or black has been added; from the artist's point of view, it is called the pure tone. In colorimetry, the saturated tone is the narrowest band extracted form the spectrum (purity factor), the wavelength of the centre of the band (chromaticity) and, for a more rigorous definition, the luminance factor or value of the tone is also considered. The value of the saturated tone varies with the color. This can be best described by picturing an array where color is on the horizontal scale and value on the vertical scale. The saturated tones would then form an inverted pyramid with pure yellow being in position (4,4), pure orange and pure green in (3,3) and (3,5) respectively, followed by blue (2,2) and red (2,6), the base of the pyramid boasting purple (1,1) and magenta (1,7) at its corner points.

purple						magenta
	blue				red	
		orange		green		
			yellow			

Figure D.2 Saturated tones.

From the above come many of the problems associated with the use of color and their numerous consequences in the world of graphics. Given that the scale of pure tones combines color and value and that each of the two regions of the spectrum gives an ordered scale different from the order of values, spontaneous visual properties are established on the order of the values and mix the two regions of the spectrum. So, even if the scale of cool tones and the scale of warm tones are each well ordered, mixing the two scales can be a source of visual confusion, unless the equality curves are respected and the image reestablishes the continuity of the spectral series. Color does not give a universal sense of order by itself; to do so, it must be mixed with value, size or texture.

As for selectivity, the choice of selective colors is different depending on the value. Selectivity is at its best near the saturated tone and diminishes as it goes away from it. Within light tones, yellow, green and orange are generally chosen to color zones without overloading the design. Medium tones offer the widest choices of selective colors, blue and red being saturated at this level. Darker tones are better to color objects which detach themselves form the background. While purple and associated colors do well at this level, there is very little difference between dark yellow, dark orange and other dark dullish tones.

In nature, no color sensation is perfectly pure. It is a mix which we must spatially differentiate. A surface of a uniform color opposed to another uniformly colored surface is the closest to a significant difference. As areas become smaller, the natural mix or "color diffusion" takes over. Therefore, color is in a way a function of size. It has been suggested that a colored area should have at least 1.5mm in diameter to be considered a useful color variation. Color diffusion is a useful property as it allows the creation of some new colors with given colors. This property is the basis of trichromatic reproduction. With a blue, yellow and red we can reconstruct the entire spectrum; these three colors can also be used for color synthesis in various combinations. Finally, fuzzy or blending outlines favor the diffusion of color over and above its last spatial point.

Problems such as these are not really an impediment to the use of color, but rather something that the designer must consider not only to avoid confusion but also to create results taking advantage of these properties. Case in point: although color is, in general, seen as associative, certain colors of the spectrum do present a difficulty. If we consider points and lines on a map, light colors such as yellow are not clearly seen; therefore it is best to avoid them in these circumstances. Also, the implied ordering of warm and cool tones is not universal enough to use in a quantitative way but is extremely useful in an intuitive way. For example, using color to indicate various centuries would not by itself be a sufficiently precise ordering criterion but an auspicious choice of the implied warm and cool ranking of colors combined with another variable such as size would double the impact of the message to be conveyed.

This brings us to the symbolism and aesthetics of color which are two of its biggest advantages. These are mostly marked in nature where fire and hot climate are normally associated with warm tones, vegetation with green, water with blue etc.. Even at the abstract level, in accordance with particular cultures, the symbolism of color is evident such as pink and pale blue for the birth of a child. Aesthetic rules are necessary for the best use of the remaining margin. Harmonies of color are made with nuances or values of the same color or with contrasting colors. Complementary colors, those diametrically opposite in color circles, provide the best complement. The artistic effect depends on the size of the area colored. The complementary area must be small with regards to the dominant color.

Summarizing some of the advantages of color seen as one of the six retinal variables, we see that it is an excellent selective variable. It can easily be combined with other variables and, as such, is easy to memorize. Foremost, it has an undeniable psychological attraction; it helps keep the attention of the viewer, increases the number of interested readers, helps memorization and increases the scope of the message. Highly desirable in any form of training, it must however be made to conquer its two main disadvantages: the anomalies in color perception (color blindness) by a number of persons and its expensive reproduction costs. If it is possible to make a computerized color system in such a way as to minimize these difficulties, we will all be able to enjoy the enormous advantages of one of the best retinal variables combined with the appeal to intuitive attractiveness that only it can bring to the two-dimensional plan.

APPENDIX E

COLOR AND CARTOGRAPHY

One of the leading researchers who has had a profound influence in cartography, especially when dealing with color, is Eduard Imhof. Although Imhof is best known for his use of altitudinal tinting, he has also contributed greatly in other aspects related to color mapping. To examine some of his theories is a necessity in any work on color, on mapping and especially on the design process of colored maps. (Imhof, 1982).

Imhof places great importance on the necessity for and the careful development of good interplay of map elements. "Correct combination and good coordination is of great, even of decisive, significance for the success of a map", (Imhof, 1982, p.325), states Imhof in his book Cartographic Relief Representation , originally entitled in German Kartographische Gelandedarstellung. Selection of elements and a few basic representative color tints can ensure a good conceptual interplay. Good graphic interplay, on the other hand, is what simplifies map reading. For example, some patterns of lines combined with uniformely tinted areas produce a good effect when the color background does not interfere with the line image. The reproduction process must then maintain good technical interplay. This interplay is often spoiled when many cartographers work on different aspects of the same map; a computer mapping system can help preserve the integrity of the design by allowing an up-to-date view of the state of the map before, during and after every change. As far as color is concerned, the principle of appropriate relationships between elements on a map even extends to the use which will be made of the map. A topographical map for use in the field should be shaded lightly and given color tints to improve its legibility while a wall map to be seen from a distance benefits from brighter colors criss-crossed by lines and symbols like a painting would.

Overlapping and proximity of graphic elements can affect tonal mixtures, contrast effects and may even result in optical illusions. A colored land tone is modified by densely packed contours resulting in a mixture of both elements. Similarly, the color of rock areas can be affected by close networks of black, grey, brown or reddish rock hachures. Small, densely packed dots or symbols also affect the color and, the forested areas, if too dark, can destroy a three-dimensional map effect. These types of effects must be considered when combining symbols with color as well as when coloring uniform areas in close proximity. Imhof summarizes how the interplay of elements should be considered:

> "One should also consider the ways in which the colors of the combined elements may harmonize. Things which contrast in information content should also be contrasting in color. Those things which are related in their meaning should not have clashing colors. The most important element, as far as the information is concerned, should stand out." (Imhof, 1982, p.334).

One needs artistic talent to reproduce the complex play of light, shadow, color and atmosphere in a naturalistic map image. Shading should enhance, yet not submerge, surface tones. Contours are used to display and reinforce the colors of the chosen shading without being too obtrusive. Twelve colors (brown, blue, reddish-brown, blue-grey, violet-grey, light blue, yellow, pink, black, violet, green and red) are considered sufficient to produce a good colored map; with the advent of computer generated colors, these can now form the base for a much larger selection, allowing, as Imhof suggests, the production of a good artistic impression as well as reflecting as much accuracy as necessary.

Concentrating on the basic color applied to an area rather than on the rest of the superimposed graphic elements which, although necessary in the overall map, are not of immediate importance in the proposed system, let us examine which color theories have been mostly influencial in Imhof's choice of coloring systems.

The preparation of color maps requires a good knowledge of the characteristics, effects, standardization, harmony and composition of color. Considering some of the artistic, physical, chemical, physiological and psychological properties of color, mentioned in the previous chapters and appendices, Imhof has developed a way to show a naturalistic perspective. Grading colors by their intensity, brightness and contrast, from high ground to low ground, gives a subconscious impression of "far" and "near", "high" and "low".

Two of the systems previously described, the Ostwald double cone and the Hickethier three color cube, are considered more than sufficient for the production of colored maps and, as an added advantage, lead themselves very well to computerization, especially if the computer is capable of generating various intensities, reducing the immense storage which would otherwise be required to handle each of these combinations in a directory. As a practical tool for cartography, however, even if all these colors could be generated by a computer and if the human eye is, in theory, [1] capable of differentiating these one thousand different tones, it is much better to reduce the number of colors to a manageable level, not only for aesthetic and financial reasons, but because, as we have seen before, communication of a message through color is better achieved by a judicious mix of appropriate colors which appeal to the psychological and to the interpretive side of the intelligence of the viewer. For points and lines, because of their small dimensions, only a few solid colors containing as much contrast as possible should be used.

