US006128001A

United States Patent [19]

Gonsalves et al.

[11] **Patent Number: 6,128,001**

[45] **Date of Patent: Oct. 3, 2000**

[54] **METHODS AND APPARATUS FOR CHANGING A COLOR OF AN IMAGE**

[75] Inventors: **Robert Gonsalves**, Wellesley; **Chia-Sheng Chou**, Cambridge; **Brian C. Cooper**, Sharon; **Dion C. Scoppettuolo**, Brookline, all of Mass.

[73] Assignee: **Avid Technology, Inc.**, Tewksbury, Mass.

[21] Appl. No.: **08/832,862**

[22] Filed: **Apr. 4, 1997**

[51] **Int. Cl.**[7] **H04N 3/14**

[52] **U.S. Cl.** **345/150**; 345/431; 345/442; 395/431; 395/142

[58] **Field of Search** 345/431, 442, 345/150; 395/131, 431, 142; 340/701; 427/8

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,367,465	1/1983	Mati et al.	700/98
4,488,245	12/1984	Dalke et al.	382/167
4,492,978	1/1985	Thomas	348/439
4,571,632	2/1986	Schaphorst et al.	358/428
4,642,676	2/1987	Weinger	348/578
4,698,666	10/1987	Lake, Jr. et al.	348/584
4,733,230	3/1988	Kurihara et al.	340/701
4,782,397	11/1988	Kimoto	358/448
4,827,344	5/1989	Astle et al.	348/597
5,129,013	7/1992	Holzmann et al.	382/276
5,134,688	7/1992	Corthout	395/142
5,140,416	8/1992	Tinkler	348/33
5,142,273	8/1992	Wobermin	340/701
5,146,325	9/1992	Ng	348/384
5,181,113	1/1993	Chang	348/468
5,218,671	6/1993	Liao et al.	345/431
5,241,372	8/1993	Ohba	348/578
5,283,651	2/1994	Ishizuka	348/704
5,317,678	5/1994	Okawara et al.	345/426
5,325,449	6/1994	Burt et al.	382/240
5,418,895	5/1995	Lee	395/131
5,444,835	8/1995	Turkowski	345/435
5,450,134	9/1995	Legate	348/467
5,488,674	1/1996	Burt et al.	382/284
5,506,946	4/1996	Bar et al.	345/431
5,510,843	4/1996	Keene et al.	348/446
5,532,751	7/1996	Lui	348/452
5,557,334	9/1996	Legate	348/473
5,557,339	9/1996	Dadourian	348/586

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0 517 035 A2	12/1992	European Pat. Off. .
0 709 806	1/1996	European Pat. Off. .
0 709 806 A2	1/1996	European Pat. Off. .
WO 97/06631	2/1997	WIPO .
WO 98/11510	3/1998	WIPO .

OTHER PUBLICATIONS

Alan Wm. Paeth, Distance Approximations and Bounding Polyhedra, copyright © 1995 by Academic Press, Inc., pp 78–87.

James D. Foley and Andries Van Dam, Fundamentals of Interactive Computer Graphics copyright © 1982 by Addison–Wesley Publishing Company, Inc., pp 245–249.

Eric N. Mortensen and William A. Barrett, Intelligent Scissors for Image Composition Computer Graphics Proceedings, Annual Conference Series 1995, Siggraph 95 Conference Proceedings Aug. 6–11, 1995, A publication of ACM Siggraph, pp. 191–198.

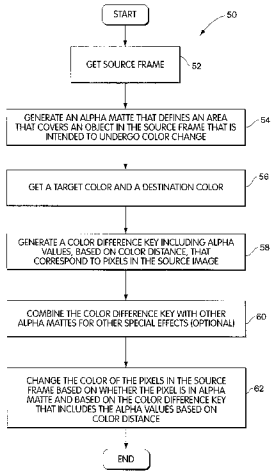
Alvy Ray Smith and James F. Blinn, Blue Screen Matting, Computer Graphics Proceedings, Annual Conference Series, 1996, SIGGRAPH 96, New Orleans, Louisiana, Aug. 4–9, 1996, pp 259–268.

Primary Examiner—William A. Cuchlinski, Jr.
Assistant Examiner—Marthe Y. Marc-Coleman
Attorney, Agent, or Firm—Kenneth L. Milik; Lawrence E. Monks

[57] **ABSTRACT**

The invention provides a mechanism and process for changing a color of an image. In this process, a source frame that includes the image is received. An alpha matte that covers an object in the source frame is created. An original color of a pixel in the source frame is changed when the pixel is covered by the alpha matte, and when a color distance between the original color of the pixel and a target color is below a predetermined threshold.

11 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

5,600,376	2/1997	Casavant et al.	348/423
5,611,030	3/1997	Stokes	395/131
5,611,036	3/1997	Berend et al.	395/141
5,615,324	3/1997	Kuboyama	395/141
5,625,421	4/1997	Faroudja et al.	348/607
5,630,037	5/1997	Schindler	345/431
5,633,687	5/1997	Bhayani et al.	348/441
5,645,884	7/1997	Harlow, Jr. et al.	427/8
5,659,639	8/1997	Mahoney et al.	382/309
5,661,525	8/1997	Kovačevićet al.	348/452
5,663,765	9/1997	Matsuse et al.	348/448
5,703,654	12/1997	Iizuka	348/446
5,703,659	12/1997	Tanaka	348/576
5,745,121	4/1998	Politis	345/433
5,805,169	8/1998	Harada et al.	345/431

