US009270965B2

(12) **United States Patent**
Sandrew et al.

(10) **Patent No.: US 9,270,965 B2**
(45) **Date of Patent: Feb. 23, 2016**

(54) **MULTI-STAGE PRODUCTION PIPELINE SYSTEM**

(71) Applicant: **LEGEND3D, Inc.**, Carlsbad, CA (US)

(72) Inventors: **Barry Sandrew**, San Diego, CA (US); **Jared Sandrew**, San Diego, CA (US); **Nancy Wang**, San Diego, CA (US); **Craig Cesareo**, San Diego, CA (US); **James Prola**, San Diego, CA (US); **Jill Hunt**, San Diego, CA (US); **Anthony Lopez**, San Diego, CA (US)

(73) Assignee: **LEGEND 3D, INC.**, Carlsbad, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/828,354**

(22) Filed: **Aug. 17, 2015**

(65) **Prior Publication Data**

US 2015/0358595 A1 Dec. 10, 2015

Related U.S. Application Data

(63) Continuation of application No. 13/895,979, filed on May 16, 2013, now Pat. No. 9,113,130, which is a continuation-in-part of application No. 13/366,899, filed on Feb. 6, 2012, now Pat. No. 9,031,383.

(51) **Int. Cl.**
H04N 7/18 (2006.01)
H04N 9/79 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04N 9/79*** (2013.01); ***G11B 27/036*** (2013.01); ***G11B 27/34*** (2013.01); ***H04N 13/0055*** (2013.01); ***H04N 13/0257*** (2013.01); ***H04N 13/0267*** (2013.01)

(58) **Field of Classification Search**
USPC 348/43
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,619,051 A 11/1971 Wright
3,621,127 A 11/1971 Hope
(Continued)

FOREIGN PATENT DOCUMENTS

DE 003444353 6/1986
EP 0302454 2/1995
(Continued)

OTHER PUBLICATIONS

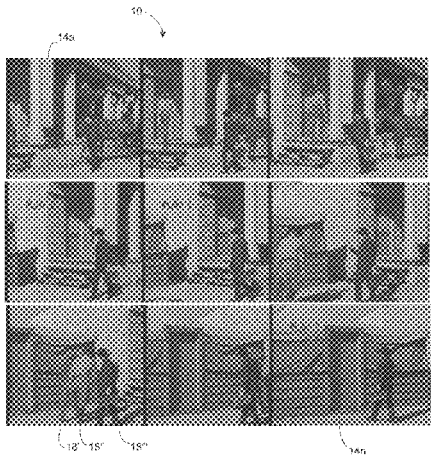
Lenny Lipton, "Foundations of the Stereo-Scopic Cinema, a Study in Depth" With an Appendix on 3D Television, 325 pages, May 1982.
(Continued)

Primary Examiner — Dave Czekaj
Assistant Examiner — Leron Beck
(74) *Attorney, Agent, or Firm* — ARC IP Law, PC; Joseph J. Mayo

(57) **ABSTRACT**

Multi-stage production pipeline system that may be utilized in conjunction with a motion picture project management system. The multi-stage production pipeline system includes a computer and a database. The database includes metadata associated with at least one shot or associated with regions within the plurality of images in the at least one shot, or both. The computer includes a grouping tool interface for presenting user interface elements and accepting input of the metadata associated with the at least one shot or regions within the plurality of images in the at least one shot, or both. The system enables a large studio workforce to work non-linearly on a film while maintaining a unified vision driven by key creative figures, allowing for more consistent, higher quality, faster, less expensive work product and more efficient project management techniques. The system also enables reuse of project files, masks and other production elements across projects.

30 Claims, 112 Drawing Sheets



(51) **Int. Cl.**

H04N 13/02	(2006.01)
G11B 27/036	(2006.01)
G11B 27/34	(2006.01)
H04N 13/00	(2006.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,705,762	A		12/1972	Ladd et al.	
3,737,567	A		6/1973	Kratomi	
3,772,465	A		11/1973	Vlahos et al.	
3,851,955	A		12/1974	Kent et al.	
4,021,841	A		5/1977	Weinger	
4,021,846	A		5/1977	Roese	
4,149,185	A		4/1979	Weinger	
4,183,633	A		1/1980	Kent et al.	
4,235,503	A		11/1980	Condon	
4,436,369	A		3/1984	Bukowski	
4,475,104	A		10/1984	Shen	
4,544,247	A		10/1985	Ohno et al.	
4,558,359	A		12/1985	Kuperman et al.	
4,563,703	A	*	1/1986	Taylor	G06T 3/4007 348/580
4,600,919	A		7/1986	Stern	
4,603,952	A		8/1986	Sybenga	
4,606,625	A		8/1986	Geshwind	
4,608,596	A	*	8/1986	Williams	H04N 9/43 345/536
4,642,676	A		2/1987	Weinger	
4,645,459	A		2/1987	Graf et al.	
4,647,965	A		3/1987	Imsand	
4,697,178	A		9/1987	Heckel	
4,723,159	A	*	2/1988	Imsand	H04N 13/00 348/42
4,755,870	A		7/1988	Markle et al.	
4,774,583	A		9/1988	Kellar et al.	
4,809,065	A		2/1989	Harris et al.	
4,888,713	A		12/1989	Falk	
4,903,131	A		2/1990	Lingermann et al.	
4,925,294	A		5/1990	Geshwind et al.	
4,933,670	A		6/1990	Wislocki	
4,965,844	A	*	10/1990	Oka	G06T 1/0007 345/426
4,984,072	A		1/1991	Sandrew	
5,002,387	A		3/1991	Baljet et al.	
5,038,161	A		8/1991	Ki	
5,050,984	A		9/1991	Geshwind	
5,055,939	A		10/1991	Karamon et al.	
5,093,717	A		3/1992	Sandrew	
5,181,181	A		1/1993	Glynn	
5,185,852	A		2/1993	Mayer	
5,237,647	A		8/1993	Roberts et al.	
5,243,460	A		9/1993	Kornberg	
5,252,953	A		10/1993	Sandrew	
5,328,073	A		7/1994	Blanding et al.	
5,341,462	A		8/1994	Obata	
5,363,476	A		11/1994	Kurashige et al.	
5,402,191	A		3/1995	Dean et al.	
5,428,721	A		6/1995	Sato et al.	
5,495,576	A		2/1996	Ritchey	
5,534,915	A		7/1996	Sandrew	
5,668,605	A		9/1997	Nachshon et al.	
5,673,081	A		9/1997	Yamashita et al.	
5,682,437	A		10/1997	Okino et al.	
5,684,715	A		11/1997	Palmer	
5,699,443	A		12/1997	Murata et al.	
5,699,444	A		12/1997	Palm	
5,729,471	A		3/1998	Jain et al.	
5,734,915	A		3/1998	Roewer	
5,739,844	A		4/1998	Kuwano et al.	
5,748,199	A		5/1998	Palm	
5,777,666	A		7/1998	Tanase et al.	
5,778,108	A		7/1998	Coleman	
5,784,175	A		7/1998	Lee	
5,784,176	A		7/1998	Narita	
5,808,664	A		9/1998	Yamashita et al.	
5,835,163	A		11/1998	Liou et al.	
5,841,512	A		11/1998	Goodhill	
5,899,861	A		5/1999	Friemel et al.	
5,920,360	A		7/1999	Coleman	
5,929,859	A		7/1999	Meijers	
5,940,528	A		8/1999	Tanaka et al.	
5,959,697	A		9/1999	Coleman	
5,973,700	A		10/1999	Taylor et al.	
5,982,350	A		11/1999	Hekmatpour et al.	
5,990,900	A		11/1999	Seago	
5,990,903	A		11/1999	Donovan	
6,005,582	A	*	12/1999	Gabriel	G06T 11/40 345/586
6,011,581	A		1/2000	Swift et al.	
6,014,473	A		1/2000	Hossack et al.	
6,025,882	A		2/2000	Geshwind	
6,031,564	A		2/2000	Ma et al.	
6,049,628	A		4/2000	Chen et al.	
6,056,691	A		5/2000	Urbano et al.	
6,067,125	A		5/2000	May	
6,086,537	A		7/2000	Urbano et al.	
6,088,006	A		7/2000	Tabata	
6,091,421	A		7/2000	Terrasson	
6,102,865	A		8/2000	Hossack et al.	
6,108,005	A		8/2000	Starks et al.	
6,119,123	A		9/2000	Elenbaas et al.	
6,132,376	A		10/2000	Hossack et al.	
6,141,433	A		10/2000	Moed et al.	
6,198,484	B1		3/2001	Kameyama	
6,201,900	B1		3/2001	Hossack et al.	
6,208,348	B1		3/2001	Kaye	
6,211,941	B1		4/2001	Erland	
6,222,948	B1		4/2001	Hossack et al.	
6,226,015	B1		5/2001	Danneels et al.	
6,228,030	B1		5/2001	Urbano et al.	
6,263,101	B1		7/2001	Klein et al.	
6,314,211	B1		11/2001	Kim et al.	
6,337,709	B1		1/2002	Yamaashi et al.	
6,360,027	B1		3/2002	Hossack et al.	
6,364,835	B1		4/2002	Hossack et al.	
6,373,970	B1		4/2002	Dong et al.	
6,390,980	B1		5/2002	Peterson et al.	
6,405,366	B1		6/2002	Lorenz et al.	
6,414,678	B1		7/2002	Goddard et al.	
6,416,477	B1		7/2002	Jago	
6,429,867	B1		8/2002	Deering	
6,445,816	B1		9/2002	Pettigrew	
6,456,340	B1		9/2002	Margulis	
6,466,205	B2		10/2002	Simpson et al.	
6,477,267	B1		11/2002	Richards	
6,492,986	B1		12/2002	Metaxas et al.	
6,496,598	B1		12/2002	Harman	
6,509,926	B1		1/2003	Mills et al.	
6,515,659	B1		2/2003	Kaye et al.	
6,535,233	B1		3/2003	Smith	
6,553,184	B1		4/2003	Ando et al.	
6,590,573	B1		7/2003	Geshwind	
6,677,944	B1		1/2004	Yamamoto	
6,686,591	B2		2/2004	Ito et al.	
6,686,926	B1		2/2004	Kaye	
6,707,487	B1		3/2004	Aman et al.	
6,727,938	B1		4/2004	Randall	
6,744,461	B1		6/2004	Wada et al.	
6,765,568	B2		7/2004	Swift et al.	
6,791,542	B2		9/2004	Matusik et al.	
6,798,406	B1		9/2004	Jones et al.	
6,850,252	B1		2/2005	Hoffberg	
6,853,383	B2		2/2005	Duquesnois	
6,919,892	B1		7/2005	Cheiky et al.	
6,965,379	B2		11/2005	Lee et al.	
6,985,187	B2		1/2006	Han et al.	
7,035,451	B2		4/2006	Harman et al.	
7,098,910	B2		8/2006	Petrovic et al.	
7,102,633	B2		9/2006	Kaye et al.	
7,110,605	B2	*	9/2006	Marcellin	G11B 27/034 375/E7.04
7,116,323	B2		10/2006	Kaye et al.	
7,116,324	B2		10/2006	Kaye et al.	
7,181,081	B2		2/2007	Sandrew	

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,190,496	B2	3/2007	Klug et al.	
7,254,264	B2	8/2007	Naske et al.	
7,254,265	B2	8/2007	Naske et al.	
7,321,374	B2	1/2008	Naske	
7,327,360	B2	2/2008	Petrovic et al.	
7,333,670	B2	2/2008	Sandrew	
7,542,034	B2	6/2009	Spooner et al.	
7,573,475	B2	8/2009	Sullivan et al.	
7,573,489	B2	8/2009	Davidson et al.	
7,577,312	B2	8/2009	Sandrew	
8,217,931	B2	7/2012	Lowe et al.	
8,244,104	B2 *	8/2012	Kashiwa	G11B 27/034 386/278
8,670,651	B2 *	3/2014	Sakuragi	G06F 17/30589 386/278
2002/0048395	A1	4/2002	Harman et al.	
2002/0063780	A1	5/2002	Harman et al.	
2002/0075384	A1	6/2002	Harman	
2004/0062439	A1 *	4/2004	Cahill	G06T 5/50 382/173
2004/0130680	A1 *	7/2004	Zhou	H04N 19/85 352/38
2004/0151471	A1 *	8/2004	Ogikubo	G11B 27/031 386/230
2004/0181444	A1	9/2004	Sandrew	
2005/0104878	A1	5/2005	Kaye et al.	
2005/0146521	A1	7/2005	Kaye et al.	
2005/0207623	A1 *	9/2005	Liu	G06K 9/00201 382/118
2005/0231501	A1	10/2005	Nitawaki	
2005/0231505	A1	10/2005	Kaye et al.	
2005/0280643	A1	12/2005	Chen	
2006/0028543	A1	2/2006	Sohn et al.	
2006/0061583	A1 *	3/2006	Spooner	G06T 15/20 345/582
2007/0052807	A1	3/2007	Zhou et al.	
2007/0279412	A1	12/2007	Davidson et al.	
2007/0279415	A1	12/2007	Sullivan et al.	
2007/0296721	A1	12/2007	Chang et al.	
2008/0181486	A1 *	7/2008	Spooner	G06T 7/0065 382/154
2008/0225040	A1 *	9/2008	Simmons	H04N 13/026 345/419
2008/0225042	A1 *	9/2008	Birtwistle	H04N 13/0022 345/419
2008/0225045	A1 *	9/2008	Birtwistle	G06T 7/0065 345/420
2008/0225059	A1 *	9/2008	Lowe	H04N 13/0497 345/626
2008/0226123	A1 *	9/2008	Birtwistle	G06T 11/001 382/100
2008/0226128	A1 *	9/2008	Birtwistle	G06T 7/0083 382/103
2008/0226160	A1 *	9/2008	Birtwistle	H04N 13/026 382/154
2008/0226181	A1 *	9/2008	Birtwistle	H04N 13/026 382/232
2008/0226194	A1 *	9/2008	Birtwistle	H04N 13/026 382/285
2008/0228449	A1 *	9/2008	Birtwistle	G06T 7/0071 703/1
2008/0246759	A1	10/2008	Summers	
2008/0246836	A1 *	10/2008	Lowe	G06T 7/004 348/46
2008/0259073	A1 *	10/2008	Lowe	G06T 7/0075 345/419
2009/0033741	A1	2/2009	Oh et al.	
2009/0116732	A1	5/2009	Zhou et al.	
2009/0256903	A1	10/2009	Spooner et al.	
2011/0050864	A1	3/2011	Bond	
2011/0164109	A1 *	7/2011	Baldridge	H04N 13/0267 348/43
2011/0169827	A1	7/2011	Spooner et al.	
2011/0169914	A1	7/2011	Lowe et al.	
2011/0227917	A1	9/2011	Lowe et al.	
2012/0032948	A1	2/2012	Lowe et al.	
2014/0169767	A1 *	6/2014	Goldberg	G11B 27/031 386/282

FOREIGN PATENT DOCUMENTS

JP	6052190	3/1985
JP	2004207985	7/2004
RU	2376632	12/2009
SU	1192168 A	11/1985

OTHER PUBLICATIONS

Interpolation (from Wikipedia encyclopedia, article pp. 1-6), retrieved from Internet URL: http://en.wikipedia.org/wiki/Interpolation on Jun. 5, 2008.

Optical Reader (from Wikipedia encyclopedia, article p. 1), retrieved from Internet URL:http://en.wikipedia.org/wiki/Optical_reader on Jun. 5, 2008.

Declaration of Steven K. Feiner, Exhibit A, 10 pages, Nov. 2, 2007.

Declaration of Michael F. Chou, Exhibit B, 12 pages, Nov. 2, 2007.

Declaration of John Marchioro, Exhibit C, 3 pages, Nov. 2, 2007.

Exhibit 1 to Declaration of John Marchioro, Revised translation of portions of Japanese Patent Document No. 60-52190 to Hiromae, 3 pages, Nov. 2, 2007.

U.S. Patent and Trademark Office, Before the Board of Patent Appeals and Interferences, Ex Parte Three-Dimensional Media Group, Ltd., Appeal 2009-004087, Reexamination Control No. 90/007,578, U.S. Pat. No. 4,925,294, Decision on Appeal, 88 pages, Jul. 30, 2010.

International Search Report dated May 10, 2012, 8 pages.

International Patentability Preliminary Report and Written Opinion /PCT/US2013/035506, dated Aug. 21, 2014, 6 pages.

Machine translation of JP Patent No. 2004-207985, dated Jul. 22, 2008, 34 pages.

Nell et al., "Stereographic Projections by Digital Computer", Computers and Automation for May 1965, pp. 32-34.

Nell, "Computer-Generated Three-Dimensional Movies" Computers and Automation for Nov. 1965, pp. 20-23.

U.S. Patent and Trademark Office, Before the Board of Patent Appeals and Interferences, Ex Parte Three-Dimensional Media Group, Ltd., Appeal 2009-004087, Reexamination Control No. 90/007,578, U.S. Pat. No. 4,925,294, Decision on Appeal, 88 pages.

International Search Report received fro PCT Application No. PCT/US2011/067024, dated Aug. 22, 2012, 10 pages.

European Office Action dated Jun. 26, 2013, received for EP Appl. No. 02734203.9 on Jul. 22, 2013, 5 pages.

PCT Search Report and Written Opinion, dated Aug. 22, 2013, for PCT Appl. No. PCT/US2013/035506, 7 pages.

IPRP, PCT/US2013/035506, dated Aug. 21, 2014, 6 pages.

Office Action for EPO Patent Application No. 02 734 203.9 dated Sep. 12, 2006. (4 pages).

Office Action for AUS Patent Application No. 2002305387 dated Mar. 9, 2007. (2 pages).

Office Action for EPO Patent Application No. 02 734 203.9 dated Oct. 7, 2010. (5 pages).

First Examination Report for Indian Patent Application No. 01779/DELNP/2003 dated Mar. 2004. (4 pages).

International Search Report Dated Jun. 13, 2003. (3 pages).

Declaration of Barbara Frederiksen in Support of In-Three, Inc's Opposition to Plaintiffs Motion for Preliminary Injunction, Aug. 1, 2005, Imax Corporation et al v. In-Three, Inc., Case No. CV05 1795 FMC (Mcx). (25 pages).

USPTO, Board of Patent Appeals and Interferences, Decision on Appeal dated Jul. 30, 2010, Ex parte Three-Dimensional Media Group, Ltd., Appeal 2009-004087, Reexamination Control No. 90/007,578, U.S. Pat. No. 4,925,294. (88 pages).

Office Action for Canadian Patent Application No. 2,446,150 dated Oct. 8, 2010. (6 pages).

Office Action for Canadian Patent Application No. 2,446,150 dated Jun. 13, 2011. (4 pages).

(56) **References Cited**

OTHER PUBLICATIONS

Murray et al., Active Tracking, IEEE International Conference on Intelligent Robots and Systems, Sep. 1993, pp. 1021-1028.
Gao et al., Perceptual Motion Tracking from Image Sequences, IEEE, Jan. 2001, pp. 389-392.
Yasushi Mae, et al., "Object Tracking in Cluttered Background Based on Optical Flow and Edges," Proc. 13th Int. Conf. on Pattern Recognition, vol. 1, pp. 196-200, Apr. 1996.
Di Zhong, Shih-Fu Chang, "AMOS: An Active System for MPEG-4 Video Object Segmentation," ICIP (2) 8: 647-651, Apr. 1998.
Hua Zhong, et al., "Interactive Tracker—A Semi-automatic Video Object Tracking and Segmentation System," Microsoft Research China, http://research.microsoft.com (Aug. 26, 2003).
Eric N. Mortensen, William A. Barrett, "Interactive segmentation with Intelligent Scissors," Graphical Models and Image Processing, v.60 n. 5, p. 349-384, Sep. 2002.
Michael Gleicher, "Image Snapping," SIGGRAPH: 183-190, Jun. 1995.
Joseph Weber, et al., "Rigid Body Segmentation and Shape Description . . . ," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, No. 2, Feb. 1997,pp. 139-143.
E. N. Mortensen and W. A. Barrett, "Intelligent Scissors for Image Composition," Computer Graphics (SIGGRAPH '95), pp. 191-198, Los Angeles, CA, Aug. 1995.
Ohm et al., An Object-Based System for Stereopscopic Viewpoint Synthesis, IEEE transaction on Circuits and Systems for Video Technology, vol. 7, No. 5, Oct. 1997, pp. 801-811.
Izquierdo et al., Virtual 3D-View Generation from Stereoscopic Video Data, IEEE, Jan. 1998, pp. 1219-1224.
Kaufman, D., "The Big Picture", Apr. 1998, http://www.xenotech.com Apr. 1998, pp. 1-4.
Hanrahan et al., "Direct WYSIWYG painting and texturing on 3D shapes", Computer Graphics, vol. 24, Issue 4, pp. 215-223. Aug. 1990.
Grossman, "Look Ma, No Glasses", Games, Apr. 1992, pp. 12-14.
Slinker et al., "The Generation and Animation of Random Dot and Random Line Autostereograms", Journal of Imaging Science and Technology, vol. 36, No. 3, pp. 260-267, May 1992.
A. Michael Noll, Stereographic Projections by Digital Computer, Computers and Automation, vol. 14, No. 5 (May 1965), pp. 32-34.
A. Michael Noll, Computer-Generated Three-Dimensional Movies, Computers and Automation, vol. 14, No. 11 (Nov. 1965), pp. 20-23.
Selsis et al., Automatic Tracking and 3D Localization of Moving Objects by Active Contour Models, Intelligent Vehicles 95 Symposium, Sep. 1995, pp. 96-100.
Smeulders et al., Tracking Nonparameterized Object Contours in Video, IEEE Transactions on Image Processing, vol. 11, No. 9, Sep. 2002, pp. 1081-1091.