Some of the most important aspects to consider in choosing colors are the rules of composition and harmony which can make or break a map as a means of an attractive and meaningful

----

[1] In practice, we will see that it is even difficult to differentiate between 49 or 64 colors produced by a high resolution color computer monitor. If some of these colors are adjacent, the eye will fuse them together.

communications medium. When combining two or more colors, generally the unbroken sequence of colors from the color circle is perceived as harmonious. Compounds of two or three complementary colors also have an harmonious effect.

Subdued colors, lightened by white or darkened by black, are often more pleasing than pure colors. For brownish colors, the best complement is the one which has more of the original color which is weak in the brown in question. For example, a violet-blue will complement well a yellowish-brown. In general, if the subtractive mixture of two colors produces white and their additive mixture black, these colors are considered harmonious. On a map, we perceive colors to be harmonious when they correspond to our daylight perception of familiar objects. To this, Imhof adds an unconscious striving for order

> "Psychological color perception always tends, therefore, in the direction of composing complementary colors." (Imhof, 1982, p.71).

which may explain why some bright colors such as red and violet or blue and violet, placed in close proximity to each other, may have an unharmonious effect because their combination does not make up white, gray or black. In fact, a lot of the uncertainty or "maybe" encountered in the above discussions, relates to the fact that only color is considered here while on a map, the size of the area being colored is of primary importance. There is no doubt that unequal areas colored with tones of unequal intensities do produce a better harmony. Pure and bright colors do better in small areas. Large areas and background benefit from duller colors. Natural gradation colors emerging from the same hue, i.e. light blue to dark blue through several other shades of blue, produce very pleasing and calming effects, imitating what in nature can be produced by distance viewing or different light intensities.

In the whole, it is composition which determines a good or a bad graphic work and color has a large role to play in the composition rules which are especially applicable to map design. Imhof states six main rules to be followed in designing a good map. (Imhof, 1982, p.72-73).

First, "pure, bright or very strong colors have loud, unbearable effects when they stand unrelieved over large areas adjacent to each other, but extraordinary effects can be achieved when they are used sparingly on or between dull background tones".[5]

The second rule also refers to the size of the areas colored. "The placing of light, bright colors mixed with white next to each other usually produces unpleasant results, especially if the colors are used for large areas".

Thirdly, "large area background or base-colors do their work most quietly, allowing the smaller, bright areas to stand out most vividly, if the former are muted, grayish or neutral".

The fourth rule maintains that two large areas of different colors split the picture. Unity can be achieved by intermingling colors. "The colors of the main theme should be scattered like islands in the background color".

In this vein, " the composition should maintain a uniform, basic color mood. The colors of the landscape are unified or harmonized by sunlight". However, this uniform mood should not be exaggerated to avoid the map looking too " dull, jaundiced and dead".

The sixth and final rules stresses the importance of a steady and gradual de-emphasis of colors. "A continual softening of area tones is of primary importance in cartographic terrain representation", but, "this principle is in no way opposed to a contrary requirement, that of contrast effects". (Imhof, 1982).

The above points out that the artist's freedom is still alive and well. A clear map, where some important areas are emphasized while the unimportant areas are subdued, allies the best of the scientific principles and the artistic graphic laws. A good example of color theme in cartography is Atlas zur Raumentwicklung. Each of these volumes adopts a specific color as a main theme which is also the color used for the border of the pages. A secondary theme is also often present and all the maps play on these color themes using others only as necessary complements when more are needed. The effect is a beautiful as well as an informative piece of graphic work. In cartography, "clarity and beauty are closely related concepts".

Going hand in hand with the graphic laws of composition, is the ever present symbolism of color. Already well established in conventional mapping, some color choices come from nature itself (the blue water), some from climatic conditions (the red tropical zone), yet others by tradition (the red/yellow population density based on skin colors). New research is under way to associate map colors with particular physiological reactions such as the colors which are best perceived by the human eye under night flying conditions.

The graphic concept which Imhof is most famous for is called altitudinal tinting. This is the use of colored area tints to achieve a desired impression. In topographic cartography, he distinguishes three main methods of utilizing area tinting: 1) representing the surface of the earth with hues differentiating between rock zones, snow, vegetation etc. 2) hypsometric tinting where color tints represent the elevation zones of the earth's surface 3) a combination of the first two. From this can be distinguished three different approaches of color representation: 1) stressing the imitation of natural colors 2) using conventional symbolic colors 3) combining naturalism with symbolism.

Ideally, in any reproduction of terrain, one should strive to imitate nature as much as possible. But, as natural colors vary according to distance, the time of day, the season etc., a generally representative depiction of the predominant aspects of the landscape, in accordance with the intent of the map, is acceptable. This is because the impression one gets of nature is not based on one particular shade or tint but rather on the interplay or harmony of the various elements in the landscape. The importance of the interplay of graphic elements has already been mentioned and this surely holds true of colors.

Hypsometrically layered maps are those that use of color to indicate altitudinal steps. The selection of appropriate colors for the various elevation layers has always been a more difficult problem for the cartographer than the determination of the height to give to the layers or the actual fitting of colors to the layers. Since the same tint cannot repeat itself at various elevations, the number of layers has often been based on the number of tints available in a color scale, generally six to ten. In paper cartography, the distinction between a gradual vignetted transition between colors or a relatively sharp transition has also been a factor to consider. With computer cartography, to produce a very gradual transition, three or more similar colors can be used but they are still considered different colors. The discrete steps which form part of the nature of computing reduce this problem to the one of harmony or gradation between adjacent areas, infinitely small as they may be.

Historically, dark and contrasting scales have been used which have more recently been replaced with height scales. Imhof describes some thirteen types of scales which can be summarized as follows (Imhof, 1982, p.300-310):

The first type, developed as soon as color printing became available, consisted of colors aimed to achieve the maximum contrast and differentiation between layers. In relief maps, these led to confusion as the continuity of the terrain and the visual appreciation of relief was disturbed. Then came two gradation principles, one the exact opposite of the other. Their names, "the higher, the lighter", and, "the higher, the darker", are self-explanatory. This process is still used today as it provides a relationship between elevation and color intensity. Types four and five were developed at the end of the nineteenth century and they are still in wide use today. The top elevations consist of deep brown, medium brown, light brown, yellow, yellowish-green, green, blue-green, while the bottom was deep blue grey-green. The yellow seeming too often too bright and out of place, it was omitted in step five and replaced by olive-brown and olive-green. This still led to a perceptual difficulty, as steps were not easily differentiable. The sixth type replaced the top layers by grey-violet tones but was, in fact, very similar to the previous two. At the beginning of the century, Karl Peucker developed another scale similar to a portion of the spectral series. Although seen by some as a revolutionary because of his consideration of stereoscopic effects and his physical theory base, Imhof found some contradictions in his theories, whereby perception of three-dimensions is not physiological but rather psychological. His scale, however, because of its aerial perspective, does have its good points and is worth mentioning here. On the top layers, we have red, red-orange, yellow-orange, yellow, greenish-yellow, yellowish-green, grey-green with grey at the bottom. Further variations of this type of color scale were then produced leading into white glacial top steps, which gives two very light areas, one yellow at the bottom and one white at the top, again leading to some perception difficulties. Then came a color sequence better for three-dimensional effects because it conformed to phenomena present in aerial perspective, i.e. the changes in color and shadow tones with increasing distance. In this context, nearer is higher, permitting contrasts to be identified in an upward movement, getting lighter towards high altitudes. White, yellow, green and blue with appropriate mix in between constituted the subsequent two color schemes proposed. The third replaced the upper white by a pinkish tint to appear closer to nature in color and to differentiate them from real permanent ice and glacier areas. Again, light colors are predominant. Type twelve is basically the same, allowing slightly stronger colors and a more stepped gradation for smaller-scale maps. The last type mentioned, first used in 1962 by Imhof, most closely approaches type nine, giving the maximum impression of three dimensions in elevation. The principles which, in hill-shaded maps, strenghten green in the middle and soften red at the top do not apply to small scales where white is chosen for any kind of rocks at top altitude since at this scale, rocks are undistinguishable from glaciers. A successive lightening of tones toward the top ensures an increasing intensity of contrast between light and shade in the upward direction.

The effect of distance produced by colors depends more on lightness than on the color themselves. This sequence comprises the following stages from top to bottom: white (including glaciers), light yellow, reddish-yellow, fuller brownish-yellow, light yellowish-brown, light greenish-brown, light brownish-green, blue-green, strong green-blue, strong grey greenish-blue. In the whole, this scale is light, improving hypsometric gradation.