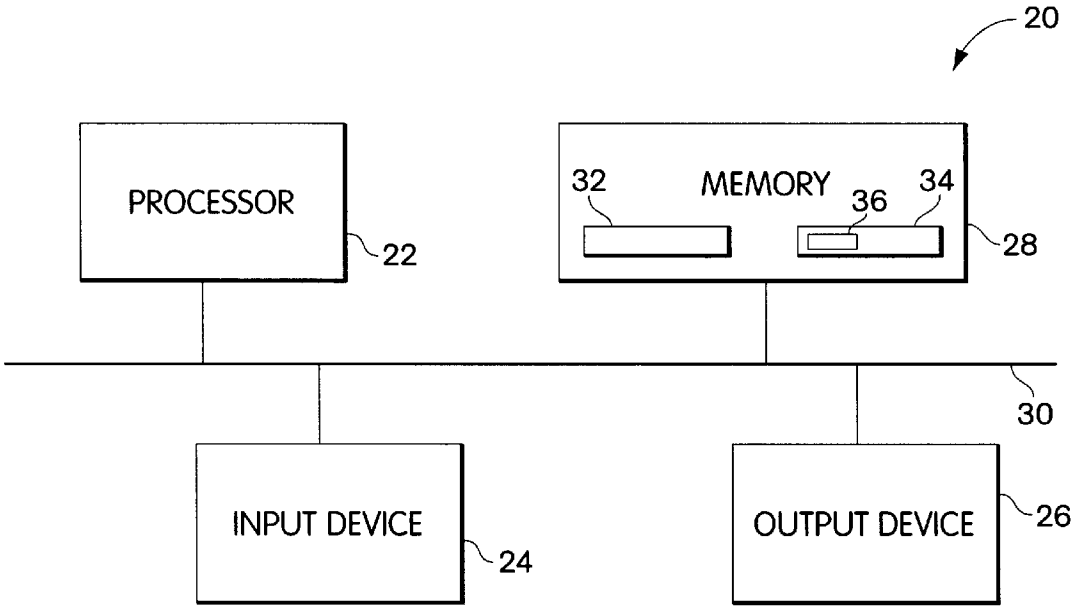

20
PROCESSOR
22
32
MEMORY
36
34
28
30
INPUT DEVICE
24
OUTPUT DEVICE
26

Fig. 1

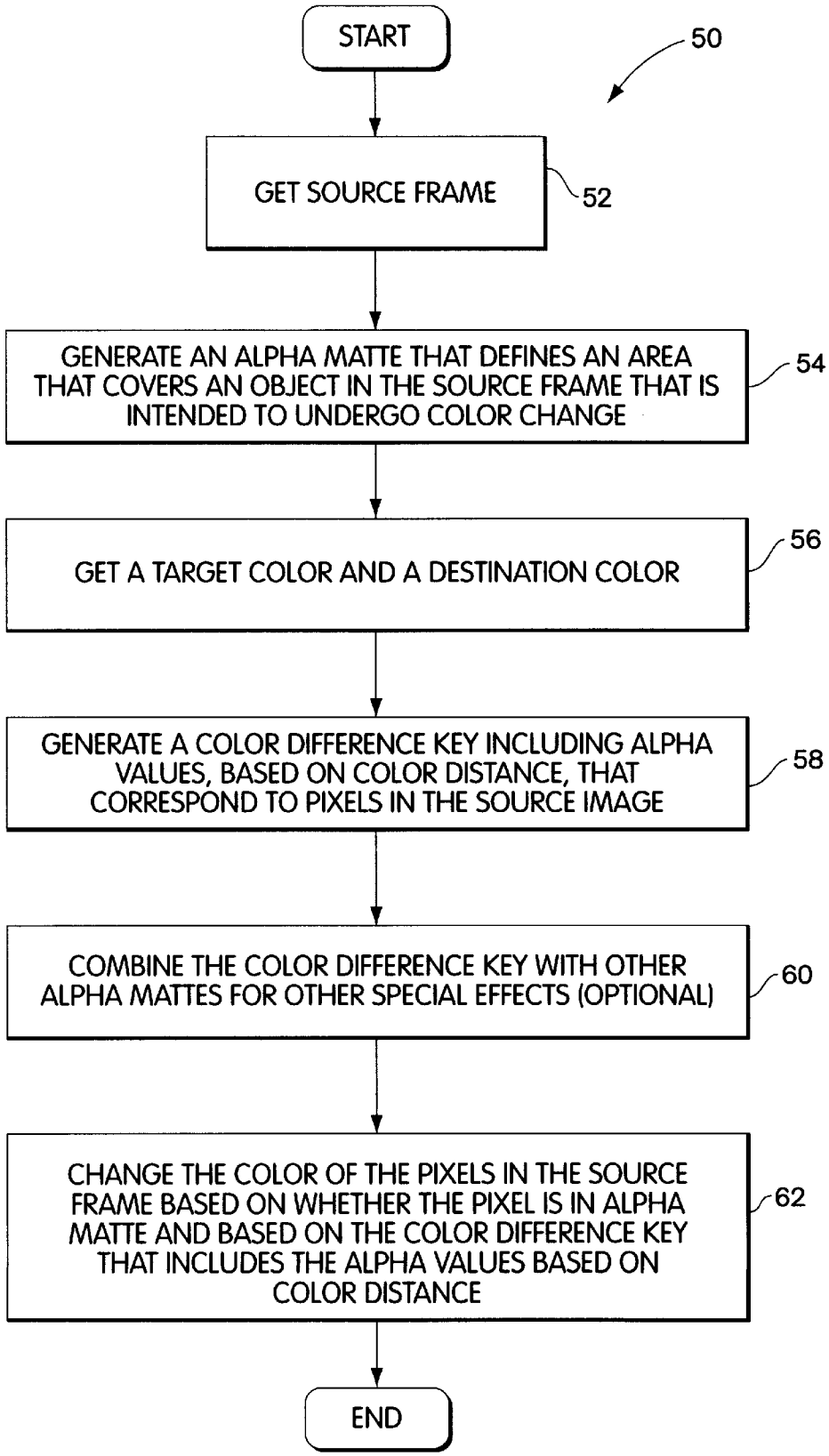

START
50
GET SOURCE FRAME
52
GENERATE AN ALPHA MATTE THAT DEFINES AN AREA THAT COVERS AN OBJECT IN THE SOURCE FRAME THAT IS INTENDED TO UNDERGO COLOR CHANGE
54
GET A TARGET COLOR AND A DESTINATION COLOR
56
GENERATE A COLOR DIFFERENCE KEY INCLUDING ALPHA VALUES, BASED ON COLOR DISTANCE, THAT CORRESPOND TO PIXELS IN THE SOURCE IMAGE
58
COMBINE THE COLOR DIFFERENCE KEY WITH OTHER ALPHA MATTES FOR OTHER SPECIAL EFFECTS (OPTIONAL)
60
CHANGE THE COLOR OF THE PIXELS IN THE SOURCE FRAME BASED ON WHETHER THE PIXEL IS IN ALPHA MATTE AND BASED ON THE COLOR DIFFERENCE KEY THAT INCLUDES THE ALPHA VALUES BASED ON COLOR DISTANCE