* cited by examiner

Figure 1

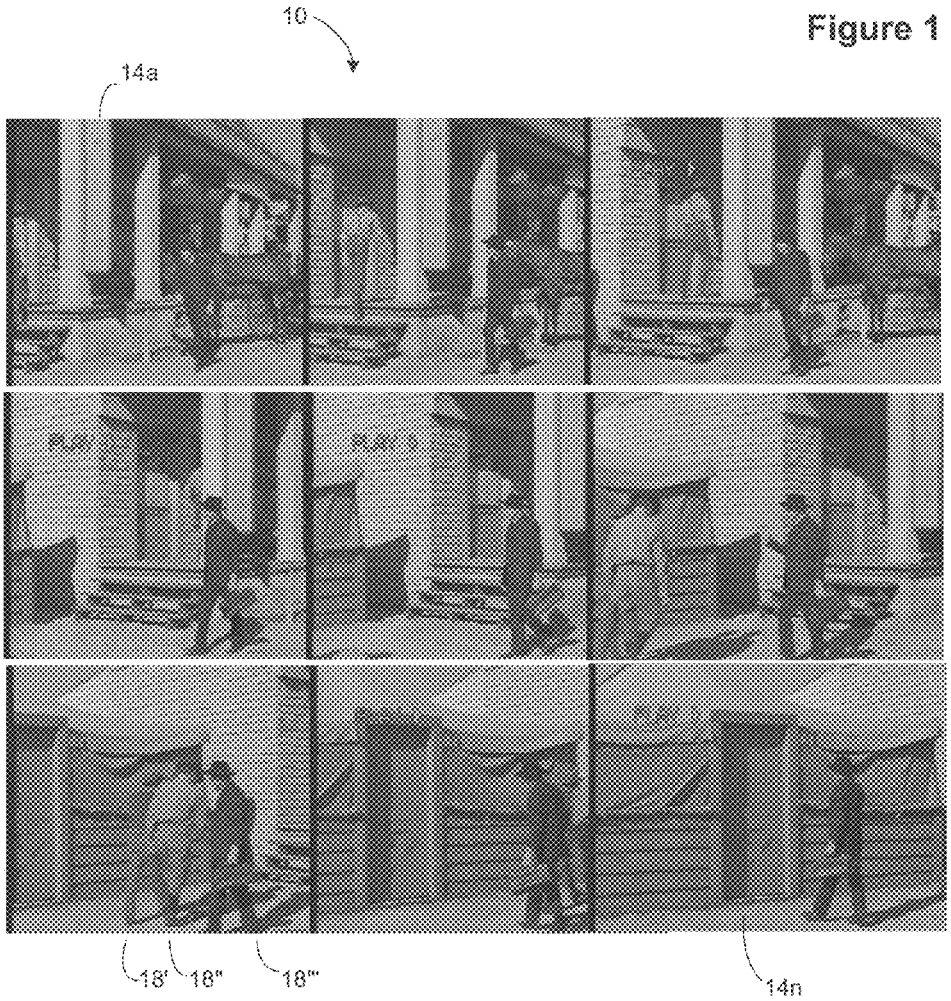


Figure 2

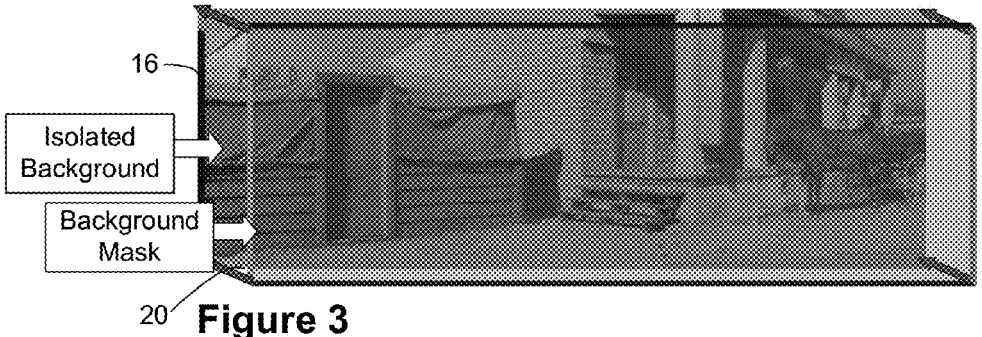


Figure 3

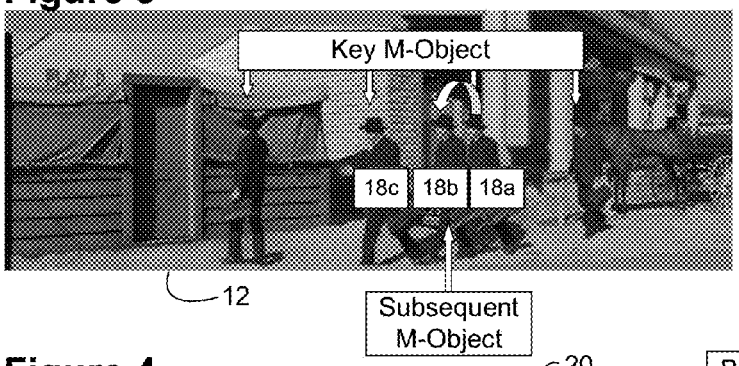


Figure 4

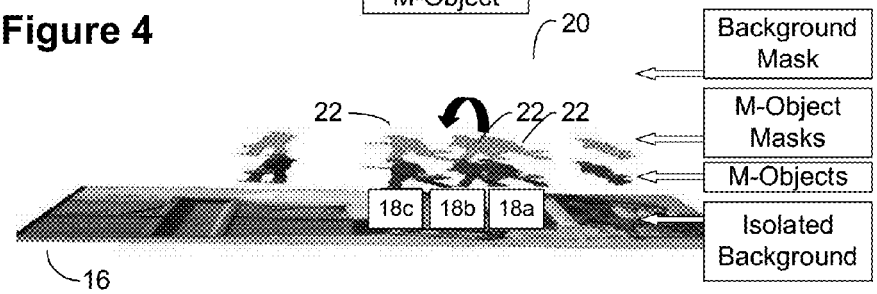


Figure 5A

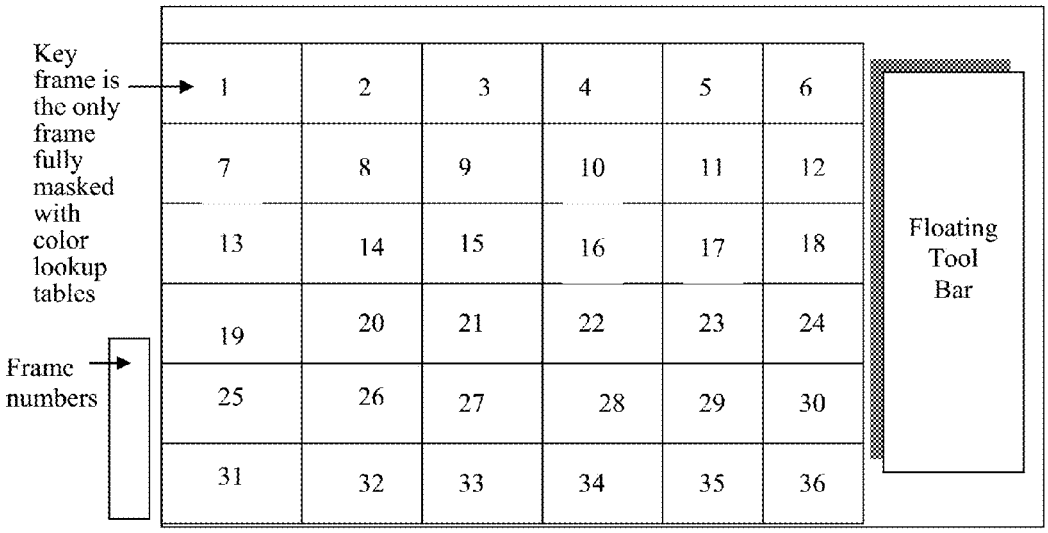


Figure 5B

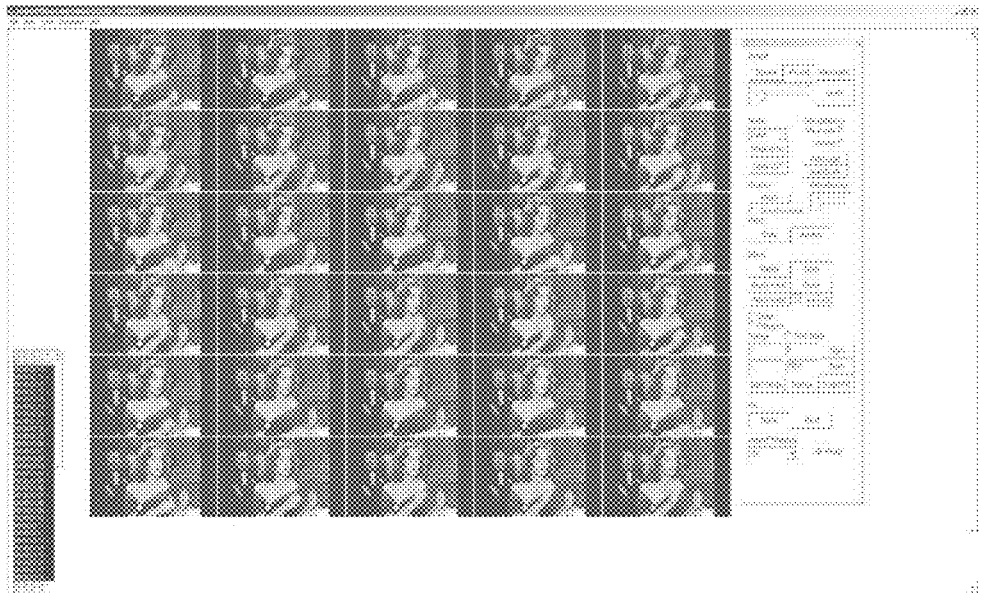


Figure 6A

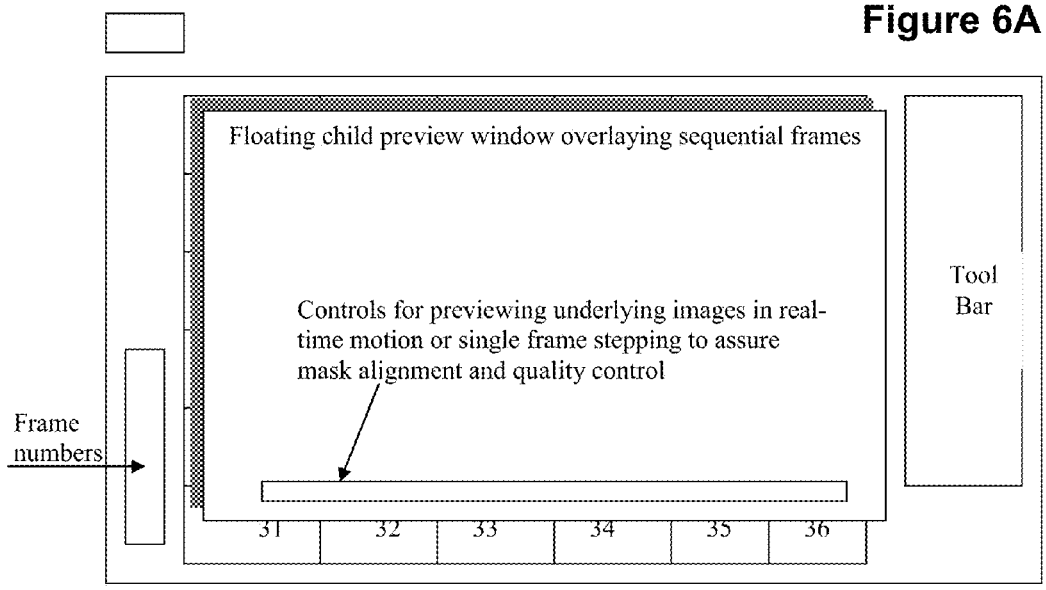


Figure 6B

Figure 7A

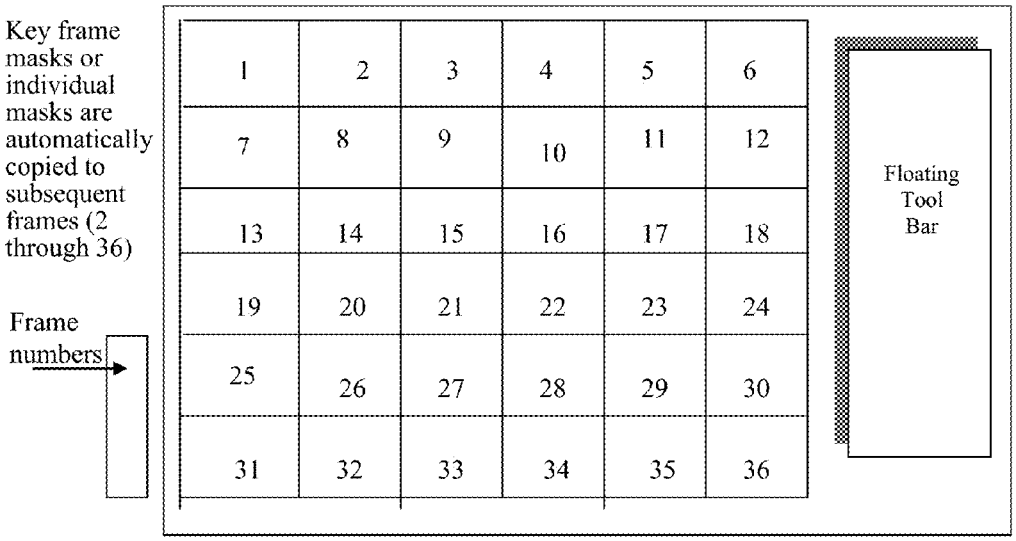


Figure 7B

Figure 8

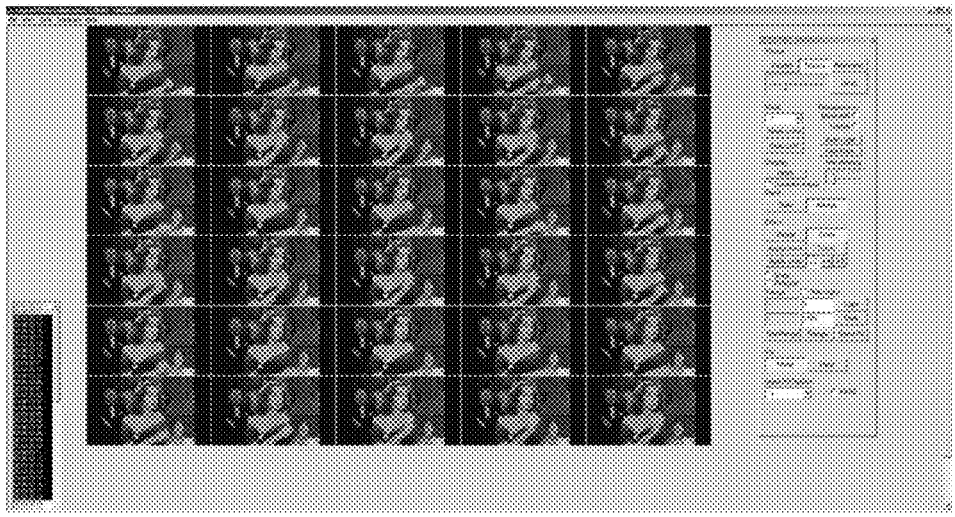


Figure 9A

Figure 9B

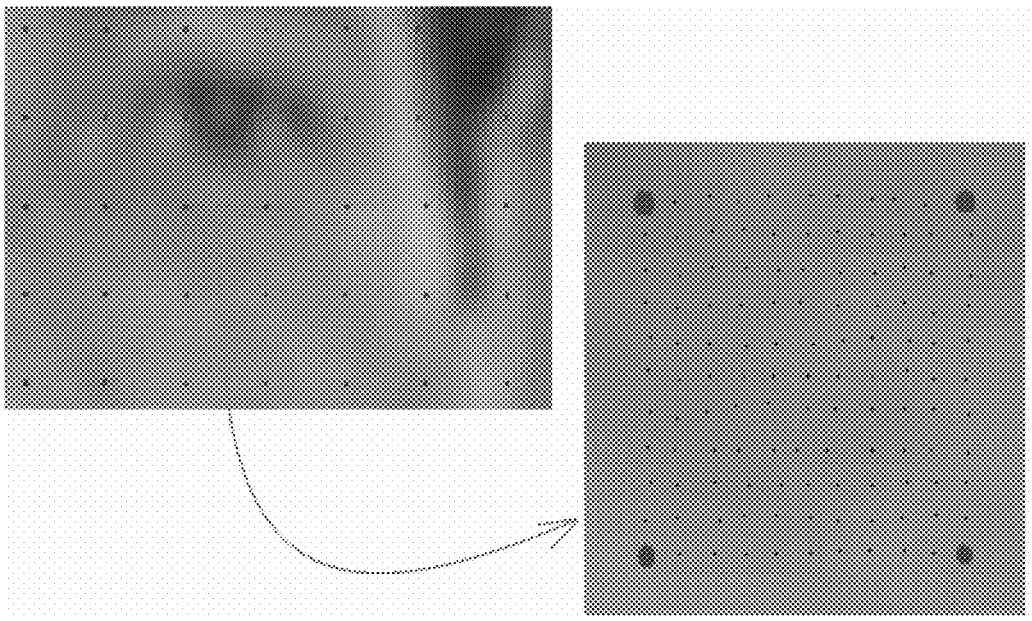


Figure 10A

Figure 10B

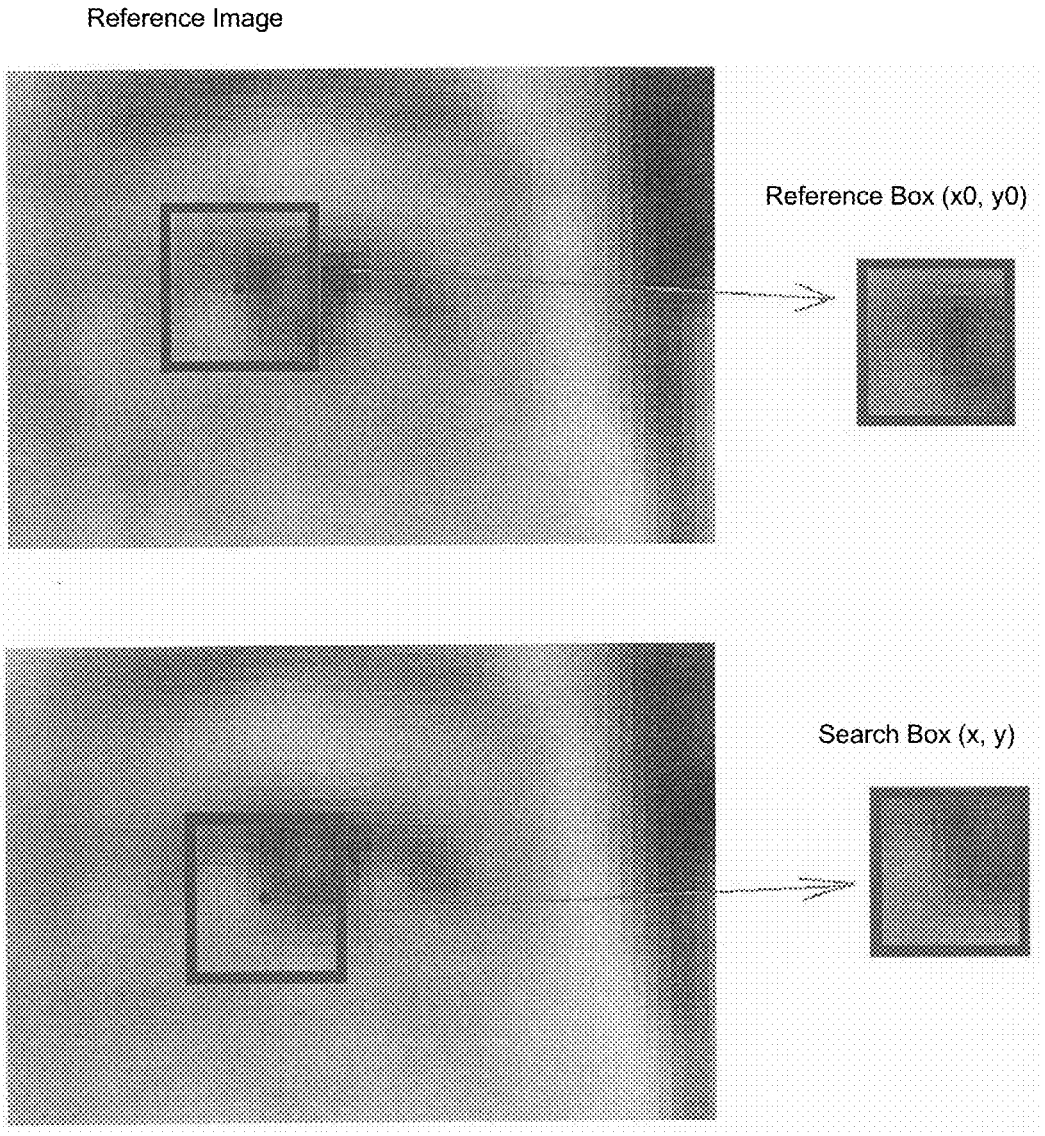


Figure 10C

Figure 10D

Figure 11B

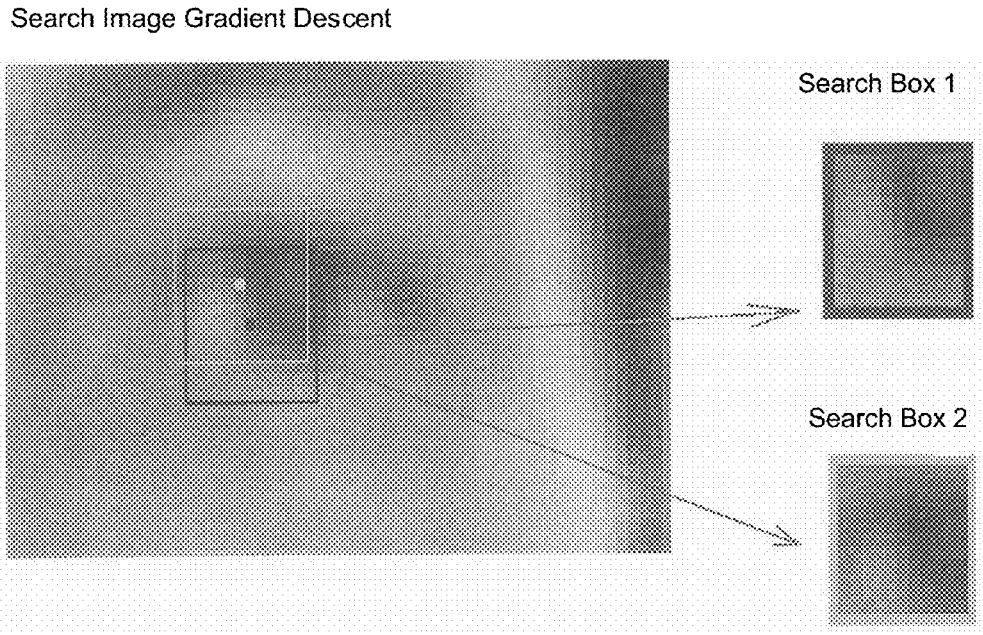


Figure 11A

Figure 11C

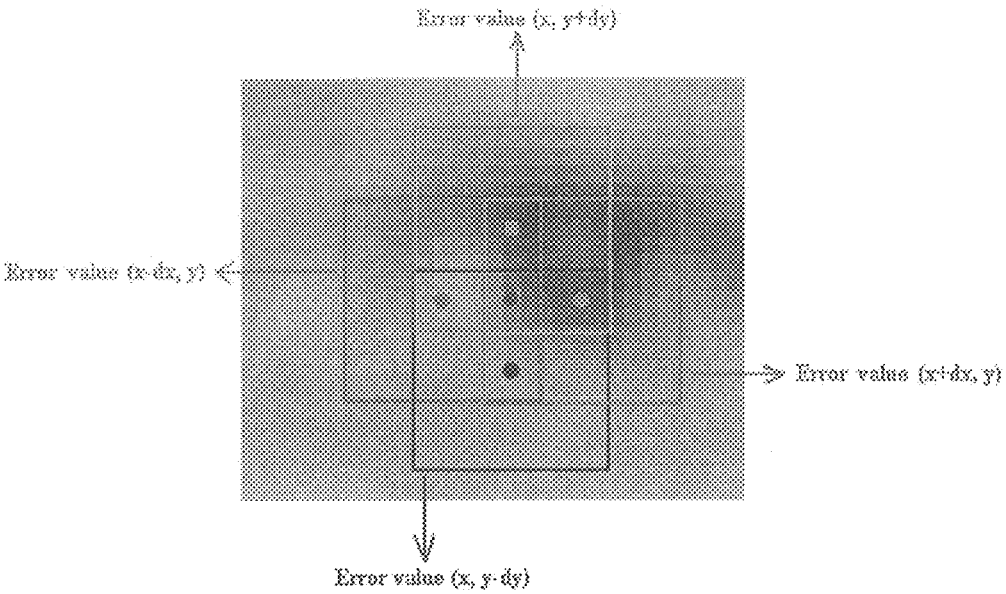

Error value (x, y+dy)
Error value (x-dx, y)
Error value (x+dx, y)
Error value (x, y-dy)