There are other maps, similar to thematic maps, which depict strictly elevation. In this case, it is quite appropriate to select a series of individual hues, increasing sharply in their intensity or in their dark shading content, in order to bring out distinctly the differences between a large number of elevation levels. However, even in these cases, a too wide and arbitrary range of color sequence is inappropriate, a tonal gradation being much better for the perception of the proper order of succession. Here, the colors themselves make the gradation sequence structure.

Many other varied arrangements are possible depending on the intent of the map. As far as ocean depths are concerned, " the deeper, the bluer", is the generally acceptable convention with the top layer of water being almost white to differentiate it properly from the land. If there are no layers, a rich blue is appropriate for water and, if the blue scale is not big enough to accommodate all layers, green can be added at the very bottom layers, as opposed to violet which would render it too similar to the land structure.

As for the layering scale, six to ten steps are usual, with the best method being smaller steps in low-lying areas and increasing with altitude according to a geometric type of progression. Under water, shallow areas have smaller steps followed by equidistant steps at depths below the upper continental shelf. Once the color tones and step heights have been selected, it remains to adjust the color tones to the steps. Such adjustments may be necessary due to the scale of the map or what features the cartographer wishes to emphasize. The ability of the observer to differentiate between layers is the main concern. The use of boundary or contour lines also has its advantages and disadvantages and remains a prerogative of the cartographer. Lines emphasize edges and permit better differentiation from similar tones in adjacent areas. However, since there are no lines in nature, they can cause confusion. Again, the first principle of the interplay of graphic elements plays a significant role in the choice made.

In conclusion, Imhof recommends the following color chart as a guide to the production of altitudinally tinted maps. In this chart, the numbers 1 to 4 refer to four different, well-balanced tint screen steps, progressing form the lightest to the strongest, number 5 representing the solid color. These are combined to produce the desired image colors. (Imhof, 1982, p.323).

	step #	elevation and depth steps in meters	light blue	light red	yellow	dark blue
L	1	over 4000	-	-	-	-
A		2000-4000	-	-	2	-
N	3	1000-2000	-	1	4	-
D	4	500-1000	(1?)	2	4	-
	5	200-500	2	2	4	-
	6	100-200	3	1	4	-
	7	0-100	5	1	3	-
	8	depressions	5	-	3	2
D	9	0-200	1	-	-	-
E	10	200-2000	2	-	-	-
P	11	2000-4000	4	-	-	-
T	12	4000-6000	5	-	-	-
H	13	6000-8000	5	-	-	1
	14	under 8000	5	-	-	2
	15	lakes	5	-	-	-

Figure E.1 Imhof's color chart for map reproduction.

One of his best examples is illustrated in color plate 5, included in his book Cartographic Relief Representation (Imhof, 1982). The following diagram shows the approximate underwater and land color gradations. Zero is used to indicate the lightest tint; shades get progressively darker as the color number increases.

depth/altitude	tint
	brown 3
a 4000	
	brown 2
a 2000	
	brown 1
a 1000	
	beige
a 500	
	yellow
a 200	
	light green
0	
	whitish
d -200	
	turquoise 1
d -1000	
	turquoise 2
d -2000	
	blue 1
d -3000	
	blue 2
d -4000	
	blue 3
d -6000	
	blue 4

Figure E.2 Hypsometric tinting example.

APPENDIX F

CARTOGRAPHIC PRINCIPLES

As we can see, cartography emerges as a "natural", in both senses of the word, for a color design system. It needs the skills and intuition of the artist to produce an attractive map; produced on a computer screen, the laws of physics on light mixing will influence the mechanics of color reproduction; since maps are read by a great variety of people, the way humans perceive colors biologically and physiologically must be taken into account; psychological reactions of readers will ensure the success or demise of a particular map; and finally, as maps are primarily intended to convey a message, the communications possible using color as a medium will be truly enhanced by the knowledge of the scientific, intuitive and expressive aspects of color.

Before coloring a map, however, it is important to understand some basic information about mapping. In mathematics, mapping is a translation from one vector space to another; in geography, it translates the real multidimensional world into a two dimensional sheet of paper or a two dimensional computer screen. [1] The main elements of cartography, like in any graphic

----

[1] Three-dimensional effects achieved through rotation and other geometric manipulations will not form part of this system but could form part of a future extension. Elevations may, in certain cases, be depicted by a choice of different tints only, not by special effects to simulate visual depths.

system, are points, lines and areas. If we let a pixel, which is one dot of light on a computer screen, represent a physical point, it is, for all practical purposes, too small to be of any use in depicting a real entity. Even a small city will consist of several pixels. Maps have the advantage of yielding a simultaneous or synoptic expression of a variety of information being presented. They isolate some important spatial properties such as orientation, shape and relative location. Through scale changes, they allow selection and generalization, modelling the real world to facilitate understanding, and act as an important communication device in many human endeavours. The disadvantage is that, without a proper understanding of the phenomena previously described, maps can also be made to lie or convey an erroneous message.

The measurements used on maps are of basically three main types: nominal, where numbers are essentially names, that is symbols not mathematically manipulable such as the binary "0" and "1"; ordinal, where data can be ranked according to some criterion such as the size of areas; and interval and ratio , where differences are measured with an inherent zero for ratio, such as the amount of precipitation, and without an inherent zero for interval, such as temperature.

Maps are also often classified as depicting mainly points (dot maps), lines (network maps), areas (colored area maps), and surfaces (colored contour maps). In any map, spatial relations are very important, often leading to what is called spatial autocorrelation. In 1970, Tobler defined this concept very simply by stating something to the effect that everything is related to everything else but near things are more related than distant things. (Unwin, 1981). Occasionally, however, the designer must sacrifice theoretical elegance for visual effect. This is where color can play an important role.

One of the most surprising things in trying to describe a map, is the sheer amount of information even the simplest map contains. Areas are made of lines, lines of points and absolutely everything is spatially related to something else. This is one of the main problems that everyone who has attempted to computerize the mapping process has been faced with. No wonder color, usually a finishing touch, has never been properly studied in that context, considering all the work which had to precede it only to get an acceptable map on the screen. Some of this work is worth examining at this time since topological data structures must form the basis of any mapping system before one even considers adding to it any amount of intelligence or intuition.

During the First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, which was held in 1978, a number of ideas were presented which show the importance that topology has assumed in the field of cartographic representation. (Ed. Dutton, 1978). Topology is concerned with spatial properties, that is, the relationships or ways in which objects in space relate to one another, which is certainly applicable to every point, line, area, surface and of course, to their color on a map.

As part of a computerized system, a data structure creates a model of spatial knowledge which determines the ability to deal with the range of geographic questions. Since cartography is more a science of representation than of measurement, topological structures in which objects perform more of a linking function to other objects are a necessity. Objects may belong to the same or to other dimensions. It is important to note, as N. Chrisman mentioned, that a basic measure of efficiency for geographic information processing systems is the ratio between the information needed and the information stored. That means that a structure may well serve more than one function. A shared border defined once but used twice, for example, is a good example of such efficiency, as most spatial problems do not truly need access to all entities at once. Another is defining a given straight line segment by storing only the two end points of the segment. This type of economy in structures, information stored and procedure used, is necessary to make these kinds of systems manageable and relatively easy to use. As David Sinton mentioned during the same symposium:

> When analysing problems associated with the planning, management, and design of our natural resources it is usually necessary to compare data from several different sources which has been generalized in a mapped thematic form. Graphically, this is a relatively simple problem of draftsmanship. However, in a digital environment this becomes an extremely complex problem of geometrical analysis."..."Elegant technical solutions and sophisticated analysis procedures are not the end product of a geographic information system. The success

of a geographic information system should be measured in terms of the extent to which it provides useful information to a diverse group of users." (Sinton, 1978, p.14-15).

Cartographic data structures have progressed considerably since the first use of digitizing machines made the data collection process relatively easier. Research is still under way to solve some of the main problems associated with cartographic information: the direct use of the input produced by the digitizer without further internal manipulation which could be beneficial to the development of intelligent programs; the lack of combination of different types of cartographic information into one file necessitating the search of various files for various features; the lack of identification of neighborhood entities, the spatial correlation so necessary in most manipulations of geographical data. (Peucker and Chrisman, 1975).