62
END

Fig. 2

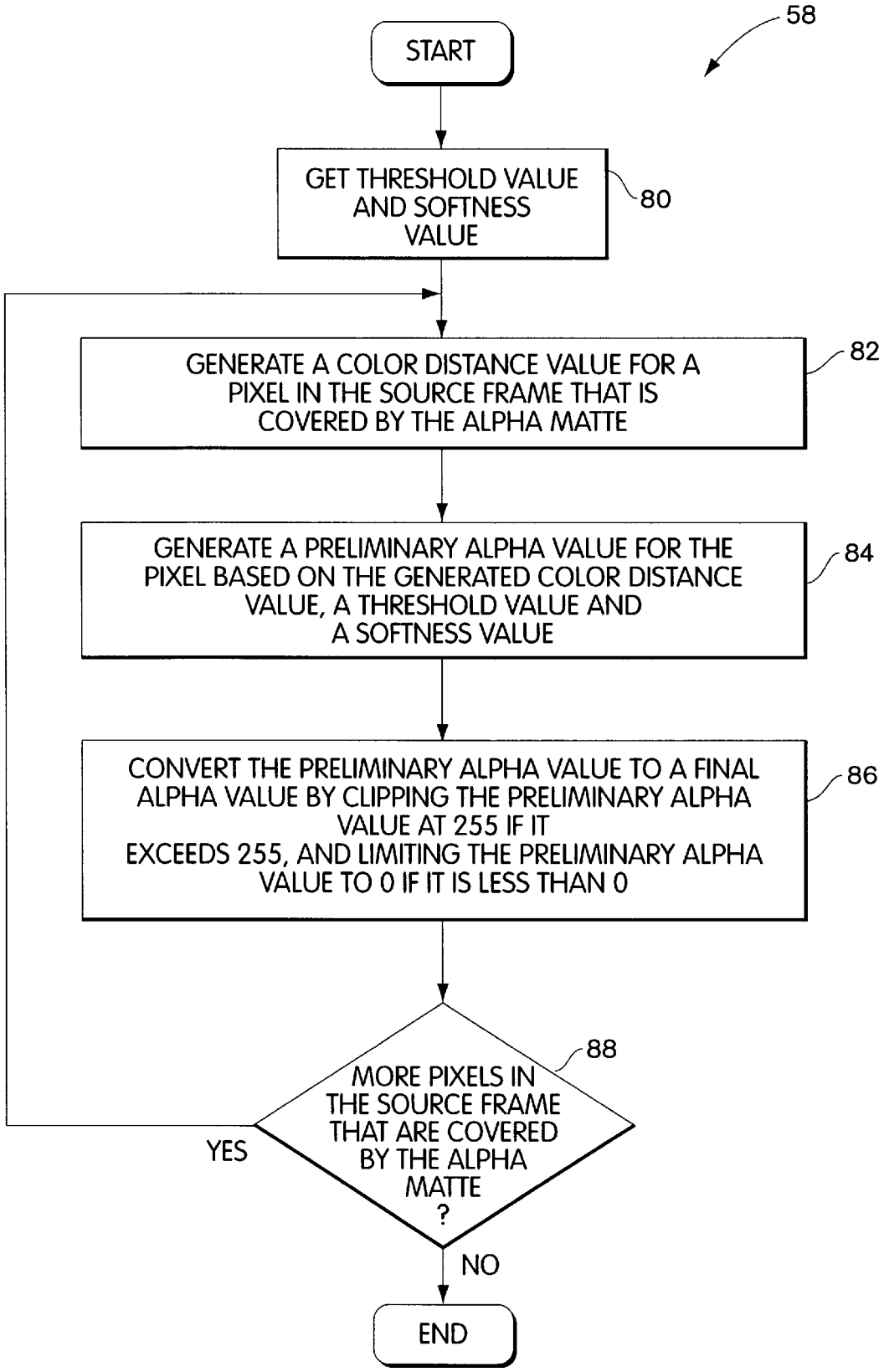

58
START
GET THRESHOLD VALUE AND SOFTNESS VALUE
80
GENERATE A COLOR DISTANCE VALUE FOR A PIXEL IN THE SOURCE FRAME THAT IS COVERED BY THE ALPHA MATTE
82
GENERATE A PRELIMINARY ALPHA VALUE FOR THE PIXEL BASED ON THE GENERATED COLOR DISTANCE VALUE, A THRESHOLD VALUE AND A SOFTNESS VALUE
84
CONVERT THE PRELIMINARY ALPHA VALUE TO A FINAL ALPHA VALUE BY CLIPPING THE PRELIMINARY ALPHA VALUE AT 255 IF IT EXCEEDS 255, AND LIMITING THE PRELIMINARY ALPHA VALUE TO 0 IF IT IS LESS THAN 0
86
88
MORE PIXELS IN THE SOURCE FRAME THAT ARE COVERED BY THE ALPHA MATTE ?
YES
NO
END