Figure 12

Figure 13

Figure 14

Figure 15

Figure 16

Figure 17

Figure 18

Figure 19

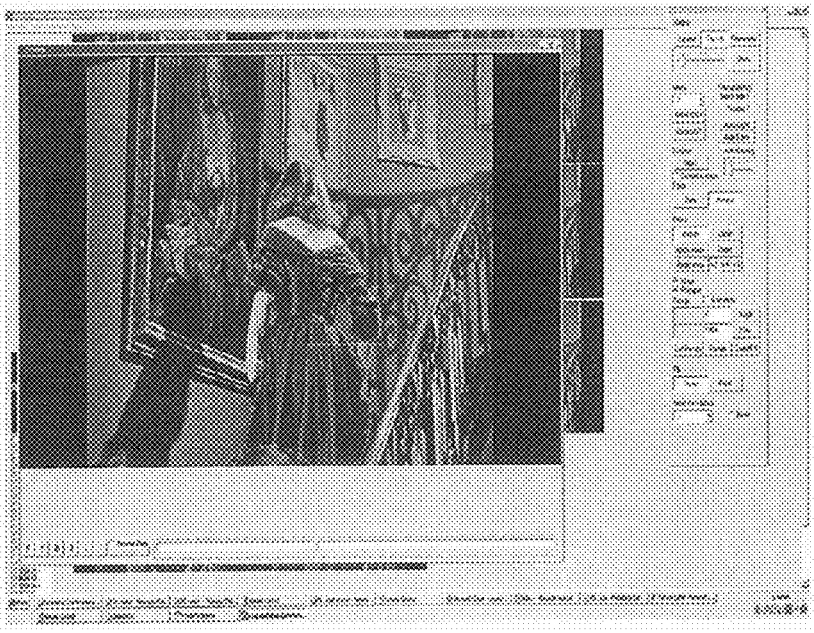


Figure 20

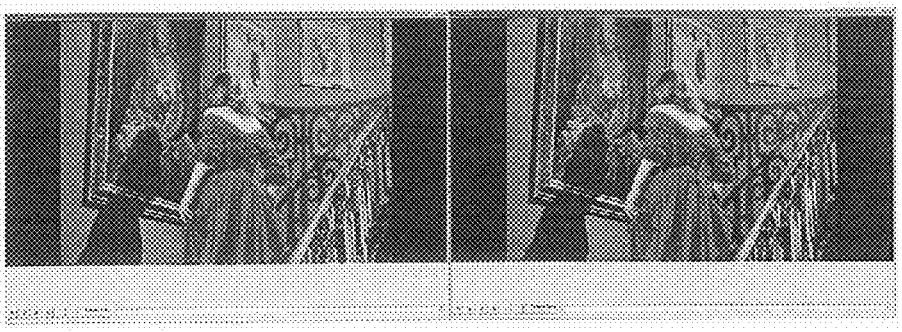


Figure 21

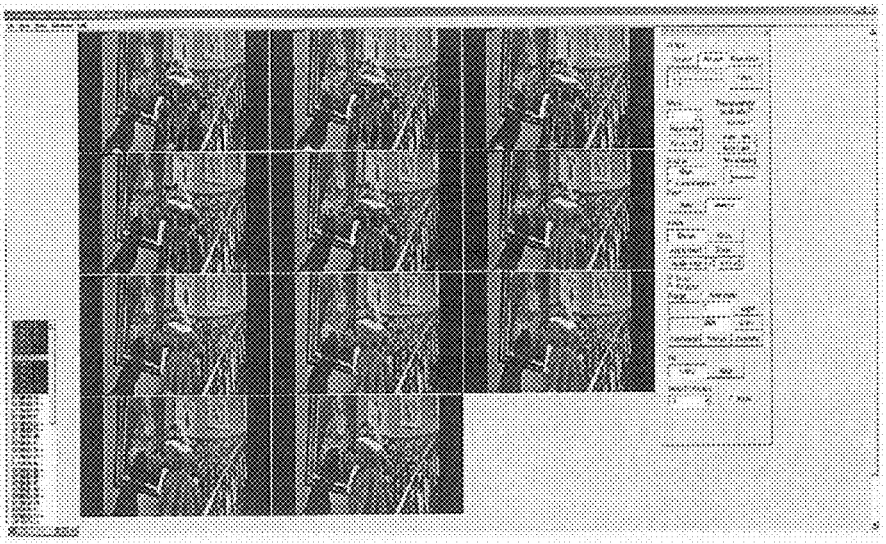


Figure 22

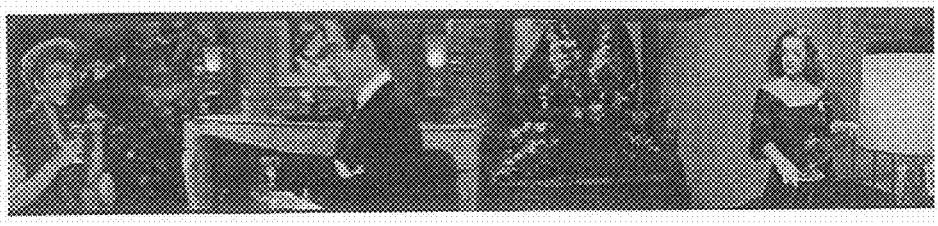


Figure 23

Figure 24

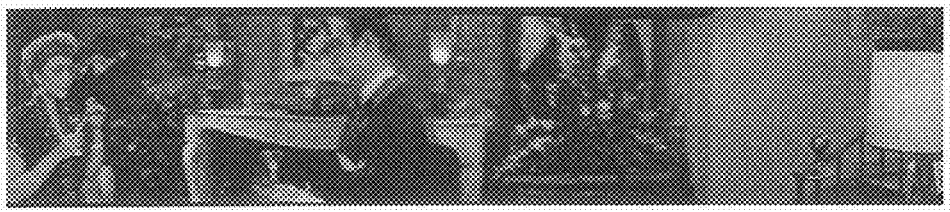


Figure 25

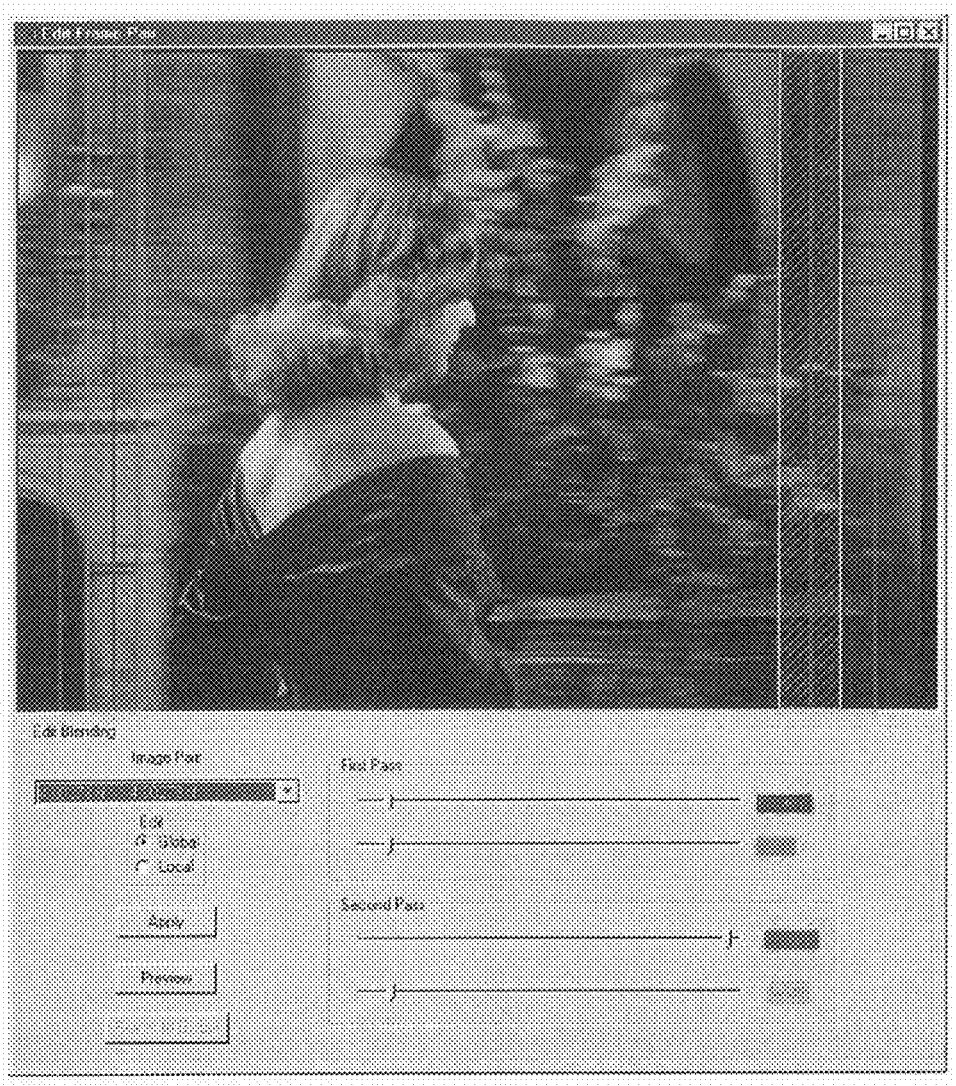

Edit Blending
Image Pair
Edit
Global
Local
Apply
Preview
First Pass
Second Pass

Figure 26

Figure 27

Figure 28

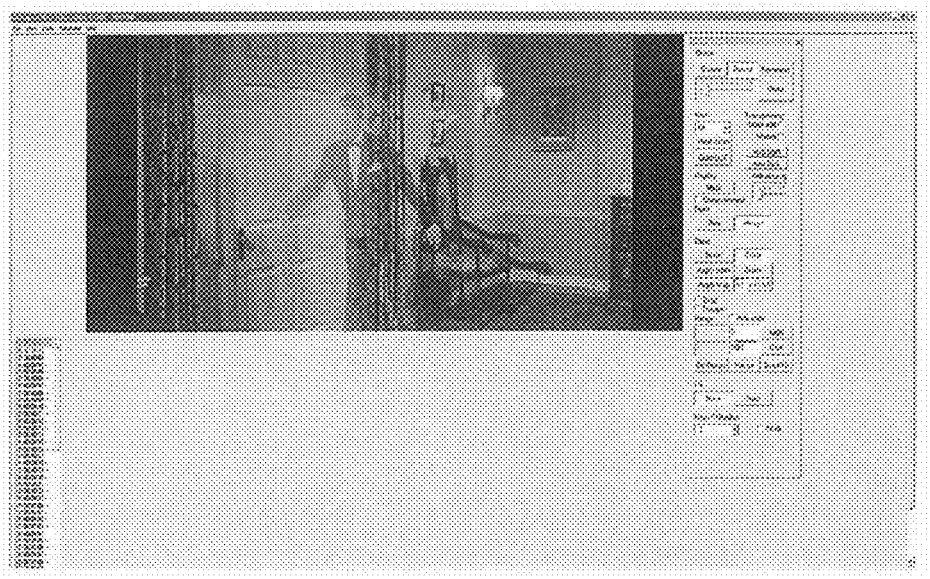


Figure 29

Figure 30

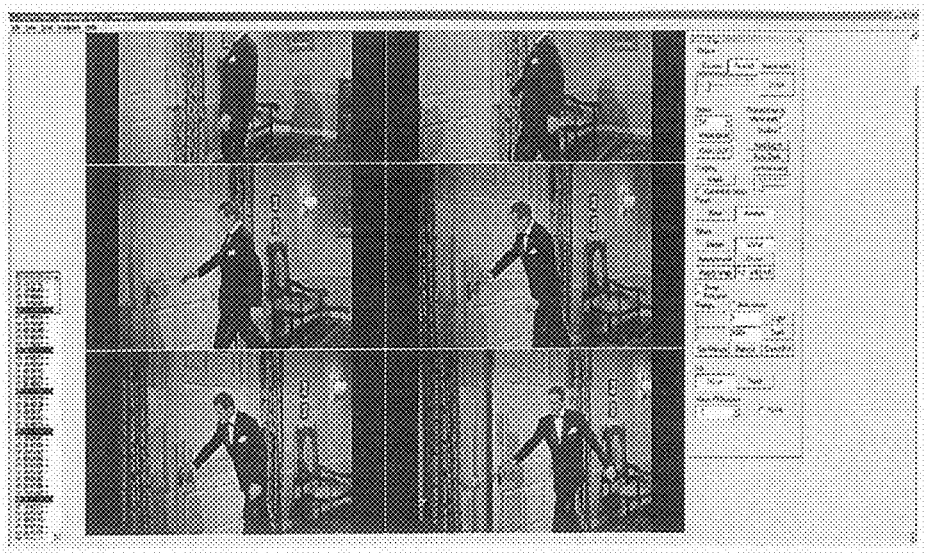


Figure 31

Figure 32

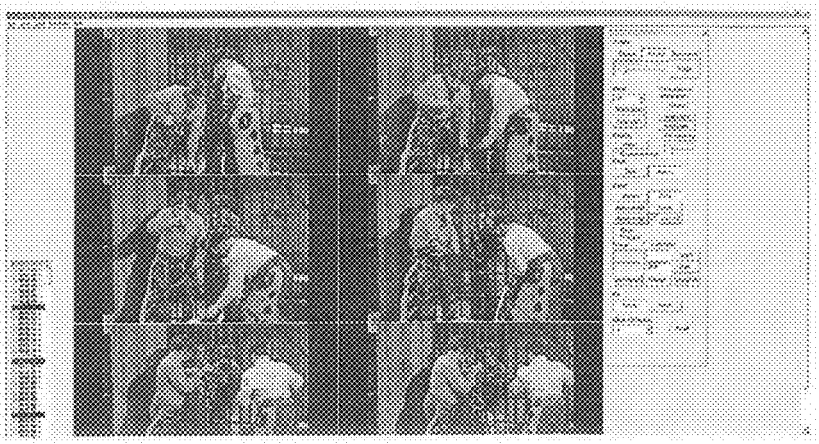


Figure 33

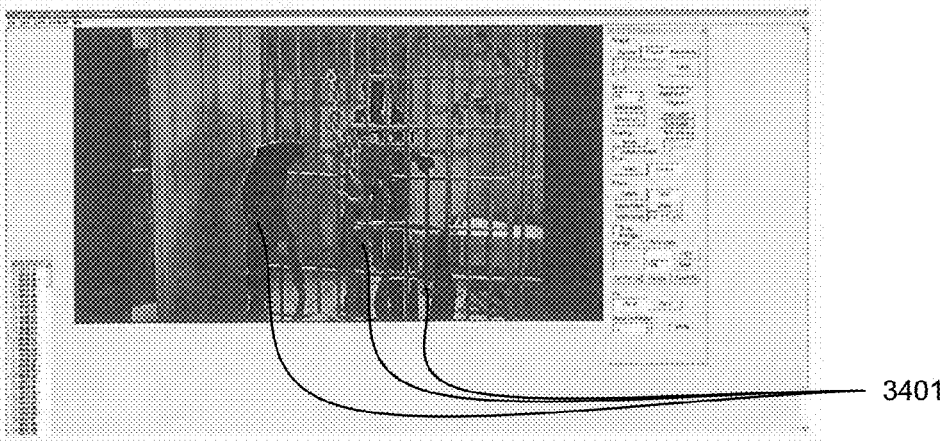


Figure 34

Figure 35

Figure 36

Figure 37A

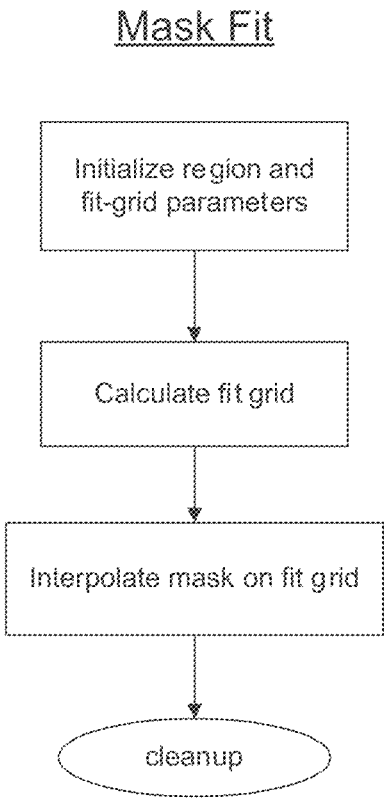

Mask Fit
Initialize region and fit-grid parameters
Calculate fit grid
Interpolate mask on fit grid
cleanup

Figure 37B

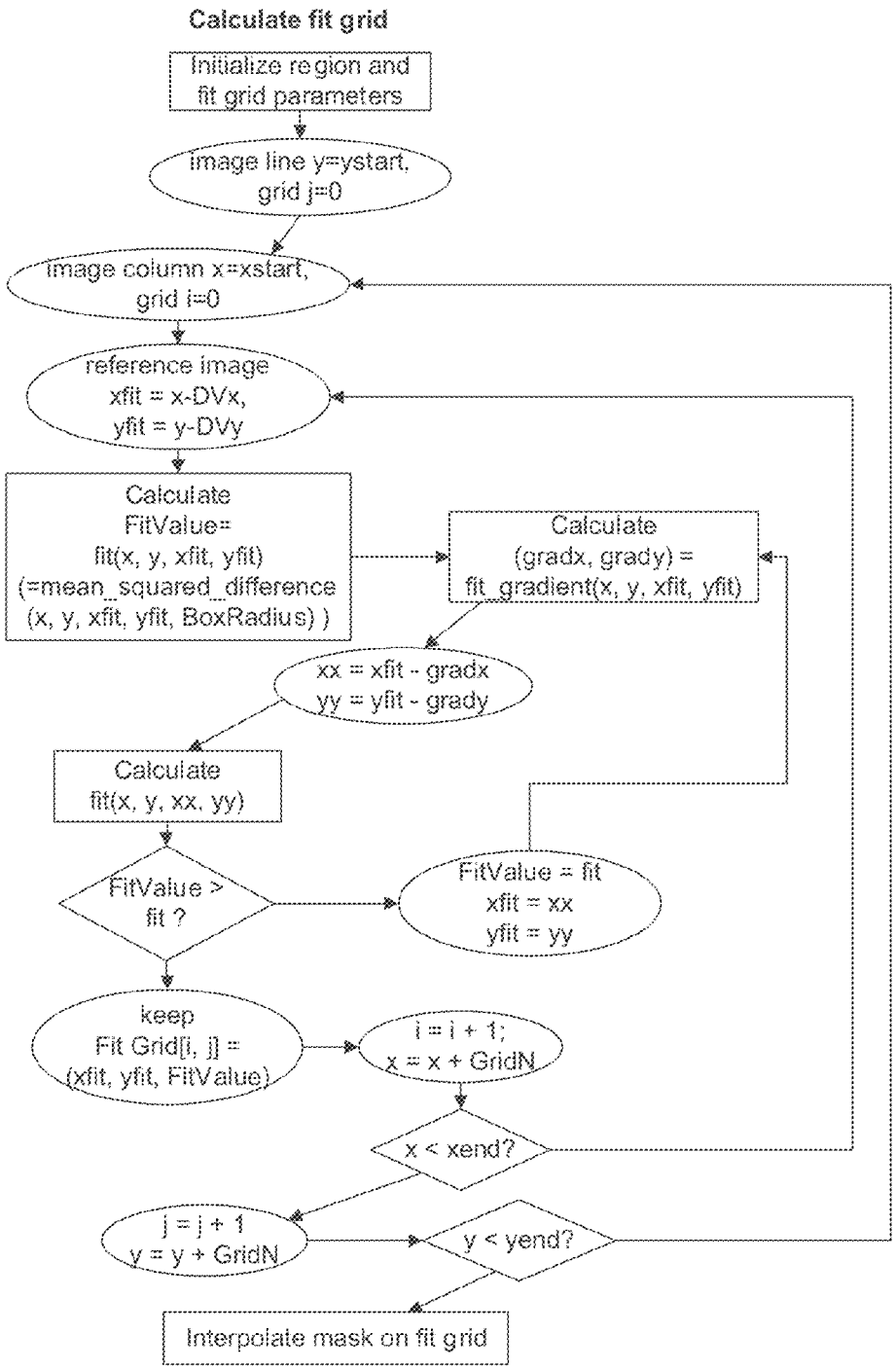

Calculate fit grid
Initialize region and fit grid parameters
image line y=ystart, grid j=0
image column x=xstart, grid i=0
reference image xfit = x-DVx, yfit = y-DVy
Calculate FitValue= fit(x, y, xfit, yfit) (=mean_squared_difference (x, y, xfit, yfit, BoxRadius))
Calculate (gradx, grady) = fit_gradient(x, y, xfit, yfit)
xx = xfit - gradx yy = yfit - grady
Calculate fit(x, y, xx, yy)
FitValue > fit ?
FitValue = fit xfit = xx yfit = yy
keep Fit Grid[i, j] = (xfit, yfit, FitValue)
i = i + 1; x = x + GridN
x < xend?
j = j + 1 y = y + GridN
y < yend?
Interpolate mask on fit grid