The first type of data base system for planar surfaces is the encoding, entity by entity, with no regard for adjacencies or overlaps. When this data is drawn on the screen, it can suffer from the problem of sliver lines since a point may not have identical coordinates in all recordings. Recording points in duplicate or even triplicate also uses a lot of storage. Data dictionaries may eliminate those sliver lines, but the resulting data still does not provide any information on neighborhood relationships which must be computed as necessary over and over during the processing itself. By adding the results of a topological neighborhood function to a data structure, flexibility and subsequent improvements in the ease of applications can be realized. One of the first known attempts to incorporate topological structures into a geographic data base was the DIME (Dual Independent Map Encoding) system developed by the U.S. Bureau of the Census. In this system, the basic element is a straight line segment defined by two endpoints. The segment has two node identifiers along with the coordinates of its two end points and codes for the polygons on each side of the segment. However, here again, neighborhood relationships are not made explicit, necessitating laborous search procedures to find neighbors and assemble the outline of a polygon. Its main importance lies in the fact that the topological structures are made explicit, paving the way for other systems, and that it has proved to be a good topological error detection system for the use which it was intended, address coding and matching in metropolitan areas during the 1970 census.

A more well known data structure is in the POLYVRT system. (POLYVRT User's Manual, 1974). A POLYVRT chain has nodes at both ends but may have an intermediate amount of points in between. The chain plays a topological role in the subsequent construction of a list data structure. It has an identification number, a length (number of points), the start point and the end point, as well as the identification codes of of the polygons to the left and right of it. These are input directly from the user. The point information is in secondary storage and is accessed through pointers from the main chain information. A separate list allows linkage from polygon to chains and vice-versa. To handle M polygons, core storage is needed for 3M chain records and 3M node records. POLYVRT is augmented by many procedures which allow for the elimination of internal points without influencing the neighborhood relationships, the compressing out of chains which no longer serve to define the topology of the simplified file, the restructuring of input and output files, insertion, deletion, replacement, updating and generalization. POLYVRT is important because of its use of topological data structures and because it can accomodate input from various existing systems such as the World Data Bank, Dime, Calform etc.. Even though the documentation describes it as user-oriented, the command language is still quite difficult to learn and its fixed field format must be well understood by the user. The procedures are also often self-generated as a chain without returning to the user for confirmation. POLYVRT's strenghts therefore mainly reside in the use of topological data structures and in file conversion capabilities.

When dealing with structures for three-dimensional surfaces, certain assumptions about surface behaviour are necessary as first suggested by T. Poiker in 1972. (Peucker, 1972). One is that the surface retains the value of a data point within the neighborhood of that data point, whether the neighborhood is a polygon or the closest area to the data point. Another is that a point is a value on a constantly changing surface. Neighborhood is then a number of closest neighboring data points, with intervening values being interpolated. Finally, since that surface may contain errors, the data point may be located close to and not on the surface. This also implies that the actual surface is smoother than the reconstructed surface. With these assumptions, data structures can be developed which can handle both two-dimensional and three-dimensional types of surfaces.

Some proposed data structures include a root object which is an area uncut by further partitioning and which is called the Least Common Geographic Unit. (Peucker and Chrisman, 1975). These LCGU would be constructed in the same way as a POLYVRT polygon directly from chains and have a hierarchical relationship to other polygons. Cross-referencing would allow the determination of nested objects. The success of this structure would depend on appropriate topological manipulation routines. Neighborhood search routines would create records of neighbors for every point or line or polygon at any specified depth of neighborhood. Neighborhood relationships can be established by triangulation; features of high geographical information content can be defined by special nodes; after triangulation and generalization of the surface, a linked list of surface points is established and portions of the data base can then be brought into memory as required by the processing.

The central idea of these proposed structures is to separate the data base from the application programs with the data structures becoming the link between the two. This is, in fact, what any intelligent system that has to process a large amount of information data will have to do and, like the human brain, classify information received, reorganize and retain what is most important and can be of immediate use and keep "pointers" to where the rest of the information is or how it can be found if the need ever arises to do so.

APPENDIX G

EXPERT-SYSTEMS

GENERAL CONCEPTS

> In most areas of artificial intelligence - that branch of computer science that attempts to have machines emulate intelligent behavior - programs fall far short of the competence of humans or even animals. Computer systems designed to see images, hear sounds, and recognize speech are still in fairly early stages of development. However, in one area of AI - that of reasoning from knowledge in a limited domain - computer programs can not only approach human performance, but in some cases they can exceed it. (Gevarter, 1983, p.39).

This rather strong statement can be considered controversial, as expert-systems have recently come under attack from several sources with regards to their performance, their speed, their limitations and some of the inferences produced. Designed to simulate the behaviour of a human expert in similar situations, expert-systems are nonetheless programs which use a collection of facts, rules of thumb and other knowledge about a given field, coupled with methods of applying those rules, to make appropriate inferences, and, as all programs, they are subject to the same pitfalls. Algorithms used and mostly the database or knowledge base of facts and rules that they work with must be carefully planned, properly designed, well organized and appropriately tested for these systems to behave correctly. Even the meaning of "correctness", whether considered from a purely logical standpoint, common sense or reasonable behaviour, has given rise to various debates and conclusions.

To summarize in a few words the beginnings of such systems, one only has to recall some of the earlier well-known artificial intelligence systems which have paved the way for more specialized and sophisticated expert-systems. They include the SHRDLU "blocks world" program of Terry Winograd, which is one of the first programs to combine natural language understanding with the simulation of action; the DOCTOR program written for the ELIZA system, by Joseph Weizenbaum, which simulates the dialogue of a psychiatrist with his patient so well that many people took it seriously and started confiding in the system; and Patrick Winston's concept-learning program which learns about "arches" from a series of examples. (Winston, 1979).

In an interview given <u>Psychology</u> <u>Today</u> , (Kendig, 1983), Roger Shank differentiated between "expert-systems", which are rule-based and are used to model very complicated technical knowledge and "advisory systems", which can understand questions and consult a large database of facts to extract the proper information. Advisory systems have no cognitive reasoning, while expert-systems can model a large percentage of the expert's reasoning process. There is, however, a small percentage missing which is often called intuition, hunch, experience, sixth-sense, or simply expertise from having been in a similar situation before and having learned from observation and trial and error. This is where rules can sometimes fail, not because they are incorrect, but rather because the ingenuity and the speed at which the expert can assess the situation and make the appropriate decision, even when faced with incomplete facts or conflicting information, cannot always be equated. Dealing with incompleteness and data which, on the surface, may appear conflicting, is an area still under study by many AI and Database researchers. (Levesque, 1983).

So much has been written on Artificial Intelligence systems that attempting to discuss and analyse it all would be a major work in its own right. The following, therefore, only represents a brief overview of the major trends and achievements on the basis of the various classification categories of knowledge bases and procedures used. Such an overview is useful to understand the choices made in the creation of the specific system dealing with Graphic Design with Color, discussed in part two of this document, and it is with this in mind that the frameworks and ideas of knowledge representation and procedural rules present in actual expert-systems are presented.

Expert-systems have several characteristics common to all of them. First, they include heuristics for reasoning with judgemental knowledge as well as with formal knowledge of established theories. Heuristics can also be referred to, in much simpler terms, as "tricks-of-the-trade". When designing a system to perform a certain task, many questions have to be asked of the form: "What would the expert do in a case like this?". For example, a mechanic doing some troubleshooting on a car does not need to check all the joints and hoses if he finds an obvious leak in the radiator. In looking for a specific lake on a map, one goes directly to areas colored blue; no need to check the names of cities or red colored polygons since it would be considered an error if lakes had been colored red. These kinds of heuristics, where to start a search, when to stop, what to try first, are the kinds of procedures which differentiate an "intelligent" system from one which repeats the same pattern in all circumstances and make it appear to "judge" the situation.

Transparency, flexibility, extendability and simplicity are other characteristics useful to any expert-system. The user can then see the system behave in a simple, uncomplicated way, hiding its complex manipulations under a dialogue which uses every day expressions and clear questions. This is not to say that the design of the system is simple but rather that, if well done, it will appear simple to the user. The flexibility of the system is usually also a question of language as well as adapting to circumstances through various heuristics. A system which can perform the same task given various types of input, or user options, would be considered flexible. Extendability is closely linked to simplicity of knowledge representation. Various forms of knowledge representation will be discussed in the following paragraphs, but for the moment, let us consider as example the the PL/1 structure format or the Pascal record format. These structures, if well designed, are essentially simple to add to. Conceived in a hierarchical manner, adding a field and accessing that field is not a complicated task and can be done uniformely for all fields. The complexity resides in the initial design.