Fig. 3

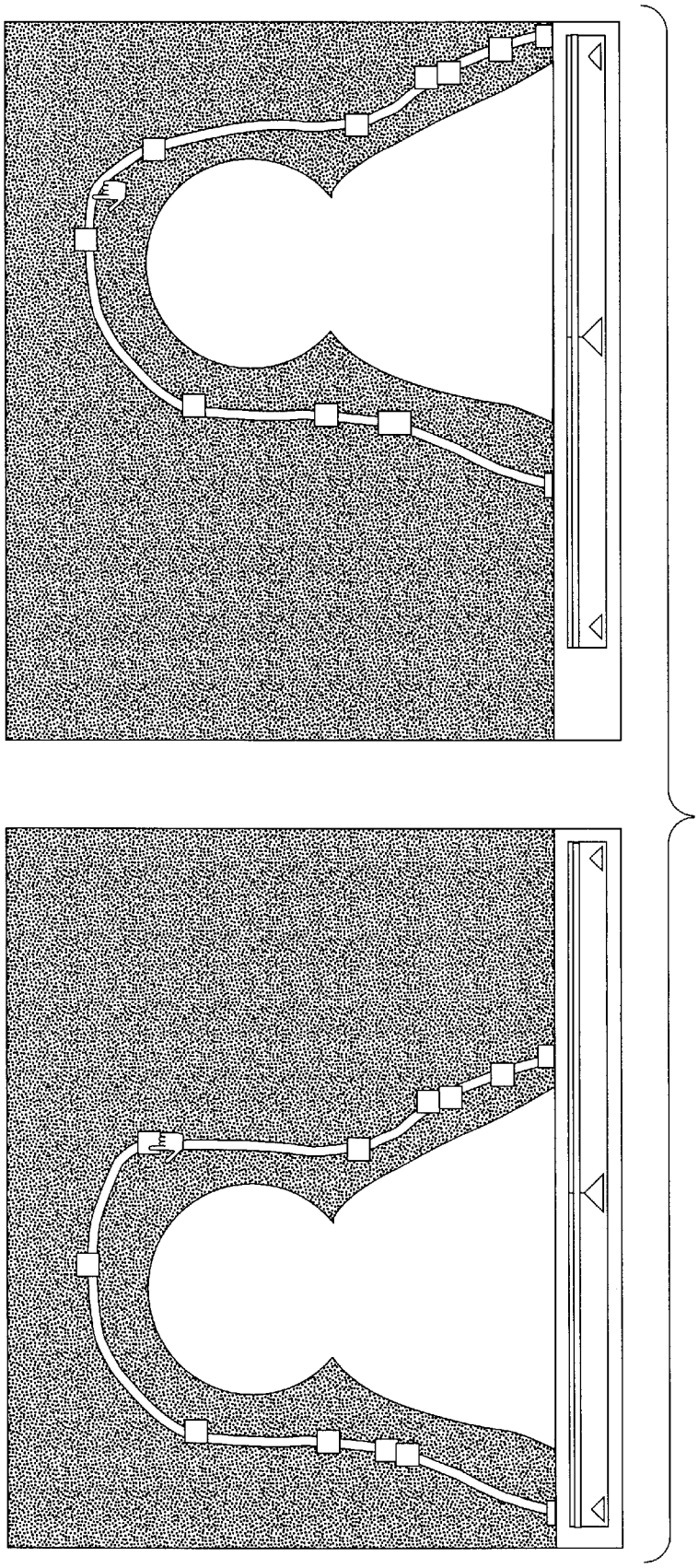


Fig. 4

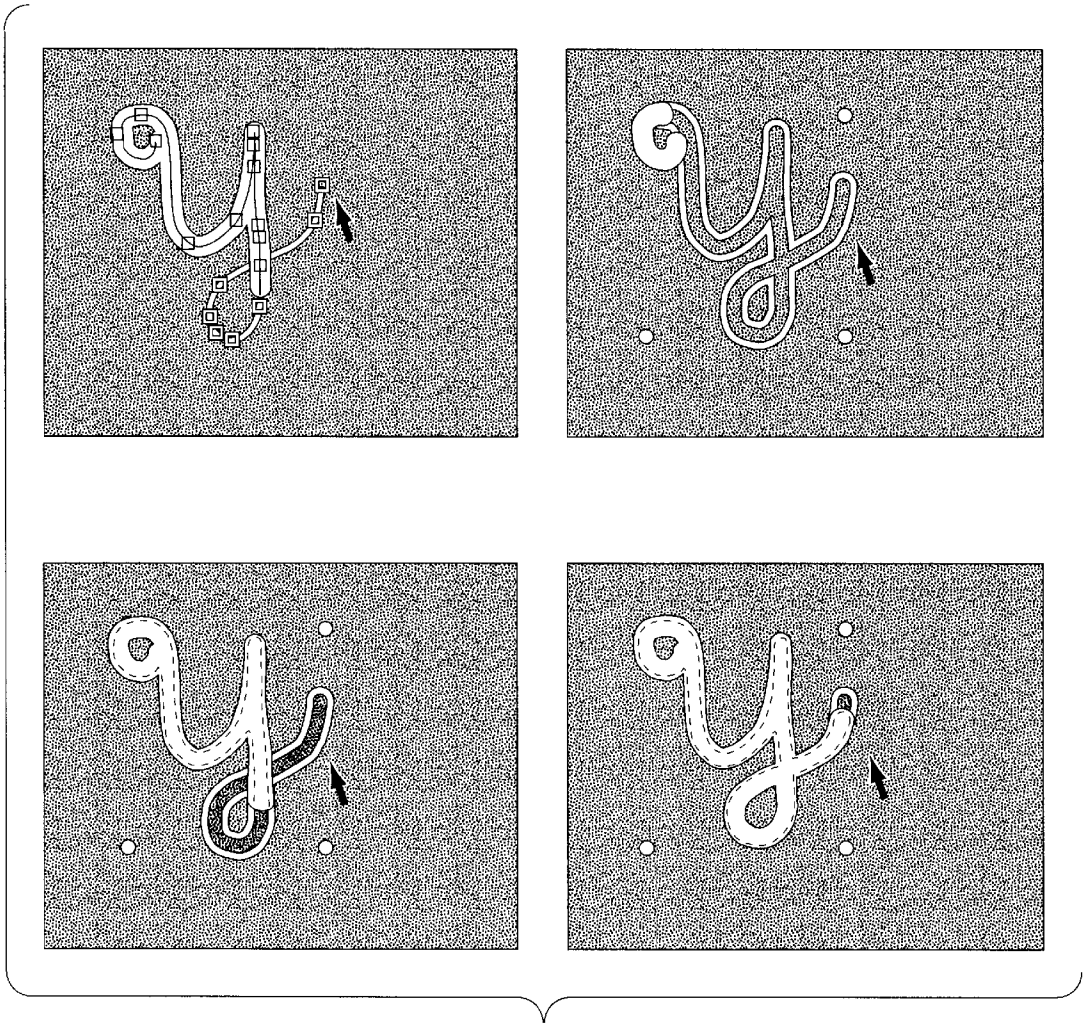


Fig. 5

METHODS AND APPARATUS FOR CHANGING A COLOR OF AN IMAGE

FIELD OF THE INVENTION

The present invention relates generally to image editing systems, and more particularly, to a method and apparatus for changing a color of an image.

BACKGROUND OF THE INVENTION

Producers of still pictures and motion video programs use a variety of special effects to produce a final product. A graphics editor performs the task of adding special effects to still pictures and to motion video segments using a graphics workstation.

Color changing is a special effect that involves changing the color of certain pixels within one or more video image frames. One application of color changing involves modifying the color of an object to make it more or less noticeable in the video image frame. Another application of color changing is to repair a damaged portion of the video image frame. A third application of color changing is to add color to a video image frame to generate the appearance of one or more new objects in the video image frame.

To achieve a color change effect, the graphics editor, using a mouse, graphics tablet or similar input device, manually circumscribes an area of a video image frame that is to receive the color change effect using the graphics workstation. Then, the graphics editor changes the color of each pixel within the circumscribed area to a specified destination color.

Since every pixel within the circumscribed area is changed to the new destination color, the graphics editor must circumscribe the intended color change area with precision. If the graphics editor does a poor job of circumscribing the area, some pixels that were not intended to be changed may be changed, and some pixels intended to be changed may be left unchanged.

Generally, the graphics editor's ability to circumscribe an object in a video image frame is limited. One way for the graphics editor to circumscribe the object is by controlling a cursor on the graphics workstation by manually moving a mouse. In particular, the graphics editor attempts to exactly trace the border around the object with the cursor. Such work requires good hand-eye coordination and is typically painstaking.

Producing color change effects using the above-described technique has additional drawbacks. In particular, setting each pixel of an object to be a destination color generally makes the object look flat and two-dimensional rather than three-dimensional which is often preferred. Additionally, it is not practical for a graphics editor to perform this process on frames of motion video footage that may include a sequence of hundreds or even thousands of individual frames. Furthermore, conventional color correction algorithms performed on computers are complex and require substantial computer power.