Figure 37C

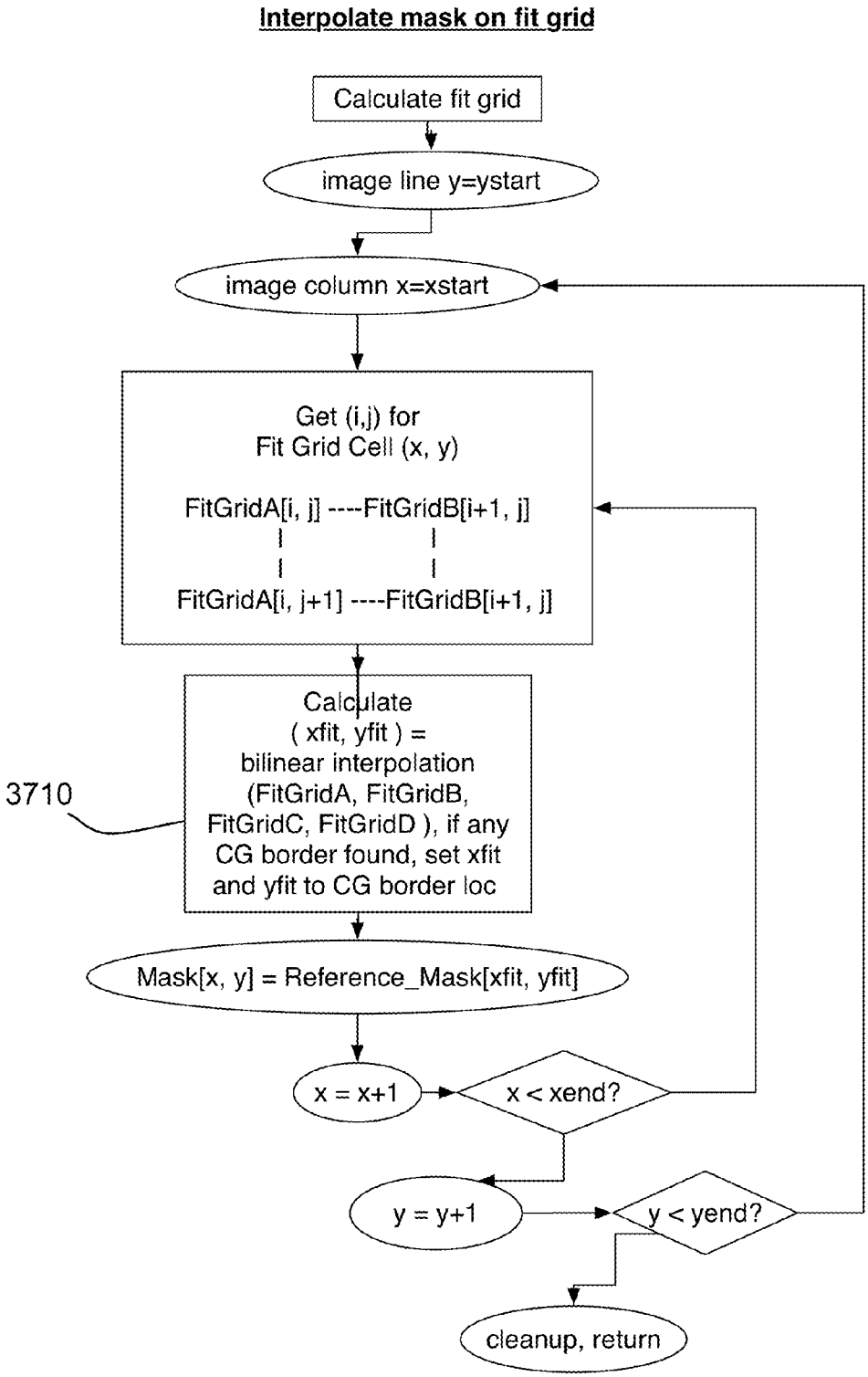

Interpolate mask on fit grid
Calculate fit grid
image line y=ystart
image column x=xstart
Get (i,j) for
Fit Grid Cell (x, y)
FitGridA[i, j] ----FitGridB[i+1, j]
FitGridA[i, j+1] ----FitGridB[i+1, j]
3710
Calculate
(xfit, yfit) =
bilinear interpolation
(FitGridA, FitGridB,
FitGridC, FitGridD), if any
CG border found, set xfit
and yfit to CG border loc
Mask[x, y] = Reference_Mask[xfit, yfit]
x = x+1
x < xend?
y = y+1
y < yend?
cleanup, return

Figure 38A

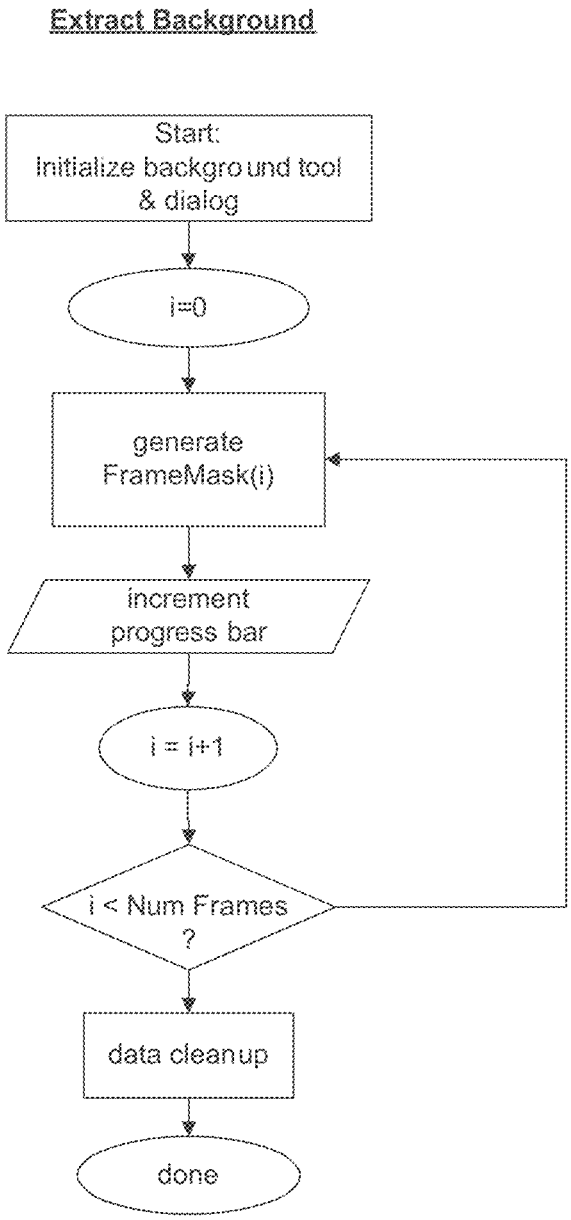

Extract Background
Start:
Initialize backgro und tool
& dialog
i=0
generate
FrameMask(i)
increment
progress bar
i = i+1
i < Num Frames
?
data cleanup
done

Figure 38B

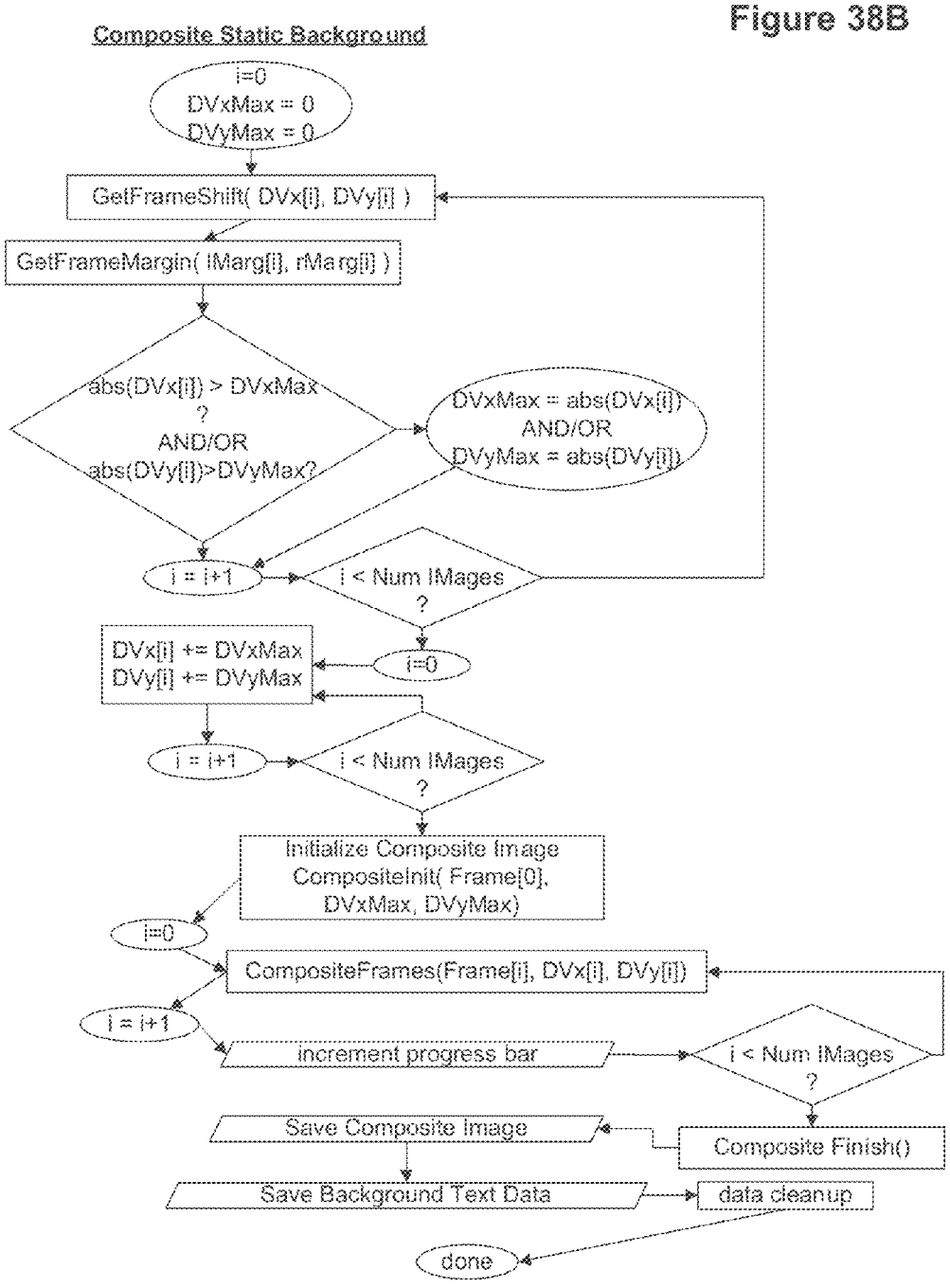

Composite Static Background
i=0
DVxMax = 0
DVyMax = 0
GetFrameShift(DVx[i], DVy[i])
GetFrameMargin(lMarg[i], rMarg[i])
abs(DVx[i]) > DVxMax ? AND/OR abs(DVy[i])>DVyMax?
DVxMax = abs(DVx[i]) AND/OR DVyMax = abs(DVy[i])
i = i+1
i < Num IMages ?
DVx[i] += DVxMax
DVy[i] += DVyMax
i=0
i = i+1
i < Num IMages ?
Initialize Composite Image
CompositeInit(Frame[0], DVxMax, DVyMax)
i=0
CompositeFrames(Frame[i], DVx[i], DVy[i])
i = i+1
increment progress bar
i < Num IMages ?
Save Composite Image
Composite Finish()
Save Background Text Data
data cleanup
done

Figure 39A

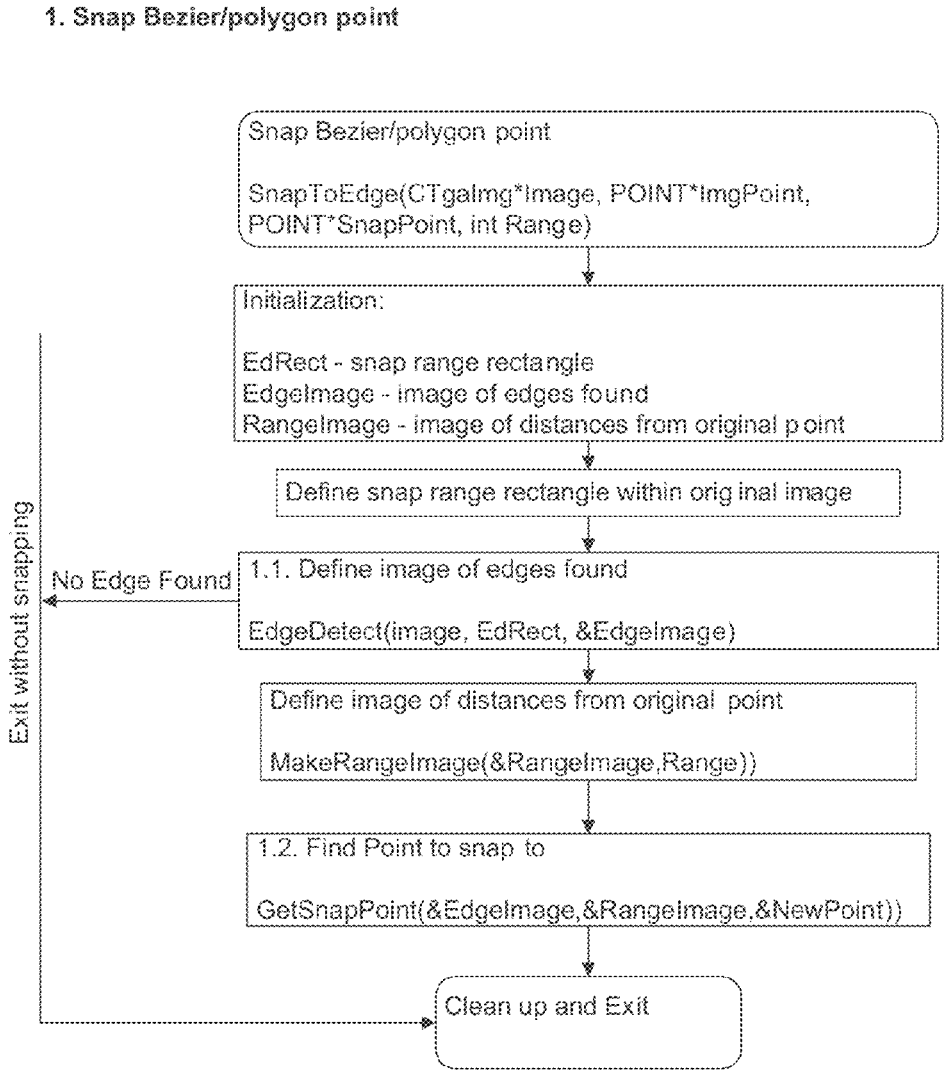

1. Snap Bezier/polygon point
Snap Bezier/polygon point
SnapToEdge(CTgaImg*Image, POINT*ImgPoint, POINT*SnapPoint, int Range)
Initialization:
EdRect - snap range rectangle
EdgeImage - image of edges found
RangeImage - image of distances from original point
Define snap range rectangle within original image
1.1. Define image of edges found
EdgeDetect(image, EdRect, &EdgeImage)
No Edge Found
Exit without snapping
Define image of distances from original point
MakeRangeImage(&RangeImage,Range))
1.2. Find Point to snap to
GetSnapPoint(&EdgeImage,&RangeImage,&NewPoint))
Clean up and Exit