Finally, since it is necessary for anyone looking at the expert-system to be able to determine which items are present by inference and which are not, explicitness in the knowledge base is highly desirable. This requirement is not really different from examining any data base of facts, with the exception that "intelligent" systems often contain more "facts" than can be seen at first glance. These facts can be hidden in procedures and may not be stored as data but are rather inferred by the system during the computing process. Deduction mechanisms may also introduce new facts. Explicitness is therefore the quality by which such inferences may be made visible either through the form of the representation used or through proper documentation.

The architecture of expert-systems is generally composed of three entities: a knowledge base, an inference engine and a user interface. The knowledge base contains ground facts, implied relations through its organization, as well as rules which will be used by the inference engine, or procedures, to analyse, synthesize and solve problems, and which will be communicated to the outside world through the interface modules. For example, the architecture of a system called MYCIN, a fairly sophisticated expert-system which analyses medical data about patients with severe infections, is composed of four main entities: a database of facts known or inferred about specific cases; a knowledge base of rules and facts covering specialized knowledge of meningitis; a set of inference rules or procedures made up of Interlisp functions which controls the interaction and updates the current state of knowledge about the case at hand; and an interface program which translates the user's input into machine readable form and converts the machine's answer to an understandable English output for the user. (Buchanan and Duda, 1983).

In addition, the EMYCIN system, the framework on which MYCIN is based, contains an explanation subsystem to answer questions about consultation or about the static knowledge base. It also contains a knowledge base editor to aid in the construction of new knowledge bases (new systems) and to aid in debugging an emerging knowledge base. The diagram on the top of the next page helps to illustrate the architecture described. (Buchanan and Duda, 1983).

As can be noticed, a good part of the behaviour of an expert-system is due to its knowledge and more precisely how the information is represented in the knowledge base to facilitate access. A number of knowledge representation methods have been used, of which the following paragraphs present a general overview. Knowledge representation is closely linked to a new concept called "knowledge engineering". The main difference appears to be that knowledge representation is the way by which the information is actually represented in the knowledge base while knowledge engineering is more concerned with the process leading to proper and accurate representation. Knowledge engineering is no small task. It involves acquiring the required

EMYCIN's ARCHITECTURE:

```
expert's           knowledge base       customized
rules   ......     editor and      ...  knowledge
                   debugging aids       base
                                         .
                                         .
                                         .
user's             interaction          problem          advice
description ...    handler        ..... solver     ....  for this
of new case                              .               case
                                         .
                                          database
                                          of facts
                      and                 about this
                                          case
                                          .
user's    .....    question       ......  explanation .. explanation
questions          answering              subsystem
                   routines

..............................................................
```

Figure G.1   EMYCIN's architecture.

knowledge from experts and reworking it in a form fit for machine consumption. Knowledge engineering basically consists of four stages which are closely related to present business information system design. (McDermott, 1983) The first stage, task definition, involves defining the role of the expert-system in the same way that a business systems analyst defines the parameters of the system he is attempting to design. The second stage also proceeds in a parallel fashion with MIS (management information system) design, that is, an initial design must be made taking into account representation and extraction of the knowledge. The third stage consists of a kind of acceptance test where earnest knowledge extraction is simulated. The expert asks questions and if the answer is incorrect, he teaches the new knowledge or reasoning process to the system. The fourth stage, in analogy with the maintenance phase of an information and management systems implementation, is the continuous updating of the knowledge base and the artificial intelligence inference rules by incorporating the user's changing needs and new developments in computing science.

KNOWLEDGE REPRESENTATION

Knowledge representation is no easier, mainly because:

> In contrast to conventional database systems, AI systems require a knowledge base with diverse kinds of knowledge. These include, but are not limited to, knowledge about objects, knowledge about processes, and hard-to-represent commonsense knowledge about goals, motivation, causality, time, actions, etc.." (McCalla and Cercone, 1983, p.12).

Several different approaches to knowledge representation have been used, the most important being based on semantic networks, frames, logic and production system rules.

Systems based on semantic networks link objects to other objects through a semantic relationship. The most typical relationship is called the "IS_A" link. (Brachman, 1983). The "IS_A" relation is roughly a hierarchical partial ordering of universally quantified conditional objects. There are various types of "IS_A" links. The following example expresses the types of relationships possible with this type of link:

```
                    mammal
                       .
                       .
      ......................................
      .                                    .
large                                small
      .                                    .
   ...................              .............
   .                 .              .           .
elephant          whale            dog         deer
   .
  .................
  .                .
  african          indian
  .                .
  ... cyl.trunk ....
                   ....clyde
                   ....prince
```

Figure G.2   Semantic network.

Generic/Generic relations are of the type where two generic entities are related but one is more general than the other. One can be considered a subset, such as an african elephant is a member of the elephant class. Elephant is taken as a generalization of african elephant. Another relationship expressed is the expression AKO (a kind of), where an elephant is seen as a kind of mammal, which has a trunk of the form of a long cylinder and is included in the concept of large mammals. Conceptual relations can also be derived, whereby an elephant is associated with the characteristic functions of the set of all elephants. Generic/Individual relations can be considered as a description of an individual by some general characteristics, whereby the individual is an "instance" of the generic term. This relationship can show membership in a set as well as predication. If Clyde is an elephant, then Clyde is a member of the set of elephants and IS_A expresses the fact that Elephant(Clyde). The generic description can be used to construct individual descriptions such as using "elephant" to construct the assertion "Prince is the Royal Elephant", or to create natural language abstraction in sentences such as: "In view of the rapid encroachment of civilization on its natural feeding ground, the elephant must now be protected against extinction", where "the elephant" represents the generic type "elephant".

The IS_A link, which was at the beginning a very simple and useful tool, has evolved considerably over the years, bringing more interpretation problems in the wake of its increasing complexity. Brachman therefore suggests that a breaking down of the "IS_A" relation into simpler components might be the the path to future developments in semantic networks.

> While semantic nets have prospered as a framework for knowledge representation, their keystone construct - the IS_A link - has wavered considerably in its interpretation. IS_A has been used principally to form sentences that could be asserted, particularly sentences with a default import. However, IS_A has been used to mean many other things, making comparisons between networks and logic all but impossible. The analysis presented in this article indicates that things might be a lot clearer if IS_A were broken down into its semantic subcomponents and those subcomponents then used as the primitives of a representation system. (Brachman, 1983, p.36).

Frames are basically a modular decomposition of the knowledge into chuncks often referred to as schemata.

> Frames are particularly useful when used to represent knowledge of certain stereotypical concepts or events. When one of these standard concepts or events is recognized, slots (frame variables) inside the appropriate frame can be filled in by tokens representing the actual actors or actions. After this step, much "precompiled" knowledge can be gleaned directly from the frame." (McCalla and Cercone, 1983, p.14).

A complex example of familiar objects presented by Winston (Winston, 1979, p.186), easily explains the basic idea behind this type of knowledge representation.

Frames can be used procedurally or simply as "scripts" with no capacity for inferences. They also present some difficulties in handling incompleteness. Debates on the use of frames as procedural tools or declarative structures have been going on for some time and are beyond the scope of this work. Reference can however be made to Terry Winograd's paper entitled "Frame Representations and the Declarative/Procedural Controversy", (Winograd, 1975), which examines the various criteria for judging representations and attempts to reconcile the best features of both views.