SUMMARY OF THE INVENTION

The present invention overcomes these drawbacks by approximately identifying an object in an image, and changing an original color of the object only when the object is covered by an alpha matte, and when a distance between the original color and a target color is below a predetermined threshold.

Accordingly, an aspect of the invention is a computer-implemented process for changing a color of an image. In this process, a source frame that includes the image is received. An alpha matte that covers an object in the source frame is created. An original color of a pixel in the source frame is changed when the pixel is covered by the alpha matte, and when a color distance between the original color of the pixel and a target color is below a predetermined threshold.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

FIG. **1** is a block diagram of a general purpose computer suitable for an embodiment of the invention;

FIG. **2** is a flow diagram of a method, according to an embodiment of the invention;

FIG. **3** is a flow diagram of a step of generating a color difference key, according to an embodiment of the invention;

FIG. **4** is a block diagram illustrating the use of key frames within a sequence of frames according to an embodiment of the invention; and

FIG. **5** is a perspective view of a display showing a plurality of Bezier curves that define a paint path, according an embodiment of the invention.

DETAILED DESCRIPTION

The present invention will be more completely understood through the following detailed description which should be read in conjunction with the attached drawing in which similar reference numbers indicate similar structures. All references cited herein are hereby expressly incorporated by reference.

An embodiment of the invention is directed to an apparatus for changing the color of an object in one or more video image frames. In particular, the apparatus includes a general purpose computer **20** having a processor **22**, an input device **24**, an output device **26**, and memory **28** connected together through a computer bus **30**. The memory includes primary memory **32** (i.e., fast volatile memory such as dynamic semiconductor memory) and secondary memory **34** (i.e., nonvolatile memory such as magnetic disks). The memory **28** stores one or more programs **36** that, when executed by the processor **22**, effect the color change. The input device **24** receives commands from a graphics editor, and the output device **26** displays results of the color change to the graphics editor. The results of the color change can be stored in the memory **28** for future use.