Figure 39B

1.1. Define image of edges found

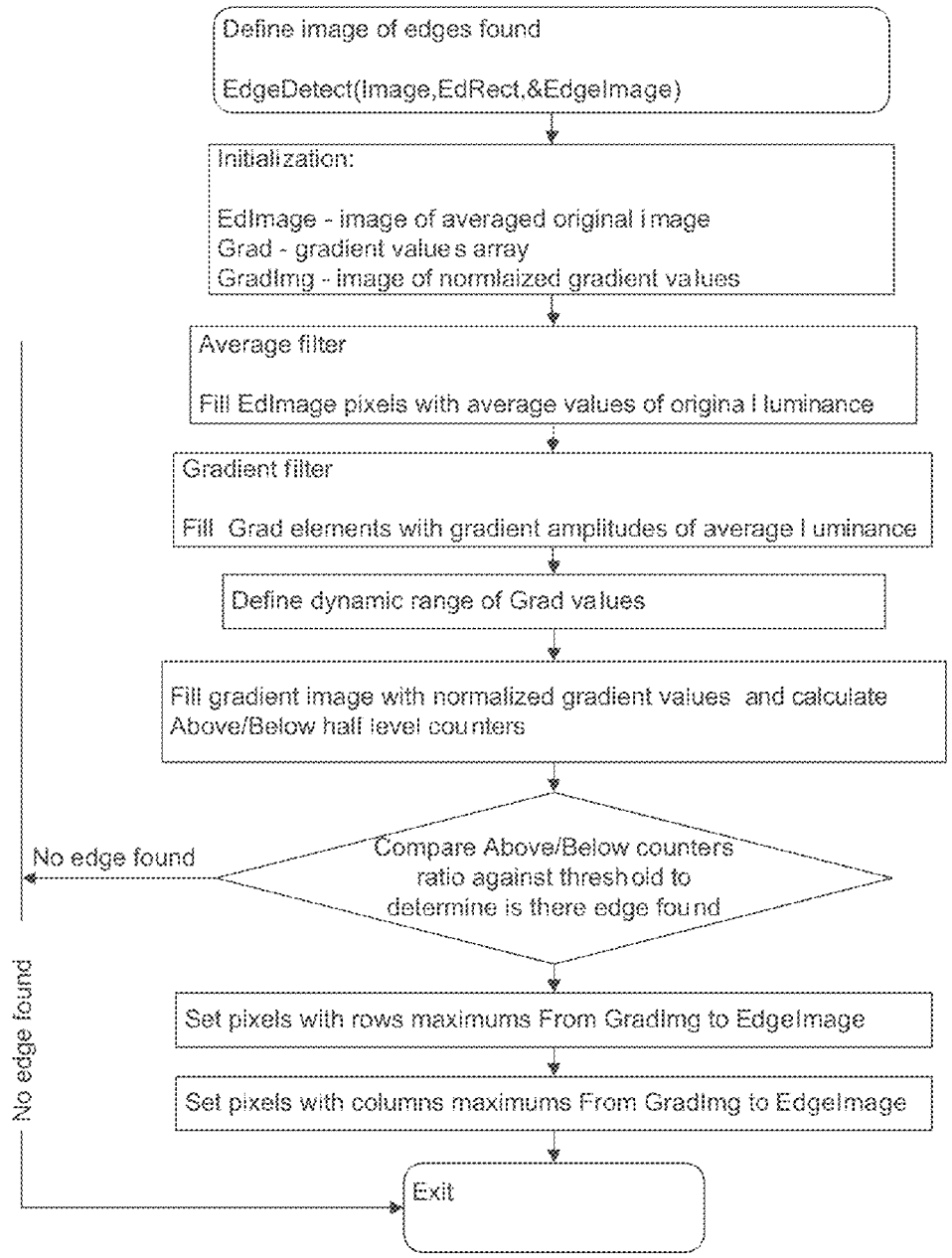


Figure 39C

1.2. Find Point to snap to

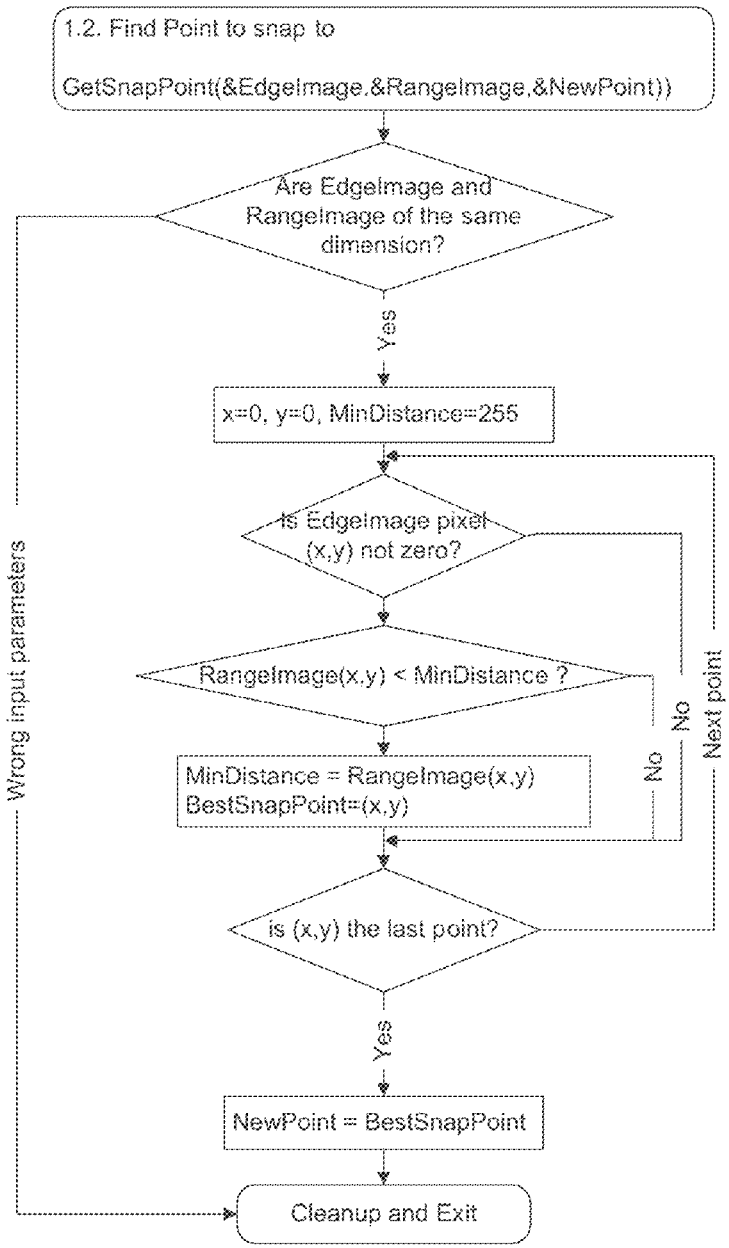


Figure 40A

2. Relative bimodal threshold tool

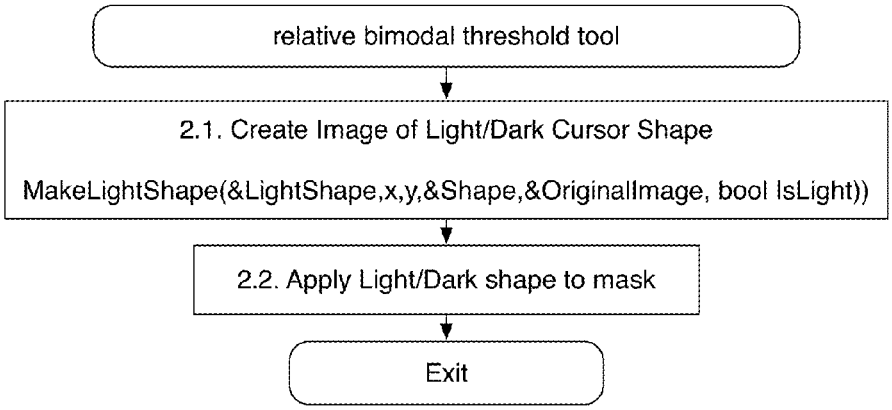


Figure 40B

2.2 Apply Light/Dark shape to mask

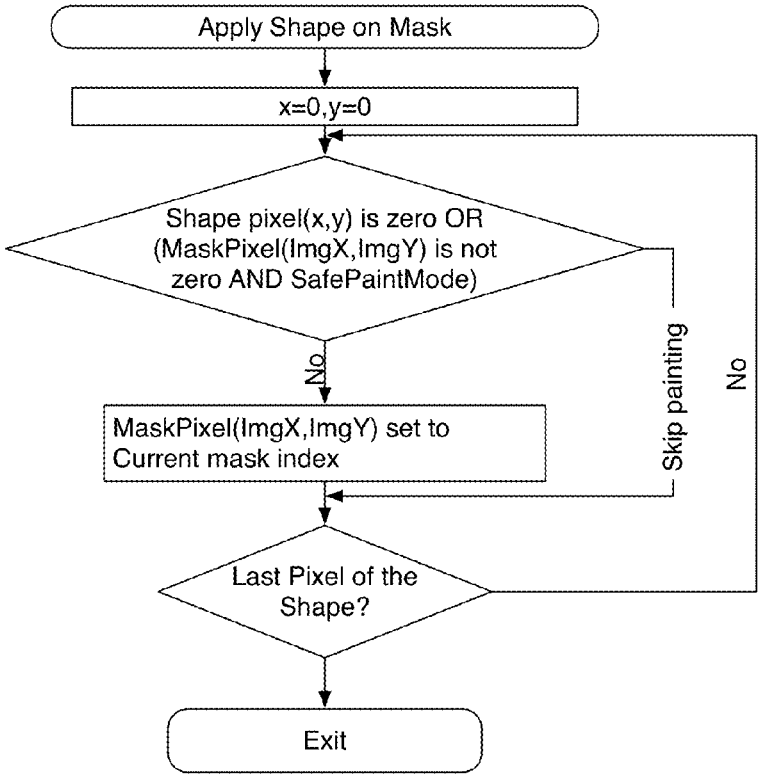


Figure 40C

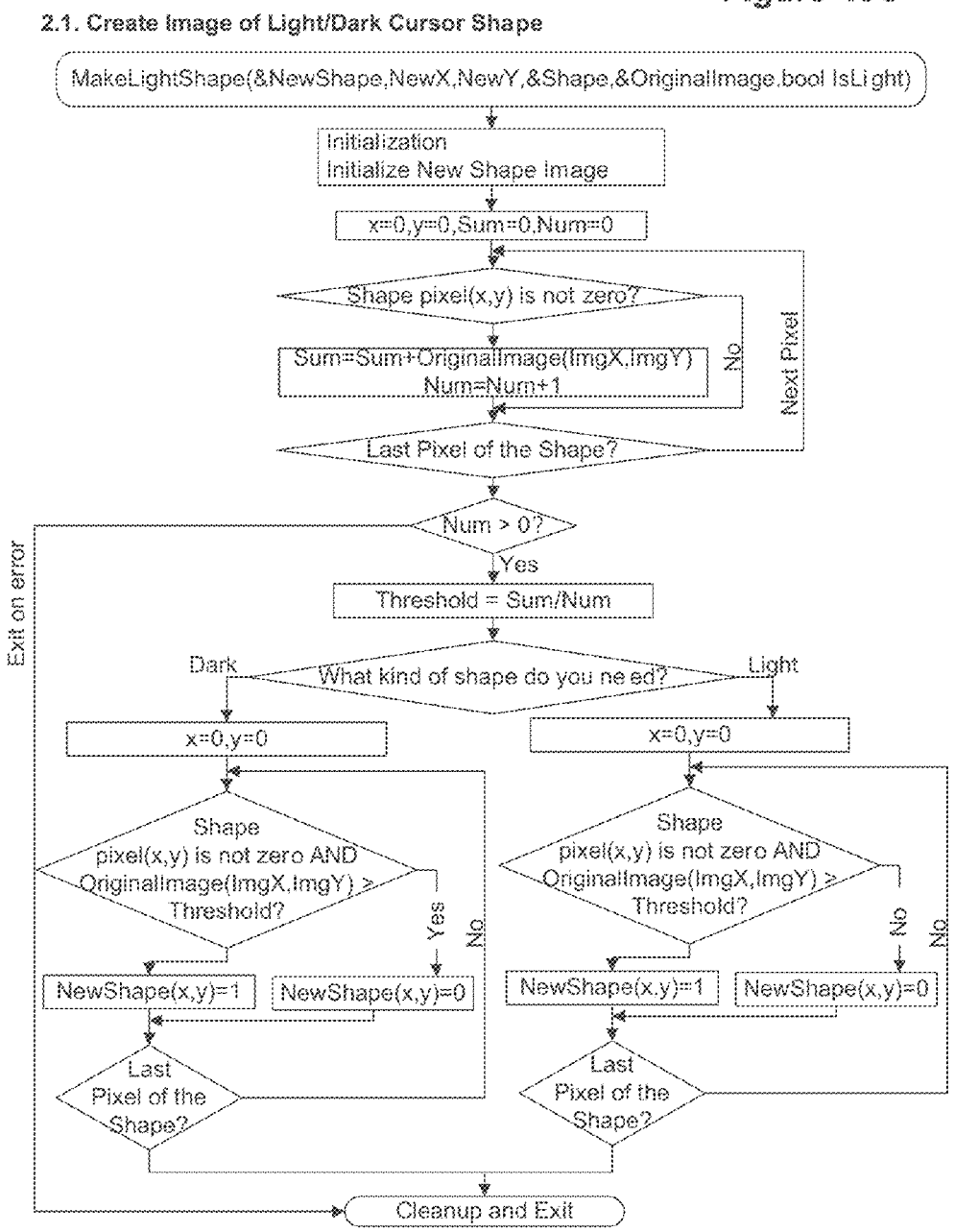

2.1. Create Image of Light/Dark Cursor Shape
MakeLightShape(&NewShape,NewX,NewY,&Shape,&OriginalImage,bool IsLight)
Initialization
Initialize New Shape Image
x=0,y=0,Sum=0,Num=0
Shape pixel(x,y) is not zero?
No
Sum=Sum+OriginalImage(ImgX,ImgY)
Num=Num+1
Next Pixel
Last Pixel of the Shape?
Num > 0?
Yes
Exit on error
Threshold = Sum/Num
Dark
What kind of shape do you need?
Light
x=0,y=0
x=0,y=0
Shape pixel(x,y) is not zero AND OriginalImage(ImgX,ImgY) > Threshold?
Yes
No
Shape pixel(x,y) is not zero AND OriginalImage(ImgX,ImgY) > Threshold?
No
No
NewShape(x,y)=1
NewShape(x,y)=0
NewShape(x,y)=1
NewShape(x,y)=0
Last Pixel of the Shape?
Last Pixel of the Shape?
Cleanup and Exit

Figure 41A

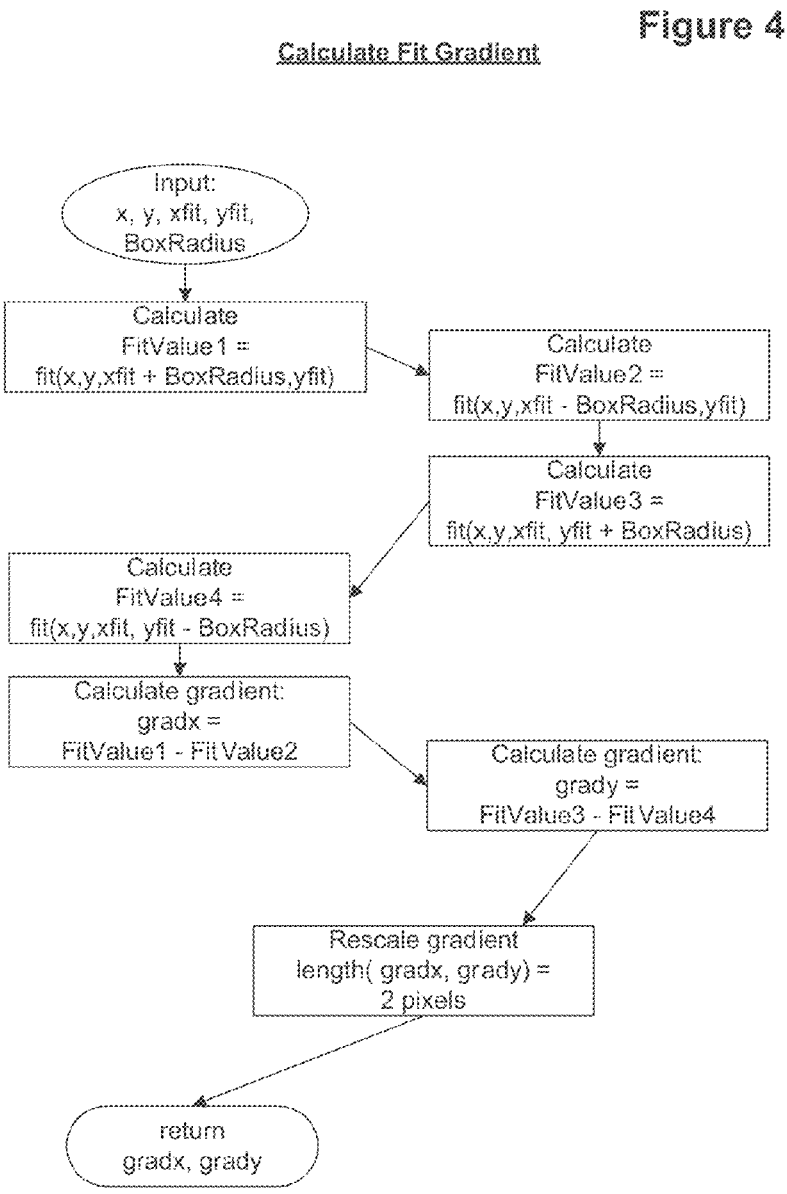

Calculate Fit Gradient
Input:
x, y, xfit, yfit,
BoxRadius
Calculate
FitValue1 =
fit(x,y,xfit + BoxRadius,yfit)
Calculate
FitValue2 =
fit(x,y,xfit - BoxRadius,yfit)
Calculate
FitValue3 =
fit(x,y,xfit, yfit + BoxRadius)
Calculate
FitValue4 =
fit(x,y,xfit, yfit - BoxRadius)
Calculate gradient:
gradx =
FitValue1 - FitValue2
Calculate gradient:
grady =
FitValue3 - FitValue4
Rescale gradient
length(gradx, grady) =
2 pixels
return
gradx, grady

Figure 41B

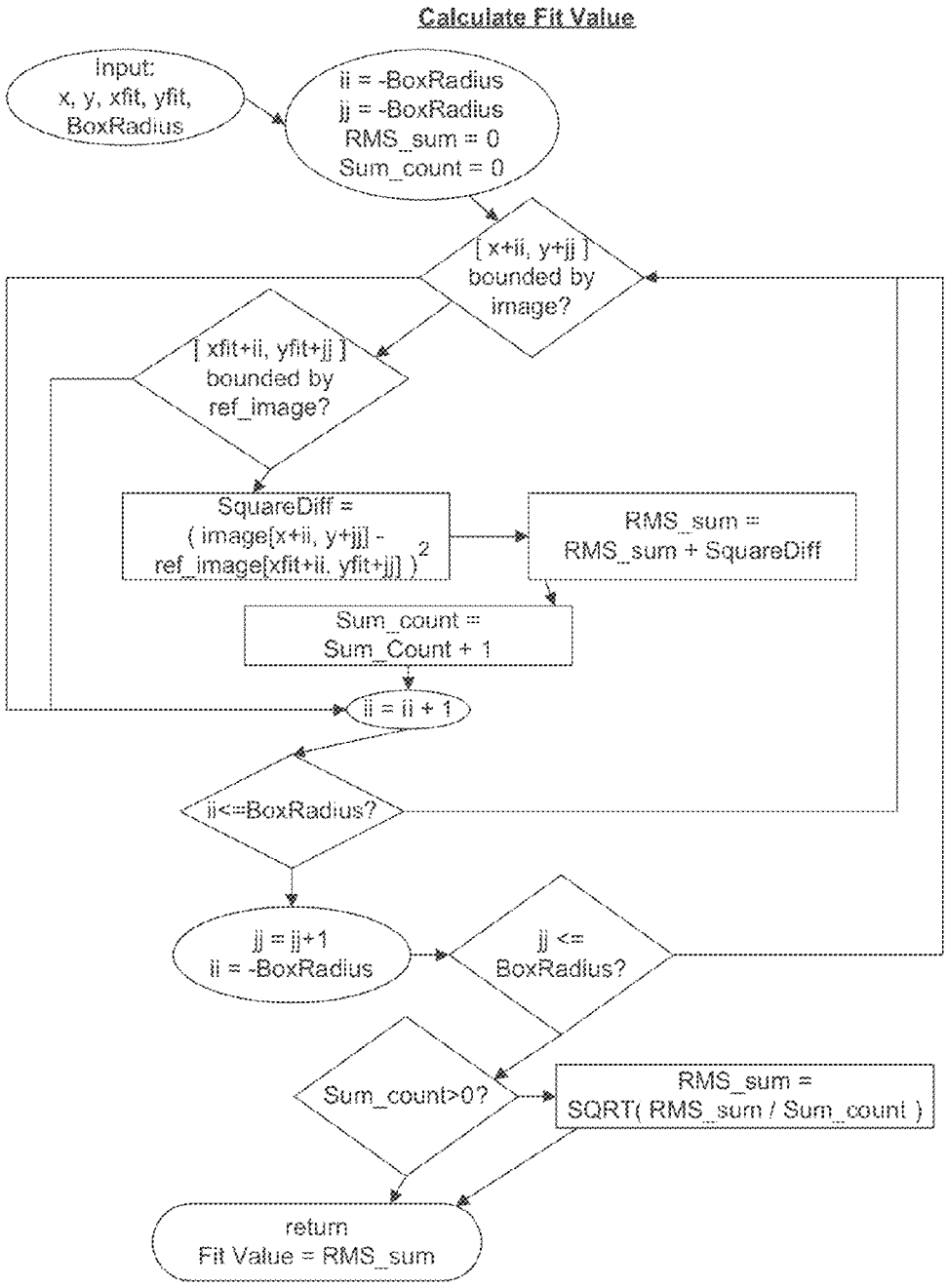

Calculate Fit Value
Input:
x, y, xfit, yfit,
BoxRadius
ii = -BoxRadius
jj = -BoxRadius
RMS_sum = 0
Sum_count = 0
[x+ii, y+jj]
bounded by
image?
[xfit+ii, yfit+jj]
bounded by
ref_image?
SquareDiff =
(image[x+ii, y+jj] -
ref_image[xfit+ii, yfit+jj])2
RMS_sum =
RMS_sum + SquareDiff
Sum_count =
Sum_Count + 1
ii = ii + 1
ii<=BoxRadius?
jj = jj+1
ii = -BoxRadius
jj <=
BoxRadius?
Sum_count>0?
RMS_sum =
SQRT(RMS_sum / Sum_count)
return
Fit Value = RMS_sum

FIGURE 42

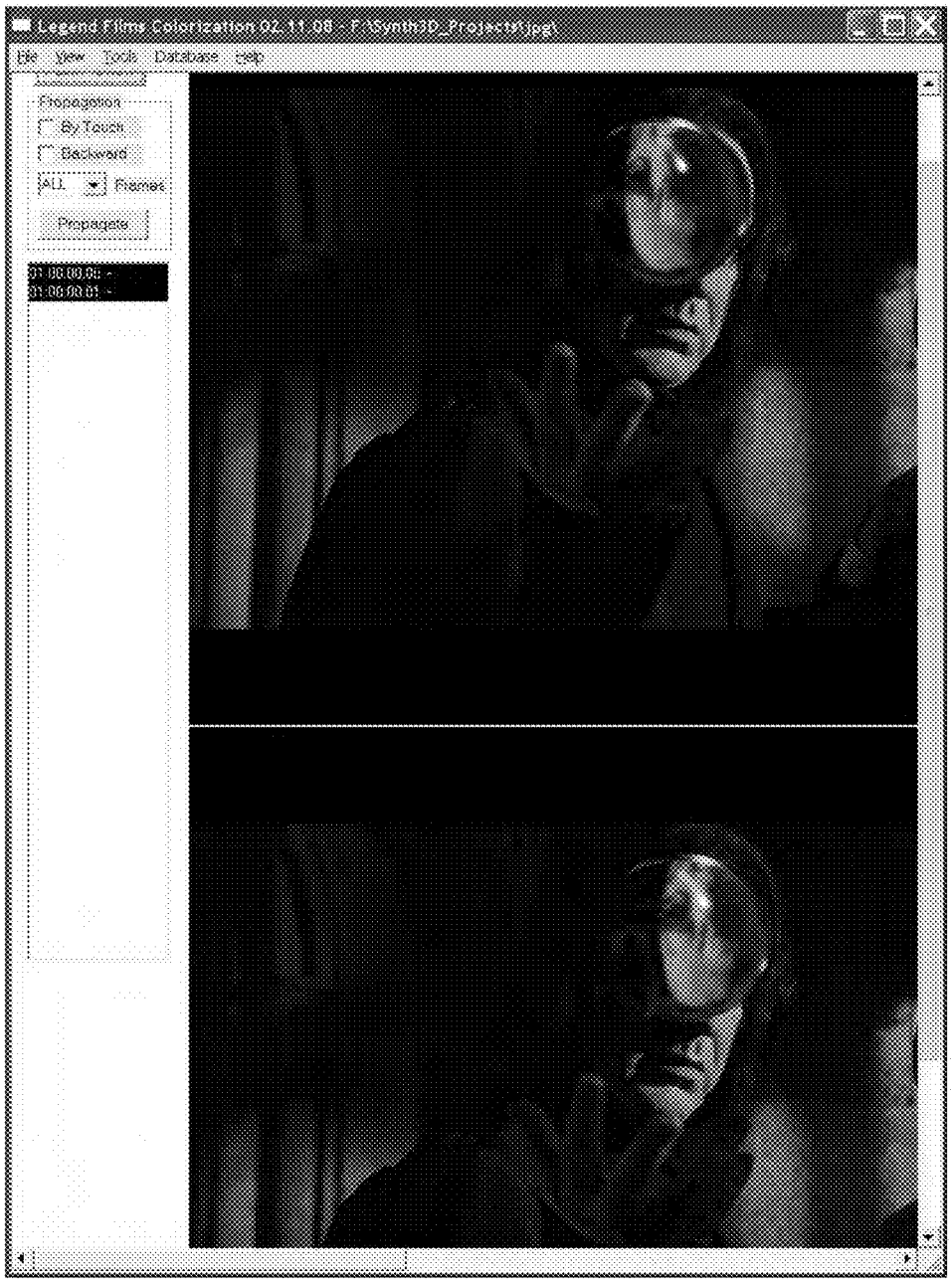

Legend Films Colorization 02.11.08 - F:\Synth3D_Projects\jpg\
File View Tools Database Help
Propagation
By Track
Backward
ALL
Frames
Propagate