Logic-based systems are usually linked to the programming language Prolog which is based on first-order logic. Knowledge is represented through predicates for asserting facts and building rules from which inferences are derived. (Dahl, 1983). Predications can represent simple things in our universe, i.e. Likes(Sweetie,Rover) means that Sweetie likes Rover, or general

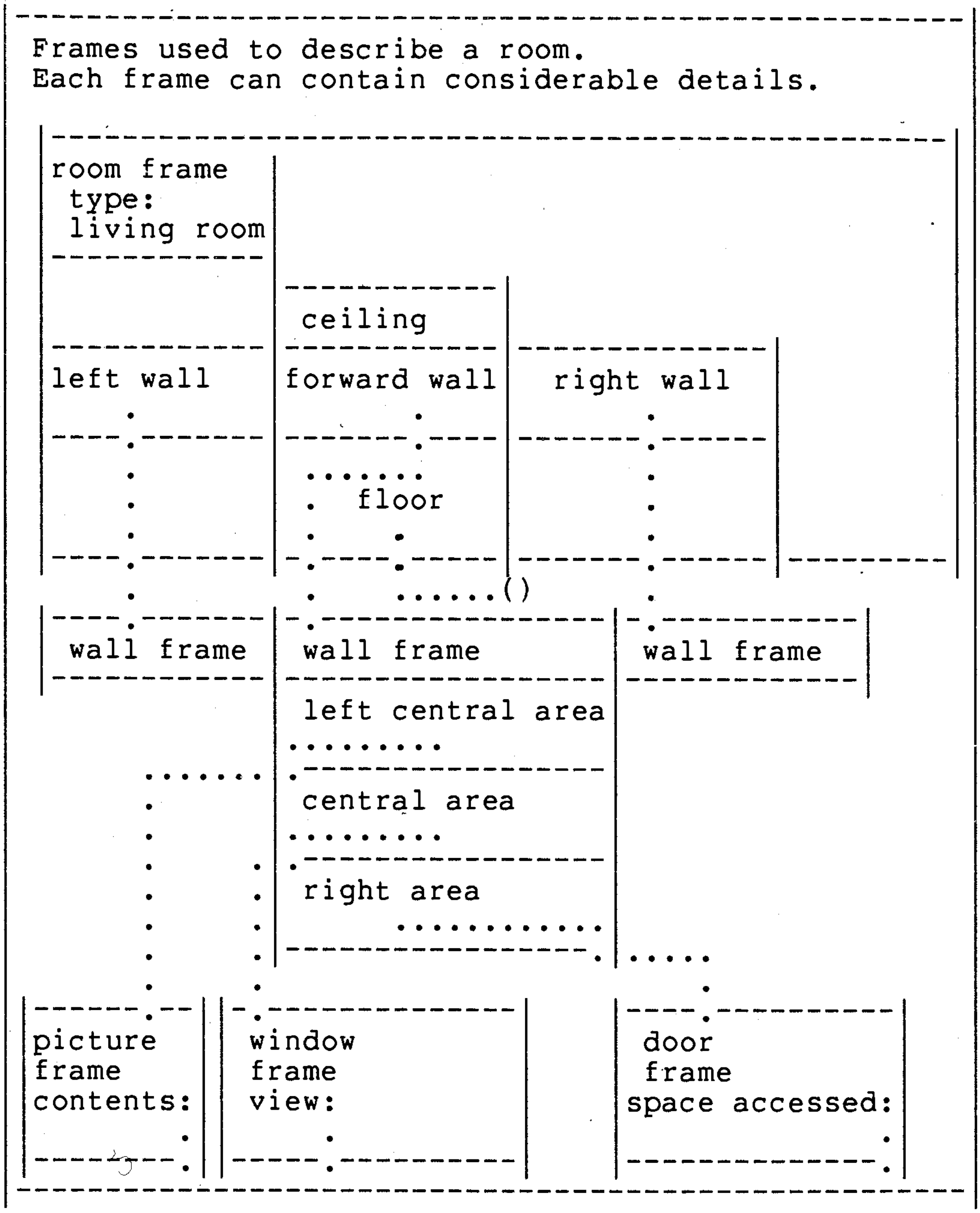


Figure G.3   Frames.

concepts such as Likes(mother(x),x), which means that everyone is liked by his/her/its mother.

Rules, on the other hand, have the general form: P1 if (P2 and P3 and ... and Pn), where Pi stands for predications and the parentheses are generally omitted. For instance, the general rule that Rover likes everyone who likes him can be stated: likes(Rover,y) if likes (y,Rover). Prolog will take a rule as stating that if all the conditions P2,P3, ... Pn hold, then the conclusion P1 holds. If the rule contains any variables, it is taken to apply for all their possible values." (Dahl (Computer), 1983, p.106).

After having defined a set of facts and rules, Prolog can check if our assumptions about the world are consistent with the description given to it by answering queries of the form: Likes(Sweetie,Rover)? to which it would answer "yes" or "no". It can also find and list all individuals who meet a certain requirement through the use of undefined variables. To the question: Likes(Rover,z)? it would answer by substituting for "z" all the individuals that Rover likes. Prolog rules can get quite complicated and skillful programming is necessary to avoid infinite loops which are easier to get into than with other procedural programming languages. Another feature of Prolog, backtracking through options, has obvious advantages since it allows the simulation of intelligence; if an option fails, try the next one. It also has the disadvantage of being capable of repeating this process for a very long time, leading to slow response time. Knowledge engineering processes therefore play a very important role in building this type of system and questions such as what options should be tried in what order, proper stopping criteria and a good general organization of facts and rules to suit the purpose in mind, have to be seriously considered during the design stage. These logic-based systems are said to be goal-driven, because their main control strategy is to start from a goal and derive sub-goals from it. As a computer language and as a control mechanism, Prolog has proved to be a useful tool for knowledge representation. Procedures and data having the same format, it offers the obvious advantage of consistency.

> Applications to expert knowledge also resulted in useful extensions that are now being incorporated into Prolog itself, e.g., dynamic query reordering on the basis of statically defined priorities for predications (in view of efficiency, among other things), and the addition of a self-constructing predicate." (Dahl, (Computer), 1983, p.111).

However, the top-down, depth-first search approach with backtracking that Prolog uses is not suited to every type of expert-system. An example of this is geological measurements of bore holes aimed a producing oil. Since in the real world, log interpretation from these measurements is based on bottom-up or forward inferences, the formal logic of Prolog would not suit well an expert-system created to deal with this kind of problem. (Buchanan and Duda, 1983). Another aspect to consider before using logic-based systems is the difficulty of the language used.

> The languages of mathematical logic were not meant for general use. Their developers did not claim they were universal symbolisms for unrestricted application. ... Of course, that these formalisms were not devised with the goal of expressing commonsense knowledge, that is, of solving the AI problem, does not mean that they cannot be used in this manner. (Israel, 1983, p.38).

Natural language parsers, another main project in its own right, may eventually bring formal logic expressions and common sense knowledge to a workable mutual communication level.

Nonetheless, logic-based systems are likely to be used more and more as fifth generation computers appear, since these computers are based on Prolog and have hard-wired associative memories and database machines coupled with a Prolog inference engine. Database machines have shown some improvement in performance over conventional database systems. (Schuster, 1979).

Finally, last but not least, are production systems, usually based on rules of the IF-THEN-ELSE type. There are several implementations of this type of system in existence and their popularity may be due to the fact that they relate well to how people usually solve problems and to most computer languages in existence. The data present in the data base must satisfy the conditions of the left-hand side of the rule; this approach is often referred to as data-driven control. A good example of a rule-based production system is the expert-system whose architecture has been described previously, namely the MYCIN system. (Buchanan and Duda, 1983).

Although MYCIN uses the INTERLISP language, which represents data as lists, any other formalism such as arrays, structures or records, or even strings may be used. The rules have the form: IF <CONDITION> THEN <ACTION>, familiar to any programming language. At the general level, production systems are capable of arbitrarily complex behaviour. In practice, however, controlled backtracking strategies and the search and maintenance of detailed records of interdependencies between the facts contained in the data base may be necessary.

> Many of the expert systems construed to date have controlled this complexity by sacrificing the ability to perform general problem-solving tasks. They have achieved their competence by specializing, by exploiting the faillible but effective heuristic methods that human experts bring to a particular class of problems. (Buchanan and Duda, 1983, p.177).

In the EMYCIN system, of which MYCIN is a sub-system, facts are depicted as associative triples, that is triples of the type attribute-object-value with a certain weight factor called degree of certainty. The syntax looks like: The (attribute) of (object) is (value) with certainty (CF). The interpreter uses a control strategy to decide what rules to apply, which can be totally internal or it may also go to the user for help or further information. It uses deduction, matching and bactracking mechanisms. Gevarter gives the following example of an interaction with MYCIN.

> Mycin, an expert system for medical diagnosis, works on the assumption that a possible disease exists for the case at hand and then tries to find evidence that supports or contradicts its hypothesis. It gathers this evidence from the physician working on a case, starting with the initial choice of a possible disease. An example of dialogue follows:
>
> MYCIN: What is the infection?
> PHYSICIAN: Endarteritis [This is an inflammation of the inner layer of an artery and can be caused by a number of agents.]
> MYCIN: Is organism-1 a rod or a coccus? [This question will narrow the range of possible agents for the disease.]
> PHYSICIAN: Rod.
>
> Mycin then asks additional questions about laboratory test results to help determine what organisms might be causing the disease. When it has reached its conclusions, it will recommend a particular drug therapy to combat the organism or organisms involved. So that

the physician will trust Mycin's conclusions, the program is designed to answer questions about its reasoning. For example:

PHYSICIAN: How did you decide that organism-1 is E. Coli?
MYCIN: I used rule 084 to conclude that organism-1 is E. Coli. This gave a cumulative certainty factor of 0.51. [The physician can question further, either by looking at the text of the rule involved, or by questioning its antecedents.] (Gevarter, 1983, p.41).