The program **36**, when executed on the general purpose computer **20**, controls the general purpose computer **20** such that it performs a method **50**, as illustrated in FIG. **2**. In step **52**, the graphics editor retrieves a video image frame on the computer **20** for color change processing. This video image frame is called the source frame.

In step **54**, the graphics editor identifies an object in the source frame that is intended to undergo color change. The graphics editor does not need to circumscribe the object in the source frame exactly as in conventional methods. Rather, the graphics editor generates an alpha matte that approximately covers the object in the source frame. The alpha matte thereby provides a general indication of the location of the object in the source frame.

The alpha matte is a Bezier form that includes one or more Bezier curves. The graphics editor creates the alpha matte using the input device **24**. The graphics editor may create the alpha matte (i.e., the Bezier form) in a number of ways.

A first way is for the graphics editor to expressly specify the Bezier form on a point-by-point basis. That is, the

graphics editor specifies points and tangent vectors that define a set of Bezier curves that form a closed Bezier form.

A second way is for the graphics editor to draw freehand, e.g., using a mouse, Bezier curves that form a closed Bezier form. This second way can be further enhanced by using a software scissors tool that allows the graphics editor to snap a portion of the closed Bezier form around the intended object in the image to more accurately surround the object. A third way is for the graphics editor to draw a rectangular box around the object to undergo the color change. A fourth way is for the graphics editor to draw an elliptical form around the object to undergo the color change. In each of the second, third and fourth ways, the system converts the graphics editor inputs into points and tangent vectors that define Bezier curves.

It should be understood that each of the above described ways of creating the alpha matte produces a closed Bezier form that surrounds the object intended to undergo the color change.

A fifth way to create the alpha matte is for the graphics editor to draw an open Bezier form that is defined by one or more Bezier curves and a width. The open Bezier form is two-dimensional and covers the object that will undergo color correction. It should be understood that this technique differs from the earlier described techniques since it generates an open Bezier form that covers the object rather than surround the object.

In step **56**, the graphics editor inputs additional information such as a target color which identifies approximately a color within the alpha matte that should undergo color change, and a destination color which identifies approximately a color that the object is to be changed to.

In step **58**, the computer generates a color change key that includes an alpha value α for each pixel of the source frame that is covered by the alpha matte. That is, the color change key is a two-dimensional array of alpha values α that resembles the shape of the alpha matte. Each alpha value α corresponds to a pixel in the source frame that is covered by the alpha matte, and indicates how much color change its corresponding source frame pixel should undergo. In particular, the alpha values α are within the range 0 to 255. An alpha value of 0 indicates that the corresponding pixel in the source frame should undergo no color change. An alpha value of 255 indicates that the corresponding pixel should undergo a full or complete color change. An alpha value between 0 and 255 indicates that the corresponding pixel should undergo a partial color change. A partial color change is a change of pixel color to a color that is between the source pixel color and the destination color. As will be described in detail later, the alpha values α are based on color distance between the actual color values of the pixels in the source frame and the target color.

In step **60**, the alpha matte may be further modified or combined with other alpha mattes to add other special effects. For example, the alpha matte can be combined with alpha mattes that add feathering and opacity to one or more areas in the source frame.

In step **62**, the general purpose computer **50** changes the color of the pixels in the source frame based on whether the pixel is in the alpha matte and based on the color difference key that includes the alpha values α that are based on color distance. In one embodiment, the pixel color is specified by RGB values ranging from 0 to 255. In this embodiment, each color for a pixel P_a of the source frame covered by the alpha matte is determined by the following equations:

$$\text{Red Component For Pixel } P_n = \frac{(\alpha_a \times [\text{Red in Dest. color}]) + ((255-\alpha_a)\times[\text{Red in Orig. Pixel}])}{255} \quad \text{(Eq. 1a)}$$

$$\text{Green Component For Pixel } P_n = \frac{(\alpha_a \times [\text{Green in Dest. color}]) + ((255-\alpha_a)\times[\text{Green in Orig. Pixel}])}{255} \quad \text{(Eq. 1b)}$$

$$\text{Blue Component For Pixel } P_n = \frac{(\alpha_a \times [\text{Blue in Dest. color}]) + ((255-\alpha_a)\times[\text{Blue in Orig. Pixel}])}{255} \quad \text{(Eq. 1c)}$$

The color of each pixel in the alpha matte is determined according to Eqs. (1a), (1b) and (1c).

When the computer **50** is finished making the above determinations for each pixel in the alpha matte, the source frame has been converted into a color changed frame.

A more detailed explanation of step **58** will now be provided. In step **58**, an alpha value is determined for each pixel in the source frame that is covered by the alpha matte, as illustrated in FIG. **3**. In step **80**, a threshold value and a softness value are obtained from the graphics editor. These values may have been preset earlier in time.

In step **82**, the computer **20** generates a color distance value for each pixel in the source frame that is covered by the alpha matte. The color distance value is an overall color distance based on individual color component distances. In one embodiment, the color distance value for each pixel is determined by individual color component distances using the following equation:

$$\text{Pixel color distance Value} = \sqrt{((\Delta R)^2 + (\Delta G)^2 + (\Delta B)^2)}, \quad \text{(Eq. 2)}$$

where:

ΔR=R comp. of source pixel−R comp. of target color,

ΔG=G comp. of source pixel−G comp. of target color, and

ΔB=B comp. of source pixel−G comp. of target color.