FIGURE 43

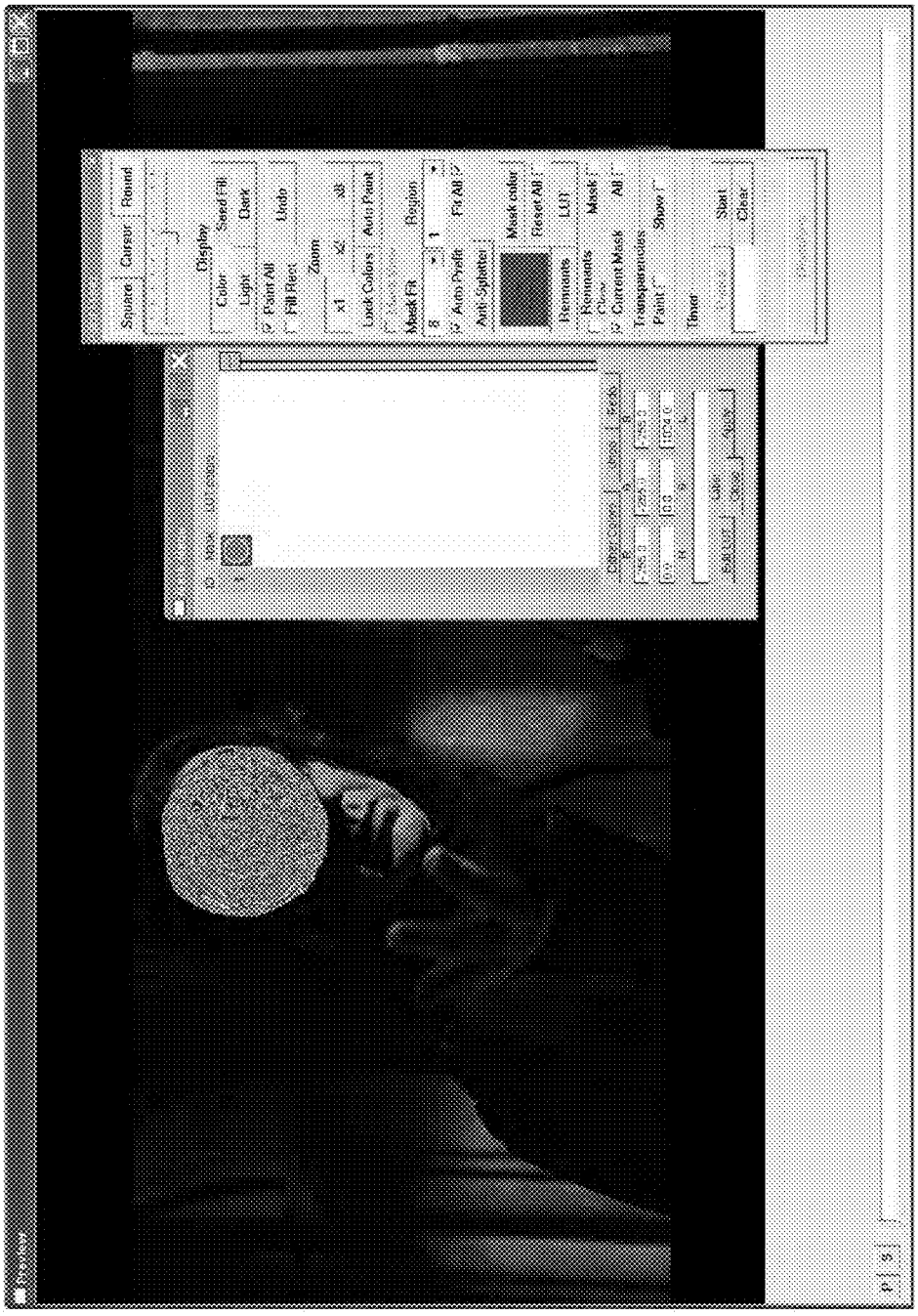

Square
Cursor
Round
Display
Color
Light
Point All
Fill Rect
Seed Fill
Dark
Undo
Zoom
x1
x2
x8
Lock Colors
Auto Paint
Mask Fit
Region
Auto Prefit
Fit All
Anti-Splatter
Mask color
Reset All
Remnants
LUT
Remnants
Mask
Clear
Current Mask
All
Transparencies
Paint
Show
Timer
Start
Clear

FIGURE 44

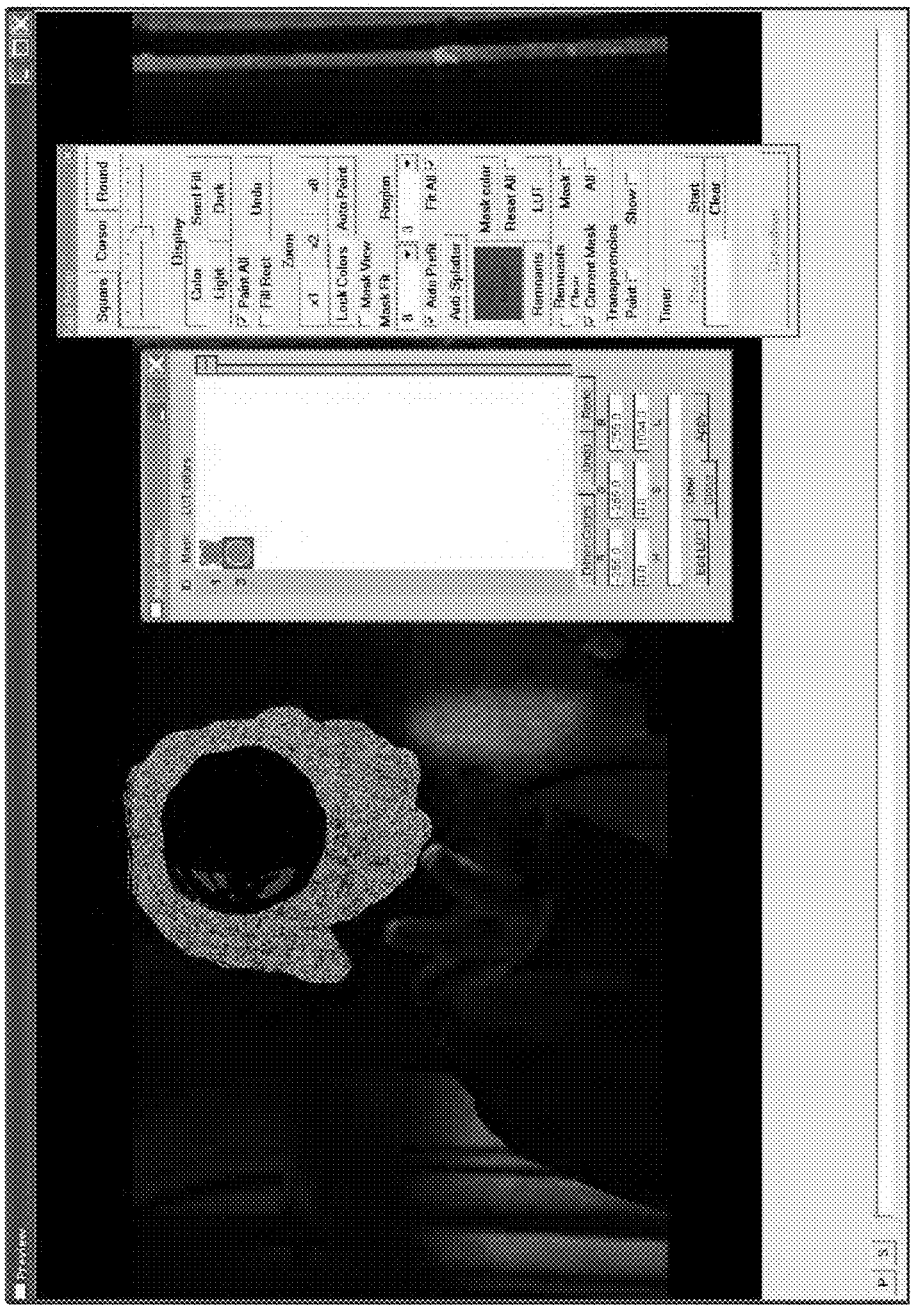

Square
Corsor
Round
Display
Color
Seed Fill
Light
Dark
Paint All
Undo
Fill Rect
Zoom
x1
x2
x8
Look Colors
Auto Paint
Mask View
Mask Fit
Region
Auto Prefit
Fit All
Auto Splatter
Mask color
Reset All
Remnants
LUT
Remnants
Mask
Clear
Current Mask
All
Transparencies
Paint
Show
Timer
Start
Clear

FIGURE 45

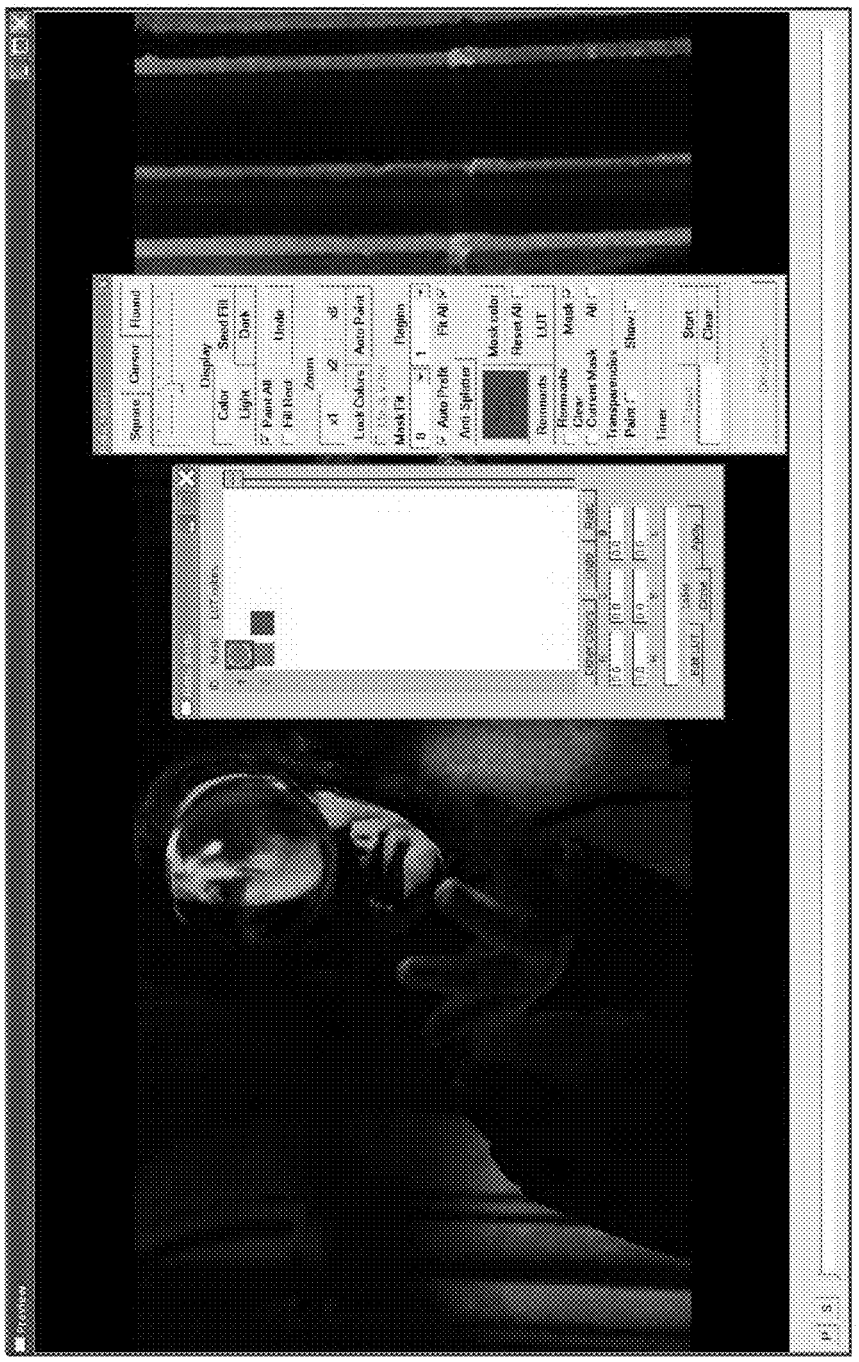

Square
Cursor
Hand
Display
Color
Light
Paint All
Fill Hole
Seed Fill
Dark
Undo
Zoom
x1
Lock Colors
Auto Paint
Mask Fit
Plugins
Auto Profit
Fit All
Anti Splatter
Mask color
Reset All
LUT
Remarks
Clear
Current Mask
Mask
All
Transparency
Paint
Show
Timer
Start
Clear

FIGURE 46

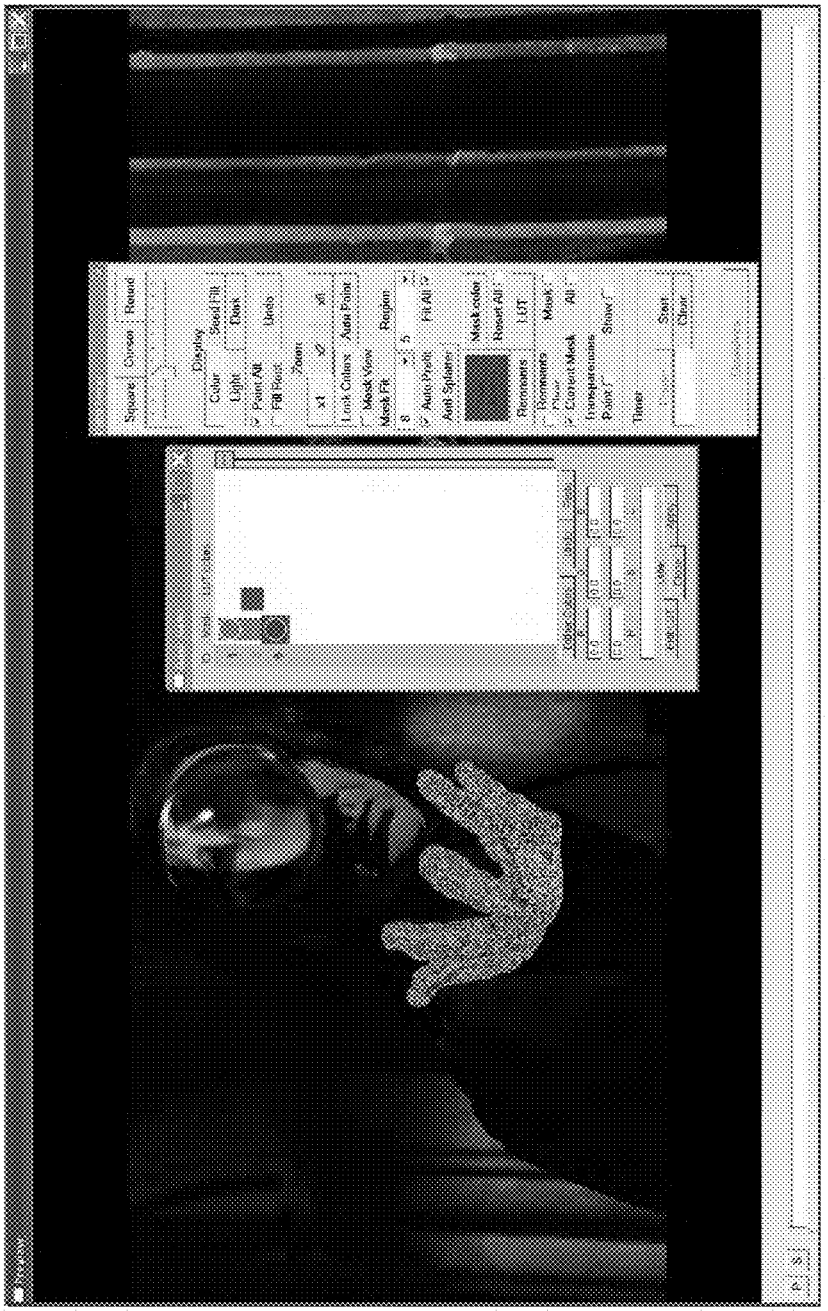

Square
Cursor
Round
Display
Color
Light
Seed Fill
Dark
Point All
Fill Rest
Undo
Zoom
x1
x2
x6
Lock Colors
Auto Point
Mask View
Mask Fit
Region
8
5
Auto Profit
Fit All
Anti Splatter
Mask color
Reset All
Remnants
LUT
Remnants
Mask
Clear
Current Mask
All
Transparencies
Point
Show
Timer
Start
Clear

FIGURE 47

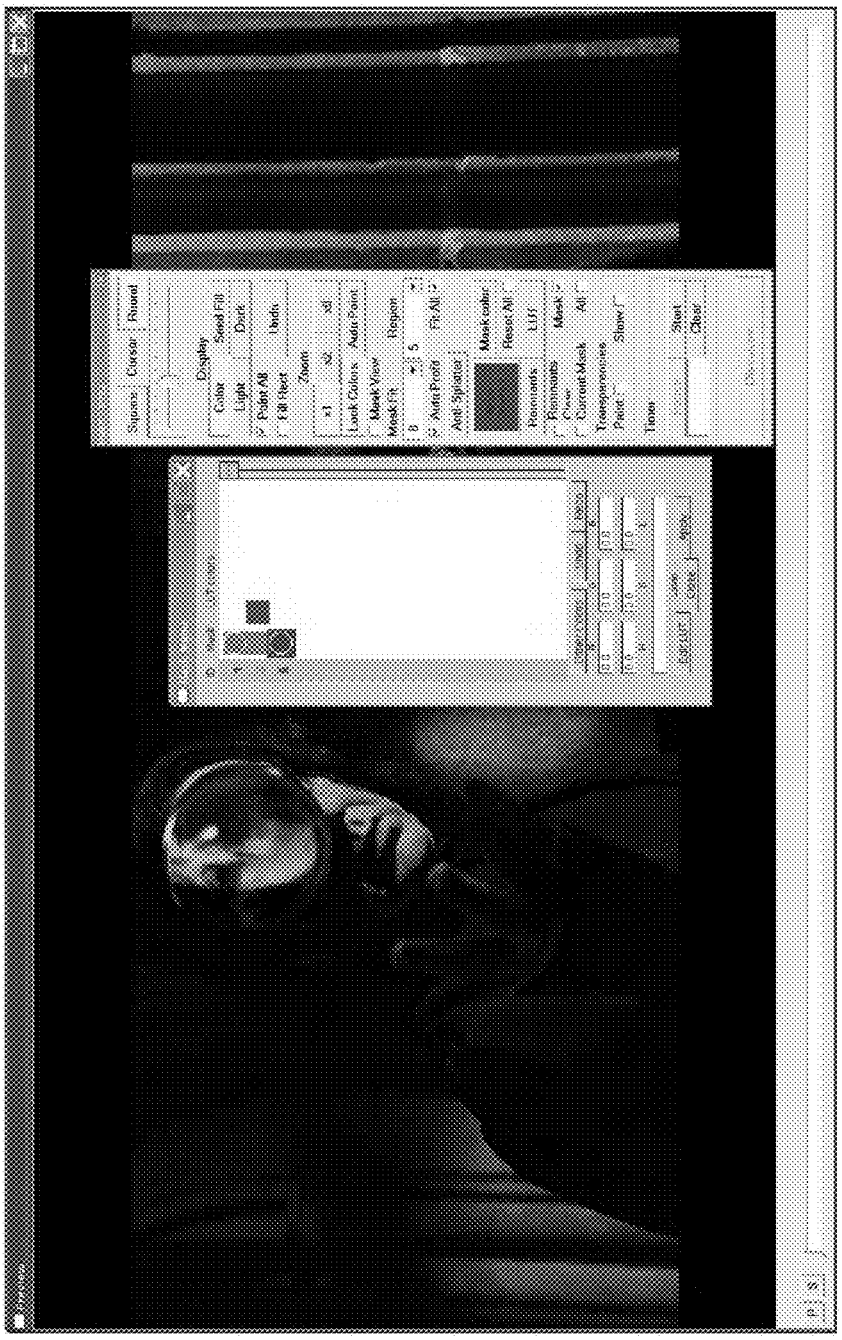

Square
Circle
Round
Display
Color
Light
Point All
Fill Hint
Seed Fill
Dark
Undo
Zoom
x1
x2
x8
Lock Colors
Auto Point
Mask View
Mask Fill
Region
5
Auto Proof
Anti-Splatter
Fit All
Mask color
Reset All
LUT
Commands
Paste
Current Mask
Mask
40
Transparencies
Paint
Show
Time
Start
Clear