The above schemes point out that what is really important in knowledge representation is what W.A. Woods calls "expressive adequacy" and "notational efficacy". (Woods, 1983). Highly simplified, these two expressions mean that the representation should adequately represent what we want to express as well as being in a format that the ordinary humans can understand and that is also efficient for the computer to process. Isnt't this close to the conception one has of a true expert?

Putting together an expert-system which attempts to combine the efficient fact retrieval mechanisms of database systems and the deductive powers of artificial intelligence systems requires that all the above-mentioned criteria and many more key notions be incorporated in a coherent whole.

Summarizing the ideas presented so far in this chapter, we can come to the following conclusions as to when an expert-system is desirable and what needs to be considered. First, the problem's scope must be rather narrow, a human expert must be available to give to the system the knowledge required, human "experts" should agree among themselves about the facts and rules, extensive test data must be available and measurable milestones be established to determine how well the system works at any given point. Secondly, task-specific knowledge must be separated from the rest of the program, great attention be given to details, and the data structures used must be uniform to facilitate manipulation and knowledge acquisition. Thirdly, the inference structures should include symbolic reasoning, a combination of deductive logic and plausible reasoning, explicit problem-solving strategies and interactive user interfaces. The system should also be able to inform the user of the content of its knowledge base. In certain cases, such as for diagnostic systems, it must also keep track of its line of reasoning to answer further queries. Its structures should be made to accommodate the acquisition of new knowledge. It should perform well on test data and a static evaluation of its knowledge base should be made periodically to confirm its validity and consistency, especially when users are allowed to make certain changes dynamically to it. (Buchanan and Duda, 1983).

CARTOGRAPHIC APPLICATIONS

In considering an expert-system for a particular area of application, such as cartography, one must make the difference between the knowledge brought by different types of people. The system designer is usually a computer scientist who creates the knowledge base, the procedures and heuristics, based on computer capabilities, and simple yet efficient internal representation. He must be able to explain the domain in a form understandable to the computer and to the user, develop routines for interactive communication with the knowledge base and procedures for internal manipulation which will facilitate the total process. The system user, on the other hand, who may be a cartographer or any other map user, will use the system to create maps and to exchange information with the system, imparting to it his knowledge in certain cases, and looking to the system for help in others. In both cases, the knowledge will stem from different sources and consider different aspects as important. For example, in a system dealing with color, the system will most likely not produce orange lakes glittering with the last rays of sunset and the user should not expect it to. But if the artistic knowledge embodied in the system can match the level of scientific reasoning the user can put up with, the result can be a satisfactory compromise which can prove of invaluable help.

Intelligent cartographic systems can in fact bridge the gap which exists between the traditional and the computer cartographer, unifying and enriching the discipline as a whole. Some systems which attempt to play the role of expert-systems or intelligent systems in part of their design were discussed briefly during the First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems. (Ed. Dutton, 1978). In the whole, these systems are still conventional in their file handling and programming language (Fortran) capabilities. The sheer amount of information they manipulate and their high costs necessitated this approach. However, a brief overview of what they do is given here as they have advanced considerably the field of computerized geographic information systems.

Hierarchical descriptions using a special map language have been attempted but are not yet implemented. (Youngmann, 1978). This type of syntactic description uses the polygon as a primitive and, modelled on Chomsky's type of grammar, describes geometric and logical structures through relations and composition rules. The particular action that occurs whenever a relation is invoked does not depend upon the relation but upon the rule with which it is used. Rules of composition are used to generate statements and to analyse them. In this kind of proposed system, data structures are fused with processing rules, Prolog style, to yield a special language of map description and manipulation.

A cartographic system which assumes some of the roles of an expert-system is called CARTE. (Wood, 1978). It is an interactive display of polygonal data which is in fact a stand-alone sub-system for entity driven mapping. Since it is only a part of a larger system, it expects as input files the formatted output of two other sub-systems, MAPEDIT which gives the boundary points and centroid of the polygon, and DOBEDO which prepares a data base. All these sub-systems form the polygonal mapping system called SEEDIS, which stands for (Socio-Economic-Environmental-Demographic Information System), written in Fortran, and presently in use at the Lawrence Berkeley Laboratory. The CARTE sub-system has the ability to do insertions, change legends, define a shading symbolism such as parallel or cross-hatched lines and can also display text. As a whole, the system is large and structured but its capabilities at this sub-system level are interesting as a hachuring procedure for shading could complement well a map coloring system.

An image based geocoding system for correlating satellite and topological data bases has also been discussed, its main aim being primarily to provide variable aggregation of data and a method for spatial arrangement which can interface with mathematical and statistical programs to aid in the analysis of spatially oriented data. (Bryant, 1978). The main benefit of the system is the use of the image raster as spatial data representation, which avoids a lot of file searching and updating during insertion and overlay operations. Some of the difficulties encountered in the system are the communications between ordinal representation of coded data and a nominal referencing procedure and the lack of available procedures in the image processing field to address the specific requirements of a geographic information system. Many other systems dealing directly or indirectly with geography are also in use, such as the geological expert-systems mentioned above, and the field is still growing in this very progressive and vast area of expansion.

Another interesting geographical system, discussed in the special October 1983 issue on Knowledge Representation of

Computer magazine, is the MAPSEE system which interprets freehand geographical sketch maps. (Havens and Mackworth, 1983).

> In this world, image lines or chains can be scene roads, rivers, bridges, mountains, towns, lakeshores, or seashores, while image regions can be land, lake, or ocean. The constraint approach uses these entities as the objects to be instantiated, while the models are derived from scene domain knowledge of how the objects can interact. For example, a T-junction of two image chains could be a road junction or a river junction or a river going under a bridge, etc.. The models are thus n-ary constraints on the objects, and the network consistency algorithms are generalized to cope with that extension. (Havens and Mackworth, 1983, p.91).

Some of the shortcomings of the first version of this system concerning descriptive adequacy issues and procedural adequacy issues have been conquered in MAPSEE2 by the use of a knowledge representation method based on a network of schema models, each model representing a class of objects, providing a description of the generic properties of every member of the class and specifying the possible relationships of the class with other schemata in the network. The composition of the schemata is hierarchical, where each node is a schema class and the arcs between nodes depict relationships between schemata. Havens and Mackworth describe in some details how these shemata overcome each of the problems associated with the first version of MAPSEE. It is interesting to note that for visual processing, a combination of data-driven and goal-driven methods of control mechanisms, mentioned in the previous paragraphs, proved to be the best method. These mixed strategies seem to attempt to get the best of both worlds by alternating between them. PROSPECTOR, which is an expert-system that analyses geological data to aid in mineral explorations, also uses this method and seems to be doing well. (Buchanan and Duda, 1983) (Reiter, 1983).

For the time being, the major difficulty in constructing an expert-system still remains engineering the knowledge that experts use into a form that is usable by the system. Every problem area and every expert is unique. Knowledge extracted from experts, from large data bases, from experience and even from intuition needs to be incorporated into a coherent whole. Expert-systems are well-suited for analysis, synthesis, classification and diagnostic types of problems. They can be used for consultation, to acquire knowledge and even to train people in their field of expertise and problem solving skills. By being able to cope with the ambiguity that incomplete data or redundant data creates, using overlapping contributions to compensate for incompleteness, and by exploiting the redundancy of expert rules as protection against spurious data or inappropriate heuristics, the expert-system can make valuable contributions to solving problems of both commercial and scientific importance. Data communication advances can further increase the power of these expert-systems by allowing them to give copies of their knowledge to other systems and to unify expertise about several specialties. Human prepared documentation when simple and complete, can greatly help in achieving this goal. As the technological framework is developed and made more and more efficient, experts will be able to concentrate more on the "knowledge engineering" aspect, making these expert-systems very powerful computational tools.

REFERENCES

Agoston, G.A., Color Theory and its Application in Art and Design , Springer Series in Optical Sciences, Vol.19. Ed. David L. MacAdam, Springer-Verlag, Berlin, 1979.

Baldwin, Lee, "Color Considerations", Byte Magazine, September 1984.