In another embodiment, the color distance value for each pixel is determined as follows. First, the following individual color distances are determined for a pixel:

$$R_{distance}=\Delta R=|R \text{ comp. of source pixel}-R \text{ comp. of target color}| \quad \text{(Eq. 3a)}$$

$$G_{distance}=\Delta G=|G \text{ comp. of source pixel}-G \text{ comp. of target color}| \quad \text{(Eq. 3b)}$$

$$B_{distance}=\Delta B=|B \text{ comp. of source pixel}-B \text{ comp. of target color}| \quad \text{(Eq. 3c)}$$

Then, a variable V_1 is set to the largest of the individual color distances for the pixel. A variable V_3 is set to the smallest of the individual color distances for the pixel. And, a variable V_2 is set to the intermediate value of the individual color distances for the pixel. The color distance value is then determined by the following equation:

$$\text{Pixel color distance value}=V_1+\tfrac{1}{2}V_2+\tfrac{1}{4}V_3. \quad \text{(Eq. 4)}$$

This second way of determining the pixel color distance value is described in *Graphic Gems V*, by Alan Wm. Paeth,

entitled "Distance Approximations and Bounding Polyhedra," p. 78, which is hereby incorporated by reference.

In step **84**, a preliminary alpha value α' is determined by the following equation:

$$\alpha' = \frac{([\text{threshold value}] - [\text{color_distance}]) \times 255}{[\text{softness}]} \quad \text{(Eq. 5)}$$

Then, in step **86**, the computer converts the preliminary alpha value α' into an actual alpha value α. In particular, if the preliminary alpha value α' is within the range of 0 and 255 it is automatically considered to be the actual alpha value α. However, if the preliminary alpha value α' exceeds 255, it is converted to an actual alpha value of 255. Similarly, if the preliminary alpha value α' is less than 0, it is converted to an actual alpha value α having a value of 0.

In step **88**, the above determinations are repeated for each pixel in the source frame that is covered by the alpha matte. The end result is a color difference key having alpha values α that correspond to the pixels in the source frame that are covered by the alpha matte.

It should be understood that the threshold value controls which pixels within the alpha matte are to undergo color change. As indicated earlier, an alpha value α of 0 causes no color change in the corresponding pixel. The alpha value α is 0 whenever the pixel color distance value exceeds the threshold value since α is a negative number that becomes adjusted to 0. On the other hand, if the pixel color distance value is below the threshold value, the alpha value is a positive number and causes some amount of color change. The graphics editor can modify the threshold value so that more or less pixels of the source frame covered by the alpha matte undergo a color change.

Additionally, it should be understood that the softness value adjusts the amount of the color change on a global scale. A large softness value reduces the determined alpha values α. However, a low softness value increases the determined alpha values α. The graphics editor may adjust the softness value to modify the degree or amount of color change performed on the pixels undergoing color change.

If the video frame that underwent the color change belongs to a sequence of frames, and the graphics editor wishes to make a similar color change to the other frames in the sequence, the graphics editor may move the alpha matte in the frame sequence using key frames, as illustrated in FIG. **4**. The graphics editor should verify that the image undergoing the color change remains within the alpha matte as it moves during the frame sequence. The steps of the method can be repeated automatically using the general purpose computer **20**, and the graphics editor need not manually define a Bezier form for each frame.

The operation of the system will now be further explained by way of example. The graphics editor is given a sequence of source frames showing a yellow frisbee that moves from one end of the image to the other. The graphics editor wishes to change the color of the frisbee from yellow to pink. The graphics editor does not need to circumscribe the frisbee with a high level of accuracy. Rather, the graphics editor, viewing the first source frame on the display device **26** of the computer **20**, may draw a rough trace around the frisbee making sure the trace completely includes the frisbee image. The computer **20** converts the graphics editor's inputs into a Bezier form having one or more Bezier curves, and the Bezier form defines an alpha matte that covers the frisbee image. The computer **20** generates a color difference key that includes a set of alpha values corresponding to the pixels in the first source frame covered by the alpha matte. The alpha values are based on a threshold value, a softness value, and a target color that have been specified by the graphics editor earlier. The computer **20** then combines the alpha matte with other alpha mattes that provide other special effects (this is optional). Next, the computer **20** changes the color of the yellow frisbee from the original yellow source color to one or more shades of pink, according to the degree of change identified in the color difference key.

After the first source frame has been processed, the graphics editor identifies key frames within the sequence of frames, and uses the key frames to move the alpha matte to new locations in other frames through the sequence of frames so that the moving frisbee image remains within the moving alpha matte. The computer **20** then color changes the remaining frames in the sequence by repeating the above described steps. If the sequence is displayed, the viewer sees a pink frisbee moving across the image.

According to an embodiment of the invention, the alpha matte continues moving throughout the sequence of frames along a paint path. For example, the graphics editor may wish to create an effect of handwriting on a chalkboard. The graphics editor defines an open Bezier form made up of one or more Bezier curves to define the paint path. The first curve is defined using two points and two tangent vectors. Subsequent Bezier curves are defined using one point from a previous Bezier curve and four tangent vectors. The general purpose computer **20** determines the total curve length of the open Bezier form. Then, the total curve length is divided by the number of frames that will be used to display creation of the paint path. Next, the curves are displayed linearly, wherein each previously displayed curve is redrawn for each frame, as illustrated in FIG. **5**. The effect is an appearance of handwriting during the frame sequence.

Discontinuities in the handwriting process such as dotting the i's and crossing the t's can be added by generating other paint paths that separately add dots and crosses.

Additionally, rather than evenly dividing the Bezier form by the number of frames, the curve segments may be given a variable size to make the handwriting appear more realistic. Alternatively, time deltas may be incorporated so that the handwriting appears more realistic.