FIGURE 48

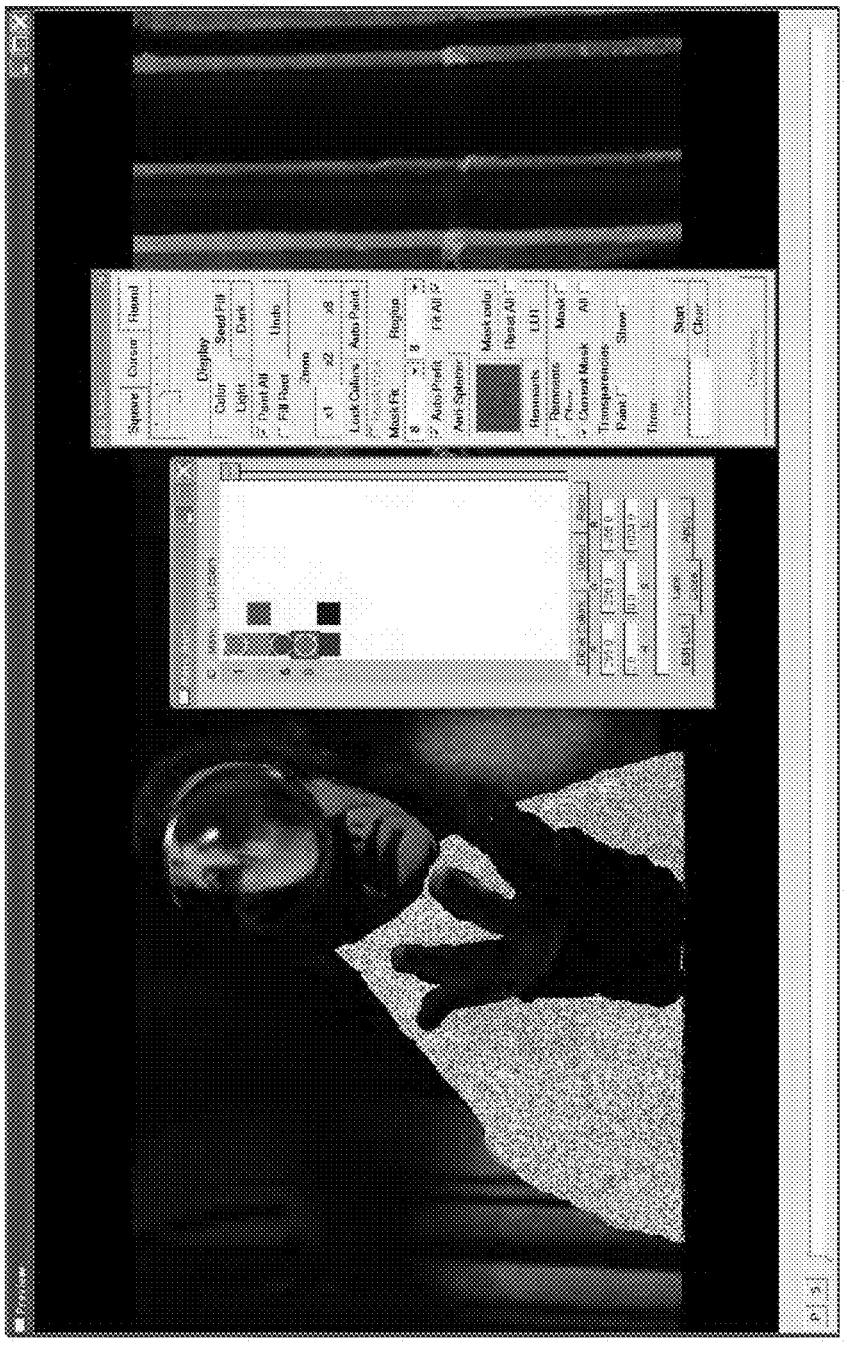

Square
Cursor
Round
Display
Color
Seed Fill
Light
Dark
Paint All
Fill Root
Undo
Zoom
x1
x2
x8
Look Colors
Auto Paint
Mask Fit
Region
Auto Prefit
Fit All
Auto Splatter
Mask color
Reset All
Remnants
LUT
Remnants
Mask
Clear
Current Mask
All
Transparencies
Point
Show
Timer
Start
Clear

FIGURE 49

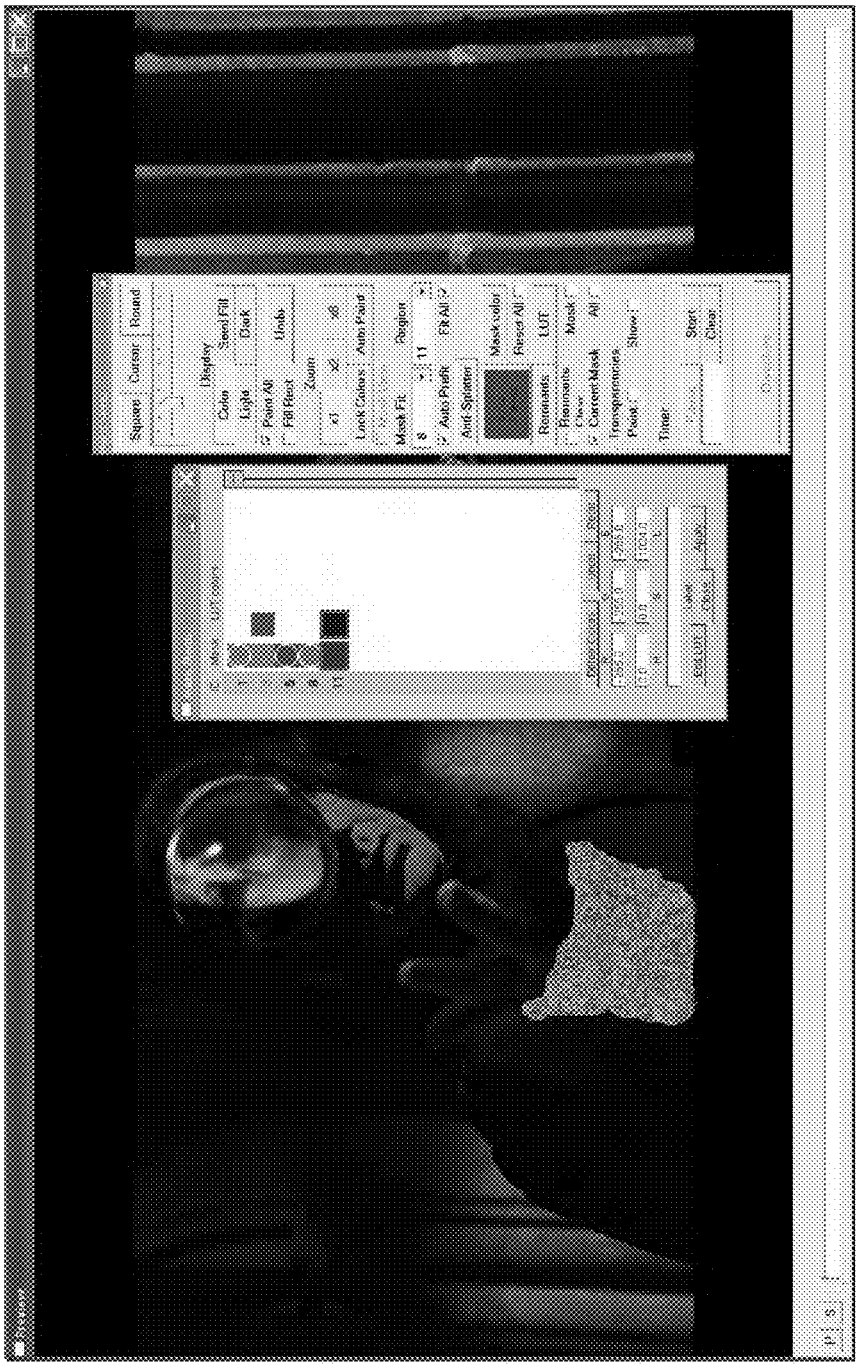

Square
Cursor
Round
Display
Color
Light
Point All
Fill Pixel
Seed Fill
Dark
Undo
Zoom
x1
x2
x8
Lock Colors
Auto Paint
Mask Fit
Region
11
Auto Push
Fit All
Anti Splatter
Mask color
Reset All
LUT
Remnants
Illuminate
Mask
Clear
Current Mask
All
Transparencies
Pixel
Show
Timer
Start
Clear

FIGURE 50

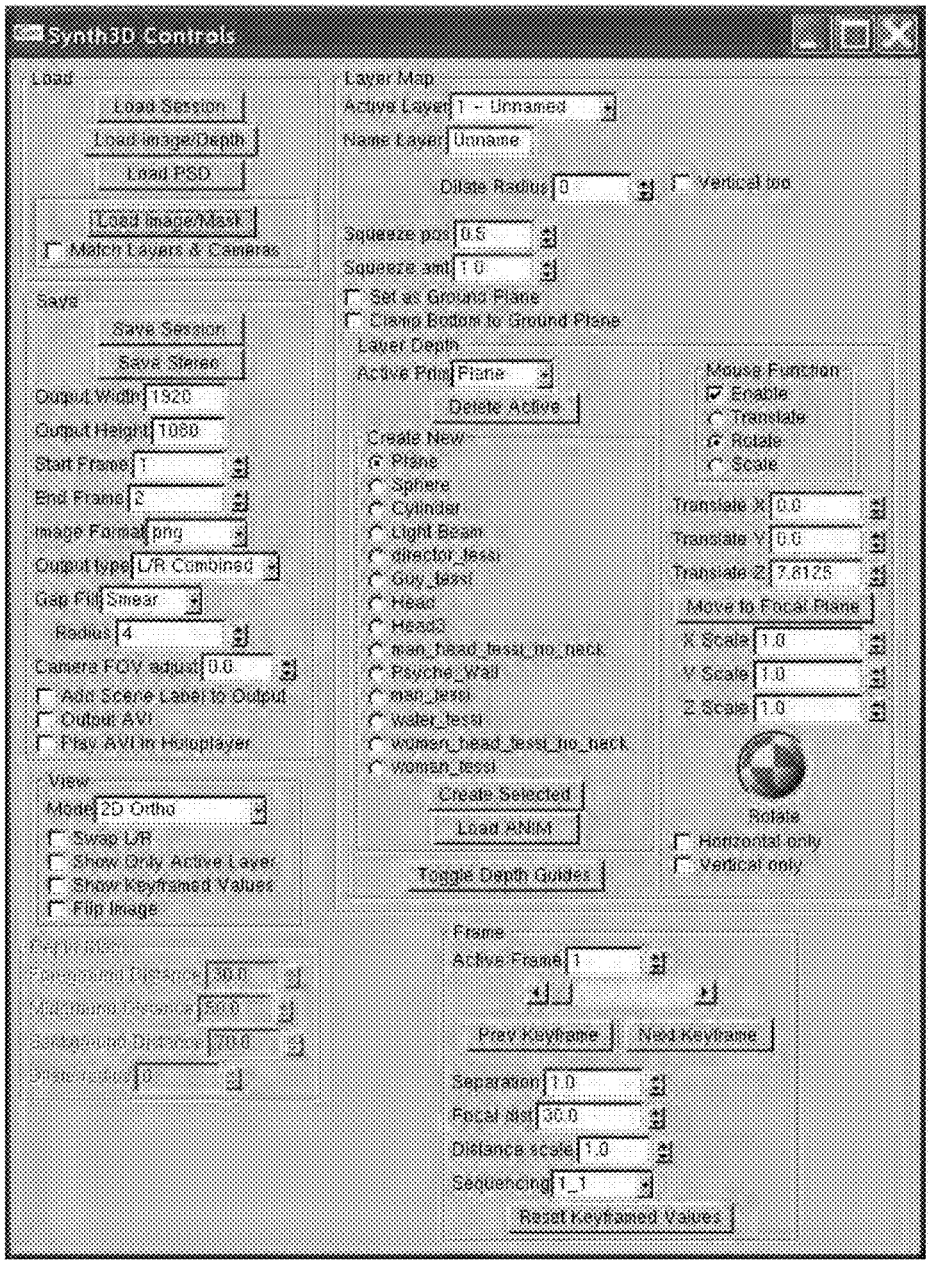

Synth3D Controls
Load
Load Session
Load Image/Depth
Load PSD
Load Image/Mask
Match Layers & Cameras
Save
Save Session
Save Stereo
Output Width 1920
Output Height 1080
Start Frame 1
End Frame 2
Image Format png
Output type L/R Combined
Gap Fill Smear
Radius 4
Camera FOV adjust 0.0
Add Scene Label to Output
Output AVI
Play AVI in Hiroplayer
View
Mode 2D Ortho
Swap L/R
Show Only Active Layer
Show Keyframed Values
Flip Image
Layer Map
Active Layer 1 -- Unnamed
Name Layer Unname
Dilate Radius 0
Vertical Iris
Squeeze pos 0.5
Squeeze amt 1.0
Set as Ground Plane
Clamp Bottom to Ground Plane
Layer Depth
Active Proj Plane
Delete Active
Create New
Plane
Sphere
Cylinder
Light Beam
director_test
Grey_test
Head
Head2
man_head_test_no_neck
Psyche_Wall
man_test
water_test
women_head_test_no_neck
Womsh_test
Create Selected
Load ANIM
Toggle Depth Guides
Mouse Function
Enable
Translate
Rotate
Scale
Translate X 0.0
Translate Y 0.0
Translate Z 7.6125
Move to Focal Plane
X Scale 1.0
Y Scale 1.0
Z Scale 1.0
Rotate
Horizontal only
Vertical only
Frame
Active Frame 1
Prev Keyframe
Next Keyframe
Separation 1.0
Focal dist 30.0
Distance scale 1.0
Sequencing 1_1
Reset Keyframed Values