Bertin, Jacques, Semiologie Graphique , Ed. Gauthiers-Villars, Paris, 1967.

Billmeyer, Fred W. Jr. and Saltzman, Max, Principles of Color Technology , Second Edition, John Wiley and Sons Inc., New-York, 1981.

Brachman, Ronald J., "What IS_A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks", Computer , Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

Brown, A., The Use of Color in Cartography , ITC, 1982.

Bryant, Nevin A. and Zobrist, Albert, "An Image-Based Information System: Architecture for Correlating Satellite and Topological Data Bases", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume four - Data Structures: Hierarchical and Overlaid, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Buchanan, Bruce G. and Duda, Richard O., "Principles of Rule-Based Expert-Systems", Advances in Computer , Marshall C. Yovits, Academic Press, Vol.22, 1983.

Cercone, Nick and Goebel, Randy, "Knowledge Representation and Data Bases", Technical Report ,No. 81-15, Computing Science Department, Simon Fraser University, 1981.

Chrisman, Nicholas R., "Concepts of Space as a Guide to Cartographic Data Structures", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume seven - Spatial Semantics: Understanding and Interacting with Map Data, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Clarke, Michael, "Computer-Generated Imagery", Compuwest , June 1984.

Clifford, Jim, Jarke, Matthias, Vassiliou, Yannis, "A Short

Introduction to Expert-Systems", A Quarterly Bulletin of the IEEE Computer Society Technical Committee on Database Engineering , Vol.6, No.4, December 1983.

Dahl, Veronica, "Logic Programming as a Representation of Knowledge", Computer , Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

Dahl, Veronica, "On Logic-Programmed Representations of Knowledge", Technical Report , No. 83-1, Computing Science Department, Simon Fraser University, 1983.

Ehlers, H.-J., "How Colour can Help Visualize Information", Proceedings of the 6th International Online Information Meeting , London, December 1982.

Fegeas, Robin G., "The Graphic Input Procedure - An Operational Line Segment / Polygon Graphic to Digital Conversion", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume seven - Spatial Semantics: Understanding and Interacting with Map Data, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Furukawa, K. and Fuchi, K., "Knowledge Engineering and Fifth Generation Computers", A Quarterly Bulletin of the IEEE Computer Society Technical Committee on Database Engineering , Vol.6, No.4, December 1983.

Gevarter, William B., "Expert Systems: Limited but Powerful", IEEE Spectrum , August 1983.

Ghezzi, Carlo and Mehdi, Jazayeri, Programming Language Concepts , John Wiley and Sons Inc., New-York, 1982.

Goldstein, E. Bruce, Sensation and Perception , Wadsworth Publishing Company, California, 1980.

Havens, William and Mackworth, Alan, "Representing Knowledge of the Visual World", Computer , Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

Hsiao, David K., "Data Base Computers", Advances in Computers , Marshall C. Yovits, Academic Press, Vol.19, 1980.

Imhof, Eduard, Cartographic Relief Representation , Ed. H.J.Steward, Walter de Gruyter, Berlin, 1982.

Israel, David J., "The Role of Logic in Knowledge Representation", Computer , Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

Keedy-Nelson, Geometry, A Modern Introduction , Second Edition, Addison-Wesley Publishing Company, 1973.

Kendig, Frank, "A Conversation with Roger Shank", Psychology Today, April 1983.

Kowalski, Robert, Logic for Problem Solving, The Computer Science Library, Artificial Intelligence Series, Editor Nils J. Nelson, New-York, 1979.

Land, Ewin H., "The Retinex Theory of Color Vision", Scientific American, December 1977.

Levesque, Hector J., "Incompleteness in Knowledge Bases", Proceedings of ACM SIGMOD, 1983.

Libby, William Charles, Color and the Structural Sense, Prentice-Hall Inc., N.J., 1974.

MacDougall, Bruce, Computer Programming for Spatial Problems, Edward Arnold Publishers Ltd., 1976.

McCalla, Gordon and Cercone, Nick, "Guest Editors' Introduction: Approaches to Knowledge Representation", Computer, Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

McDermott, J., "The Knowledge Engineering Process", A Quarterly Bulletin of the IEEE Computer Society Technical Committee on Database Engineering, Vol.6, NO.4, December 1983.

Meisters, G.H., "Polygons have ears", American Mathematics Monthly, June-July, 1975.

Monmonnier, Mark S., Computer-Assisted Cartography Principles and Prospects, Prentice-Hall Inc., N.J., 1982.

Muller, Joseph-Emile, Klee Magic Squares, Tudor Publishing Co., New-York, 1957.

Nassiers, Kurt, "The Causes of Color", Scientific American, October 1980.

Newman, William M. and Sproull, Robert F., Principles of Interactive Computer Graphics, Second Edition, McGraw-Hill Computer Science Series, McGraw-Hill Book Company, New-York, 1979.

Peucker, Thomas K., Computer Cartography Commission on College Geography Resource Paper No.17, Association of American Geographers, Washington, D.C., 1972.

Peucker, Thomas K. and Chrisman Nicholas, "Cartographic Data Structures", The American Cartographer, Vol.2, No.1, April 1975.

Poiker, Thomas K., "Principles of Data Structures for Computer Cartography", unpublished.

Poiker, Thomas K., "The Use of Computer Science and Artificial Intelligence in Cartographic Design", unpublished.

Poiker, Thomas K., "Some Problems for the Use of Artificial Intelligence in Computer Cartography", October 1981, unpublished.

Reiter, J.E., "A Review of the Prospector Project", A Quarterly Bulletin of the IEEE Computer Society Technical Committee on Database Engineering , Vol.6, No.4, December 1983.

Robertson, P.J., "A Guide to Using Color on Alphanumeric Displays", Technical Report , IBM U.K. Laboratories Limited, 1980.

Samit, Morris L., "Color Basics for Computer Graphics", Harvard Computer Graphics Week , Harvard University Graduate School of Design, 1982.

Schuster, S.A., Ozkarahan, E.A. and Smith, K.C., "RAP An Associative Processor for Data Base Management", Proceedings of AFIPS NCC , Vol.44, 1975.

Schuster, S.A., Ozkarahan, E.A. and Smith, K.C., "RAP.2 An Associative Processor for Databases and Its Applications", IEEE Transactions on Computers , Vol. C-28, No.6, June 1979.

Shamos, M.I., Computational Geometry ,(Ph.D. Thesis), Yale University, 1977.

Sheppard, Joseph J. Jr., Human Color Perception A Critical Study of the Experimental Foundation , The Rand Corporation, Santa Monica, California, American Elsevier Publishing Company, New-York, 1968.

Sinton, David, "The Inherent Structure of Information as a Constraint to Analysis: Mapped Thematic Data as a Case Study", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume seven - Spatial Semantics: Understanding and Interacting with Map Data, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Sowa, John F., "A Conceptual Schema for Knowledge-Based Systems", Proceedings of ACM SIGMOD , 1983.

Szoka, Kathryn P., "Making Effective Color Presentations", ISSCO

Graphics Week 1982 Proceedings , February 1982.

Telepak, Robert J., "How to Make Color Lecture Slides by Photographing the CRT Display", REMark , Vol.5, Issue 10, October 1984.

Thorpe, L.W., "Cartographic Data Structures in Production Use at the Experimental Cartography Unit", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume three - Geographic Information Systems: Cartographic and Analytic Applications, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Unwin, David, Introductory Spatial Analysis , Mothuen, London, 1981.

Winograd, Terry, "Frame Representations and the Declarative/Procedural Controversy", Representation and Understanding , Ed. Daniel G. Bobrow and Allan Collins, Academic Press, New-York, 1975.

Winston, Patrick Henry, Artificial Intelligence , Addison-Wesley Publishing Company, Second Printing, April 1979.

Wood, Peter M., "Interactive Display of Polygonal Data", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume seven - Spatial Semantics: Understanding and Interacting with Map Data, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

Woods, William A., "What's Important About Knowledge Representation?", Computer , Special Issue on Knowledge Representation, IEEE Computer Society, October 1983.

Youngmann, Carl, "A Linguistic Approach to Map Description", First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Volume seven - Spatial Semantics: Understanding and Interacting with Map Data, Ed. Geoffrey Dutton, Harvard University, Cambridge, Mass., 1978.

__________. World Atlas , Rand-McNally, Chicago, 1962.

__________. Atlas zur Raumentwicklung , Bonn, 1976.

__________. POLYVRT User's Manual , Version 1.1, Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University, Cambridge, Mass., November 1974.