Modifications and enhancements can be made to the above-described embodiments, and are intended to be within the scope of the invention.

For example, step **60** involves changing the RGB components as the color change. If the system uses different color components other than red, green and blue, step **60** may be modified to use the other components instead. Furthermore, step **60** may include other paint modes. For example, step **60** may change the luminance of a portion of the image, or the hue of the portion of the image. According to an embodiment of the invention, a graphics editor may command the general purpose computer **20** to perform any of various paint modes provided by Matador, a software paint tool, available from Avid Technology, Inc., Tewksbury, Mass.

Furthermore, the alpha matte may be turned on or off. Accordingly, a second alpha matte may be generated and turned off. If the second alpha matte overlaps portions of another alpha matte that performs color change, the second alpha matte effectively erases any color change made by the other alpha matte. Such a feature may be used when certain objects undergo an undesired color change. For example, if a yellow frisbee traveling from one end of an image to another end through a sequence of frames is changed to pink, and part of the image includes a person wearing a

yellow T-shirt. It is possible that the person's T-shirt may be changed to pink if it was covered by the alpha matte. The graphics editor may then create a second alpha matte to erase the color change of the T-shirt. Alternatively, the graphics editor may modify the shape of the alpha matte so that the T-shirt is not covered by the original alpha matte.

Having now described a few embodiments of the invention, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention as defined by the appended claims and equivalent thereto.

What is claimed is:

1. A computer-implemented process for changing a color of an image, comprising the steps of:

receiving a source frame that includes the image;

identifying an object in the source frame; and

changing an original color of a pixel in the source frame to a new color if the pixel is in the identified object and a color distance between the original color of the pixel and a target color is below a predetermined threshold; and

wherein the original color includes an amount of a first color component and an amount of a second color component, and wherein the step of changing the original color of the pixel in the source frame to the new color includes steps of:

determining an amount of the first color component in the new color based on the amount of the first color component in the original color, an amount of the first color component in a destination color, and an alpha value; and

determining an amount of the second color component in the new color based on the amount of the second color component in the original color, an amount of the second color component in the destination color, and the alpha value.

2. The computer-implemented process of claim **1**, further comprising a step of:

generating a color difference key having a plurality of alpha values that correspond to a plurality of pixels in the source frame, one of the plurality of alpha values being the alpha value used when determining the amounts of first color component and the second color component in the new color.

3. The computer-implemented process of claim **2**, wherein the target color includes an amount of the first color component and an amount of the second color component, and wherein the step of generating the color difference key includes steps of:

determining, for each color component, a distance between the amount of the color component in the original color and the amount of the color component in the target color;

squaring the determined distances to generate a squared result for each color component;

adding the squared result for each color component to generate a sum; and

determining a square root of the sum to generate the alpha value used when determining the amounts of first color component and the second color component in the new color.

4. The computer-implemented process of claim **2**, wherein the step of generating the color difference key includes, for each of the plurality of alpha values, steps of:

determining a preliminary value; and

converting the preliminary value into the alpha value.

5. The computer-implemented process of claim **4**, wherein the step of determining the preliminary value includes a step of:

deriving the preliminary value based on a softness value.

6. The computer-implemented process of claim **4**, wherein the step of converting the preliminary value into the alpha value includes steps of:

using the preliminary value as the alpha value if the preliminary value is below a first threshold and above a second threshold;

using the first threshold as the alpha value if the preliminary value exceeds the first threshold; and

using the second threshold as the alpha value if the preliminary value is below the second threshold.

7. A computer-implemented process for changing a color of an image, comprising the steps of:

receiving a source frame that includes the image;

identifying an object in the source frame; and

changing an original color of a pixel in the source frame to a new color if the pixel is in the identified object and a color distance between the original color of the pixel and a target color is below a predetermined threshold, and

wherein the step of identifying an object further comprises the steps of:

defining an alpha matte that covers a portion of the image; and

modifying the defined alpha matte according to another alpha matte before changing the original color of the pixel to add a special effect defined by the other alpha matte.

8. An apparatus for feathering a first image and a second image in a composite frame, comprising:

a general purpose computer including:

a memory that stores a program; and

a processor that, when executing the program, receives a source frame that includes the image; identifies an object in the source frame; and changes an original color of a pixel in the source frame to a new color if the pixel is in the identified object and a color distance between the original color of the pixel and a target color is below a predetermined threshold: and

wherein each of the original color and a destination color includes an amount of a first color component and an amount of a second color component, and wherein the processor, when executing the program, changes the original color of the pixel in the source frame to the new color by:

determining an amount of the first color component in the new color based on the amount of the first color component in the original color, the amount of the first color component in the destination color, and an alpha value; and

determining an amount of the second color component in the new color based on the amount of the second color component in the original color, the amount of the second color component in the destination color, and the alpha value.

9. The apparatus of claim **8**, wherein the processor, when executing the program, generates a color difference key having a plurality of alpha values that correspond to a plurality of pixels in the source frame, one of the plurality

of alpha values being the alpha value used when determining the amounts of first color component and the second color component in the new color.

10. The apparatus of claim **9**, wherein the target color includes an amount of the first color component and an amount of the second color component; and wherein the processor, when executing the program, determines, for each color component, a distance between the amount of the color component in the original color and the amount of the color component in the target color, squares the determined distances to generate a squared result for each color component, adds the squared result for each color component to generate a sum, and determines a square root of the sum to generate the alpha value used when determining the amounts of first color component and the second color component in the new color to generate the color difference key.

11. The apparatus of claim **9**, wherein the processor, when executing the program, determines a preliminary value for each of the plurality of alpha values, and converts the preliminary value into the alpha value for each of the plurality of alpha values to generate the color difference key.

* * * * *