FIGURE 51

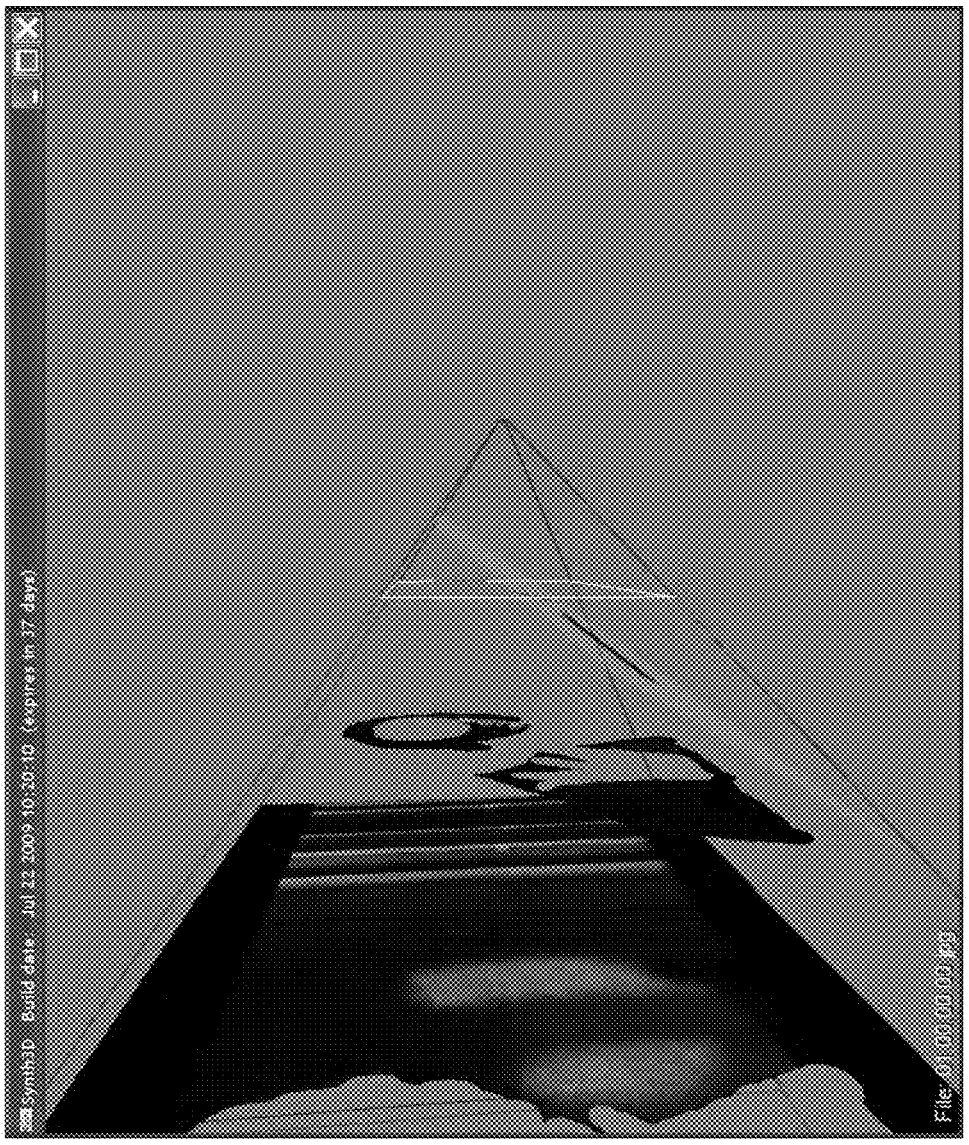


FIGURE 52

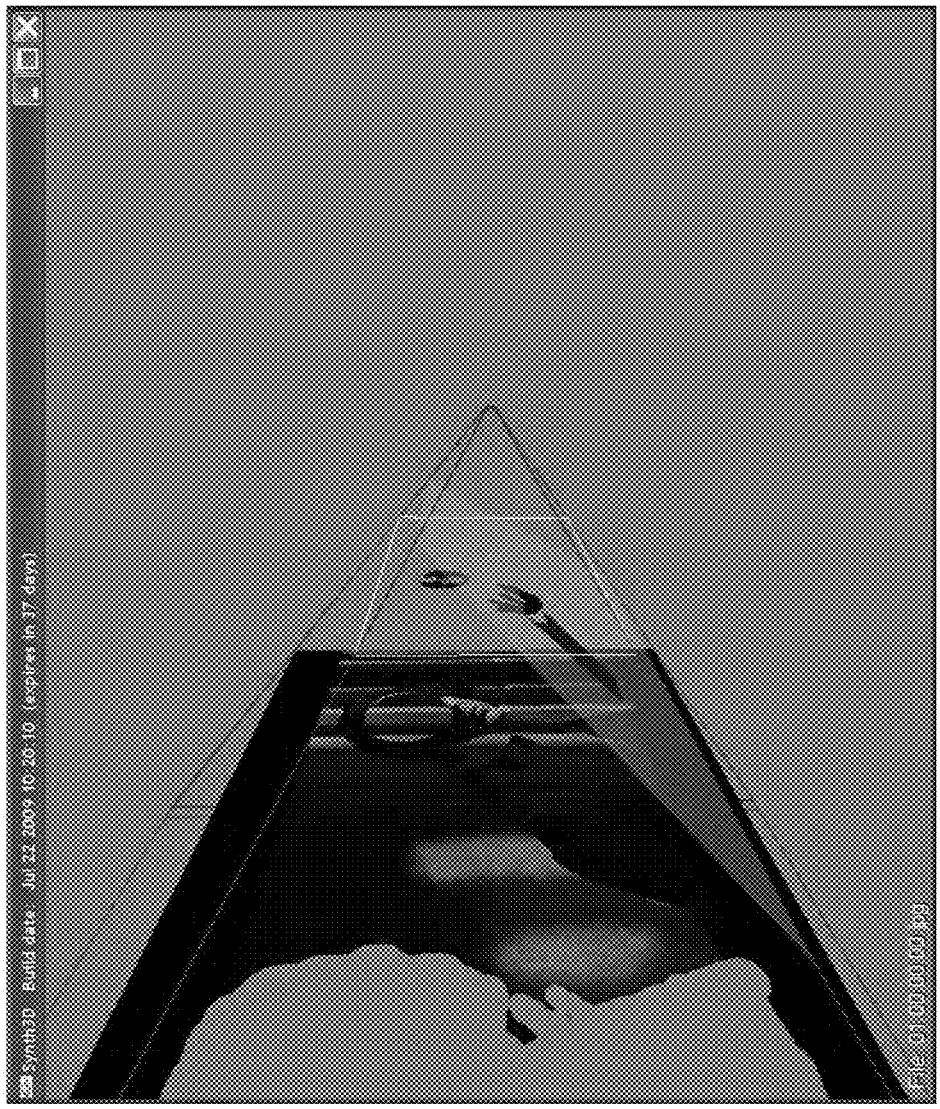


FIGURE 53

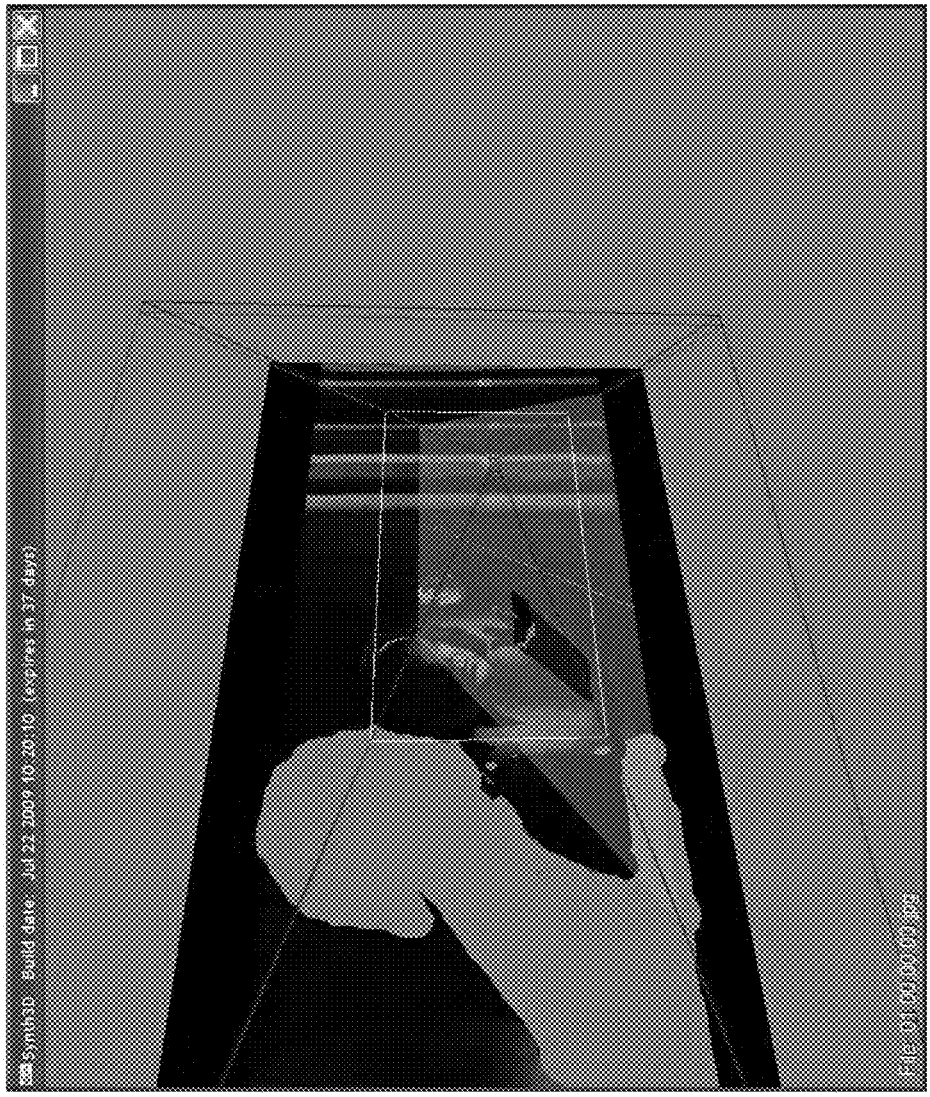


FIGURE 54

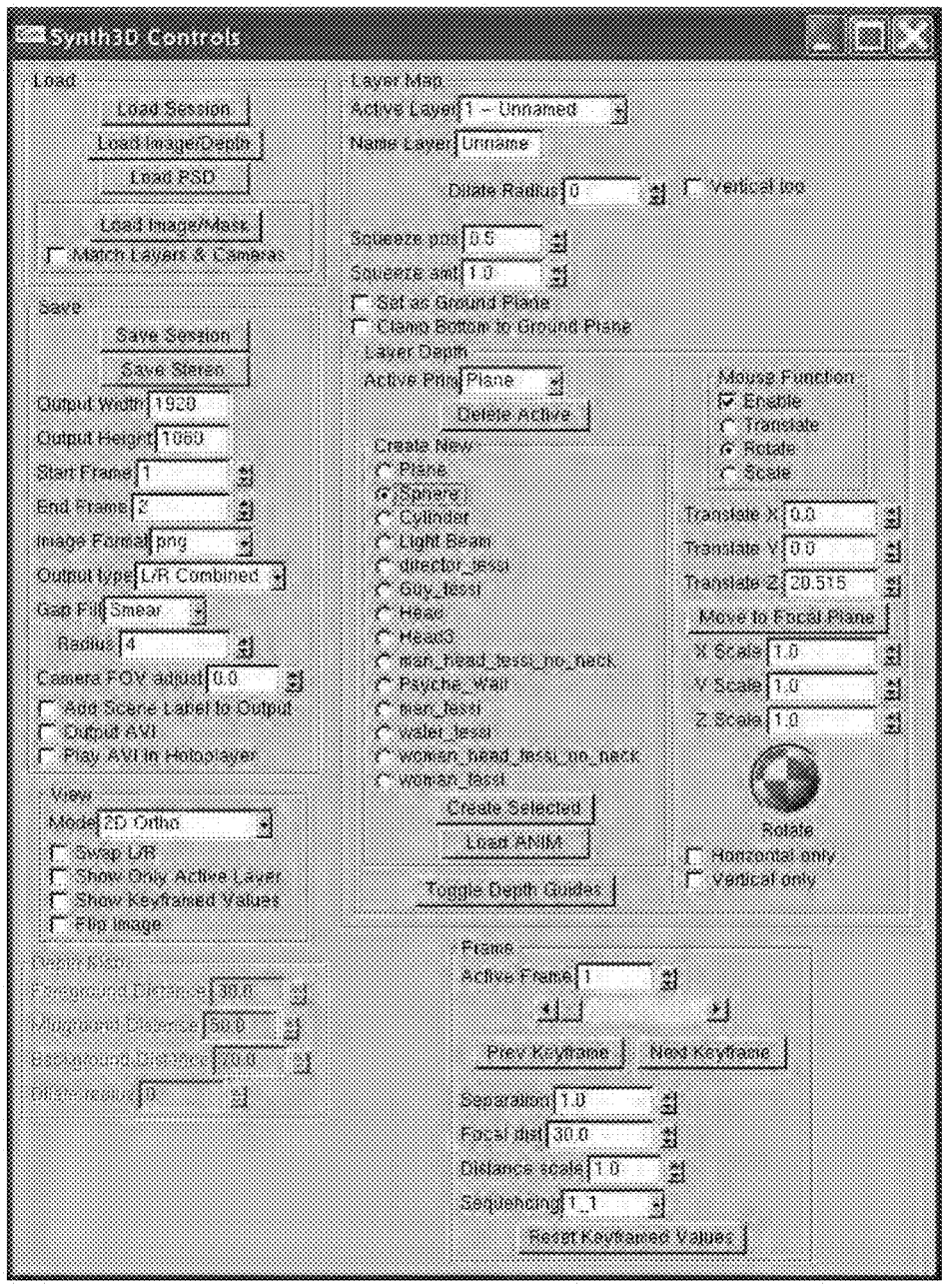

Synth3D Controls
Load
Load Session
Load Image/Depth
Load PSD
Load Image/Mask
Match Layers & Cameras
Save
Save Session
Save Stereo
Output Width 1920
Output Height 1080
Start Frame 1
End Frame 2
Image Format png
Output type L/R Combined
Gap Fill Smear
Radius 4
Camera FOV adjust 0.0
Add Scene Label to Output
Output AVI
Play AVI in Hologram
View
Mode 2D Ortho
Swap L/R
Show Only Active Layer
Show Keyframed Values
Flip Image
Layer Map
Active Layer 1 -- Unnamed
Name Layer Unname
Dilate Radius 0
Vertical loc
Squeeze pos 0.5
Squeeze amt 1.0
Set as Ground Plane
Clamp Bottom to Ground Plane
Layer Depth
Active Prim Plane
Delete Active
Create New
Plane
Sphere
Cylinder
Light Beam
directly_test
Guy_test
Head
Head3
man_head_test_no_neck
Psyche_Wall
man_test
wallet_test
woman_head_test_no_neck
woman_test
Create Selected
Load ANIM
Toggle Depth Guides
Mouse Function
Enable
Translate
Rotate
Scale
Translate X 0.0
Translate Y 0.0
Translate Z 20.515
Move to Focal Plane
X Scale 1.0
Y Scale 1.0
Z Scale 1.0
Rotate
Horizontal only
Vertical only
Frame
Active Frame 1
Prev Keyframe
Next Keyframe
Separation 1.0
Focal dist 30.0
Distance scale 1.0
Sequencing 1_1
Reset Keyframed Values

FIGURE 55

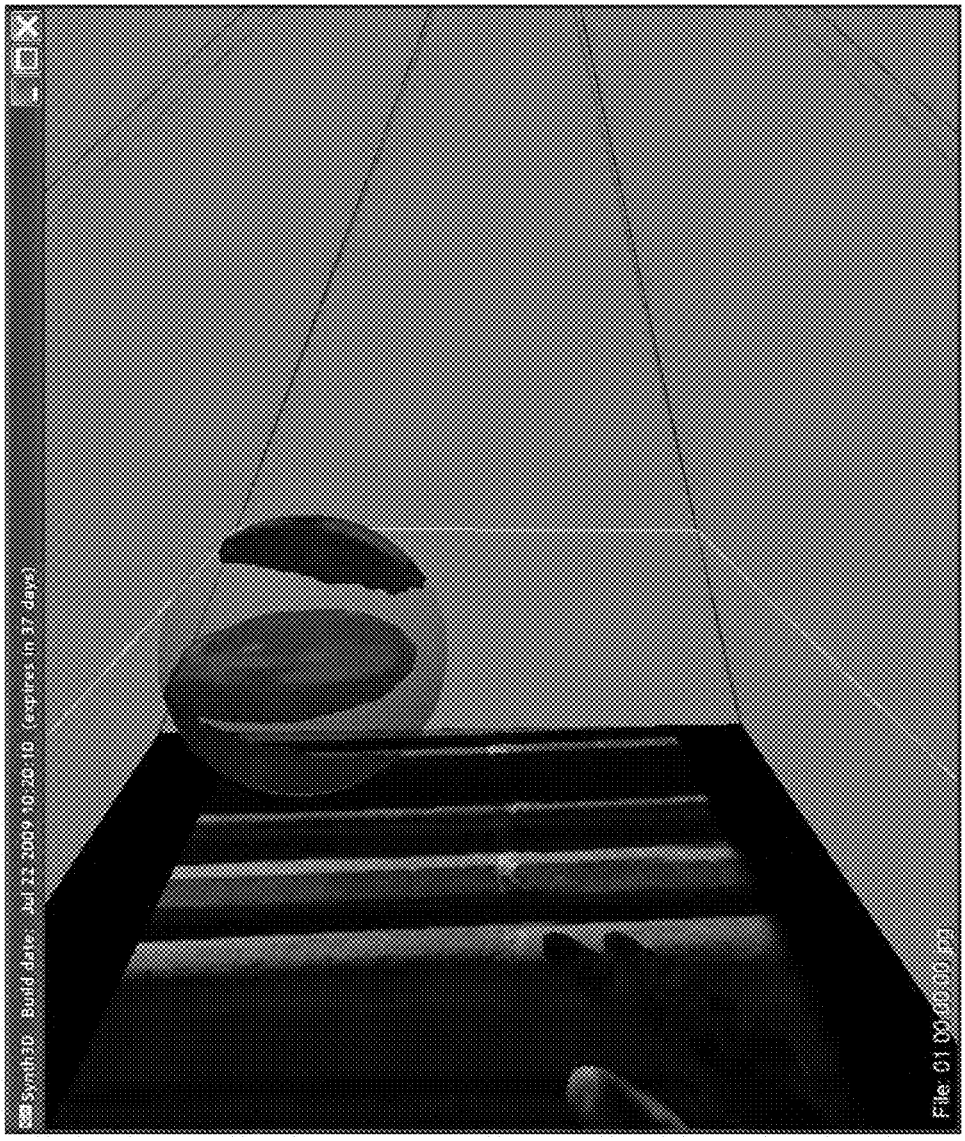


FIGURE 56

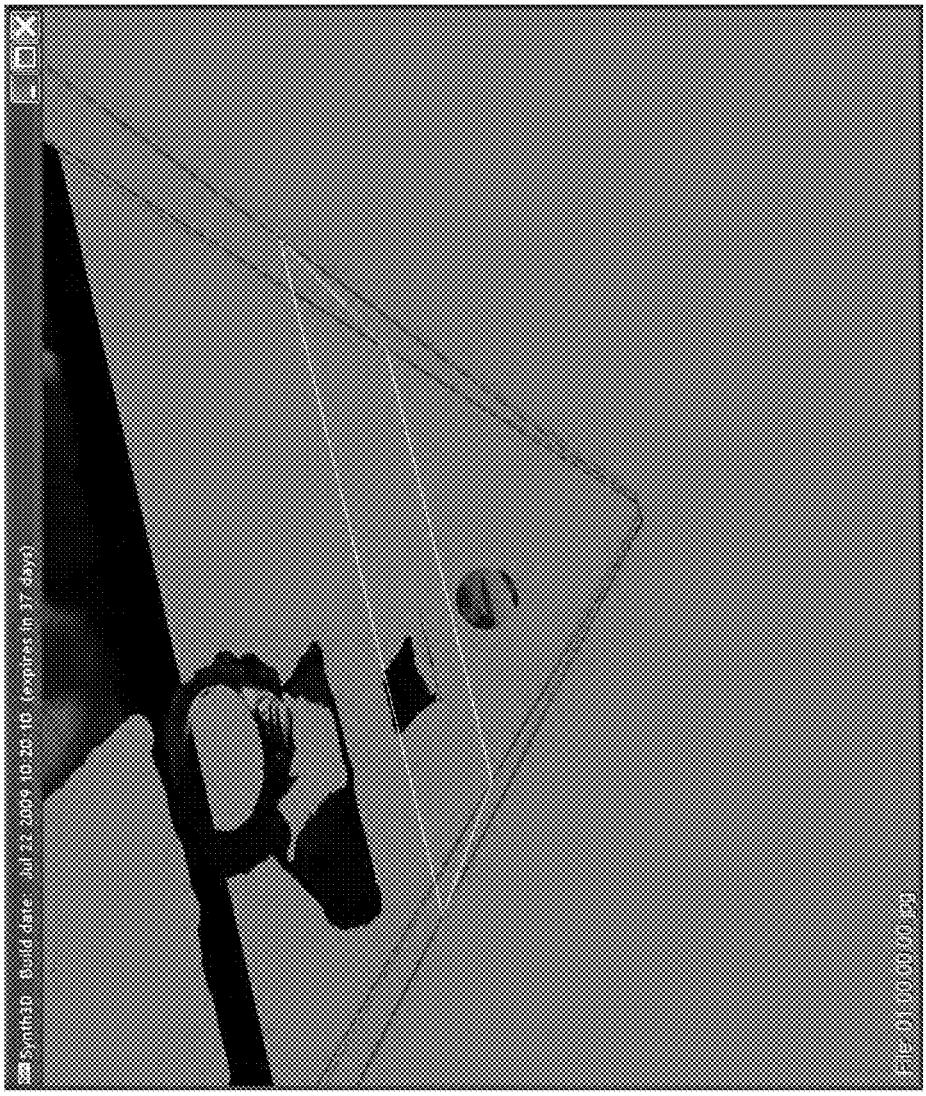


FIGURE 57

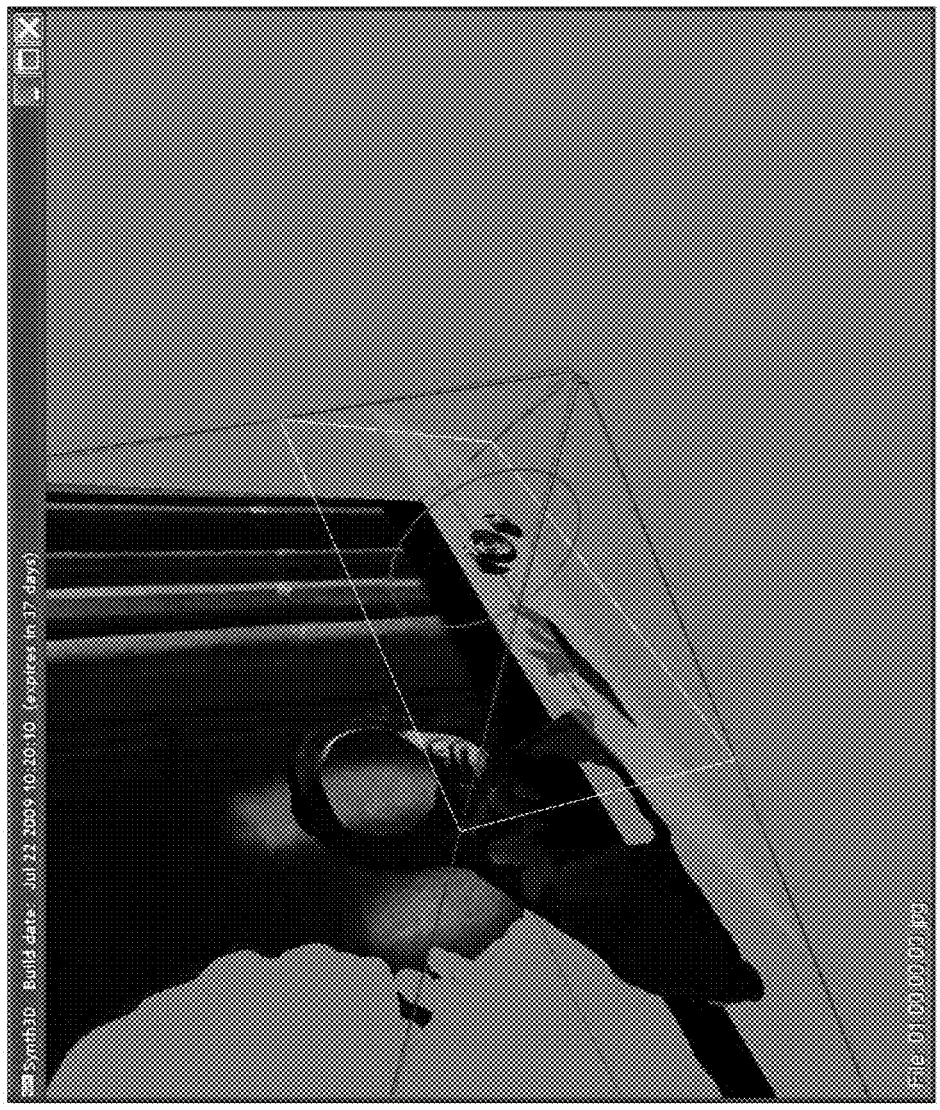


FIGURE 58

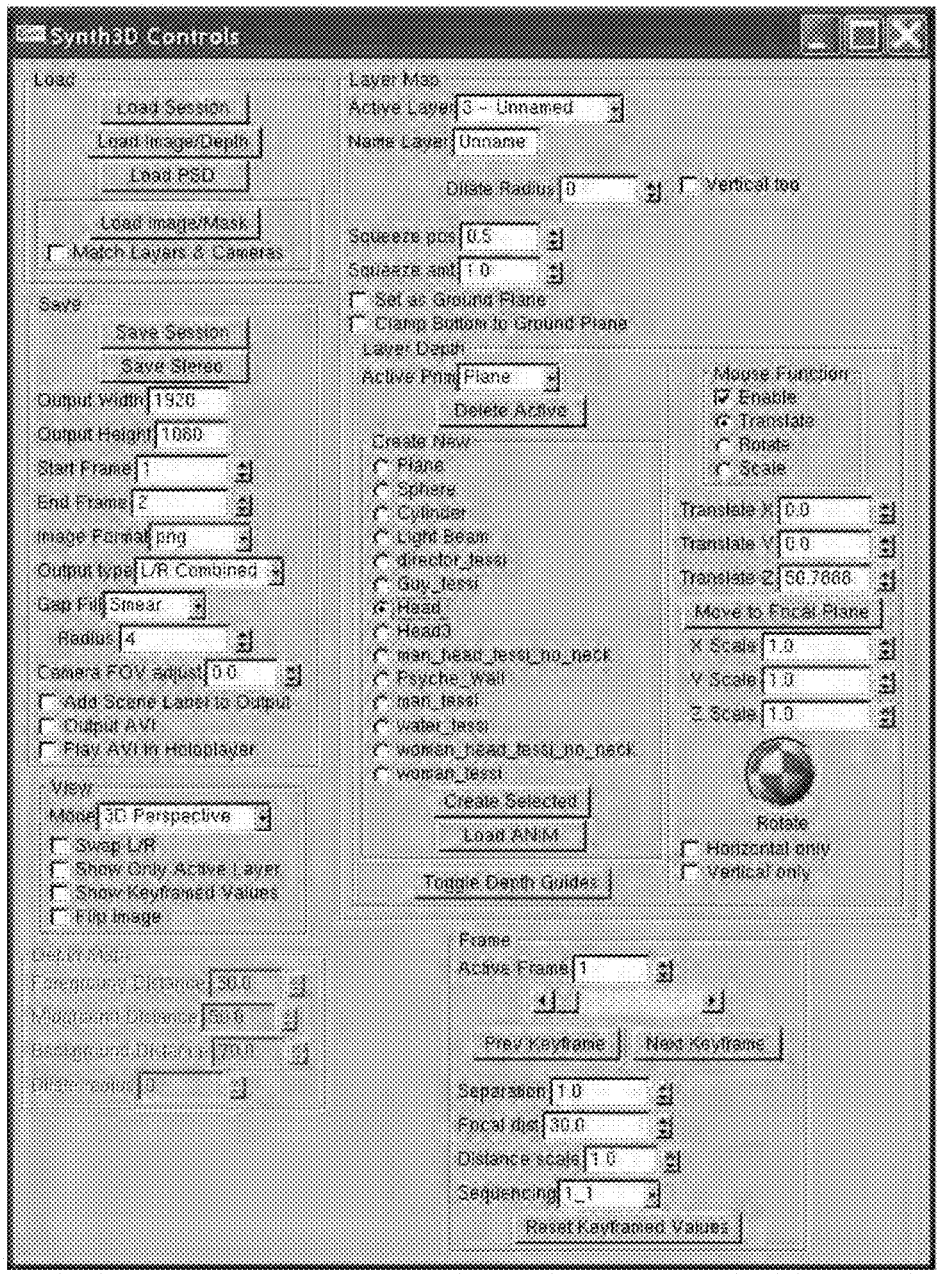

Synth3D Controls
Load
Load Session
Load Image/Depth
Load PSD
Load Image/Mask
Match Layers & Cameras
Save
Save Session
Save Stereo
Output Width 1920
Output Height 1080
Start Frame 1
End Frame 2
Image Format png
Output type L/R Combined
Gap Fill Smear
Radius 4
Camera FOV adjust 0.0
Add Scene Layer to Output
Output AVI
Play AVI in Hololayer
View
Mode 3D Perspective
Swap L/R
Show Only Active Layer
Show Keyframed Values
Fill Image
Layer Map
Active Layer 3 - Unnamed
Name Layer Unname
Dilate Radius 0
Vertical loe
Squeeze pos 0.5
Squeeze amt 1.0
Set as Ground Plane
Clamp Bottom to Ground Plane
Layer Depth
Active Prim Plane
Delete Active
Create New
Plane
Sphere
Cylinder
Light Beam
director_test
Guy_test
Head
Head3
man_head_test_no_neck
Psyche_Wall
man_test
water_test
woman_head_test_no_neck
woman_test
Create Selected
Load ANIM
Toggle Depth Guides
Mouse Function
Enable
Translate
Rotate
Scale
Translate X 0.0
Translate Y 0.0
Translate Z 50.7868
Move to Focal Plane
X Scale 1.0
Y Scale 1.0
Z Scale 1.0
Rotate
Horizontal only
Vertical only
Frame
Active Frame 1
Prev Keyframe
Next Keyframe
Separation 1.0
Focal dist 30.0
Distance scale 1.0
Sequencing 1_1
Reset Keyframed Values

FIGURE 59

FIGURE 60

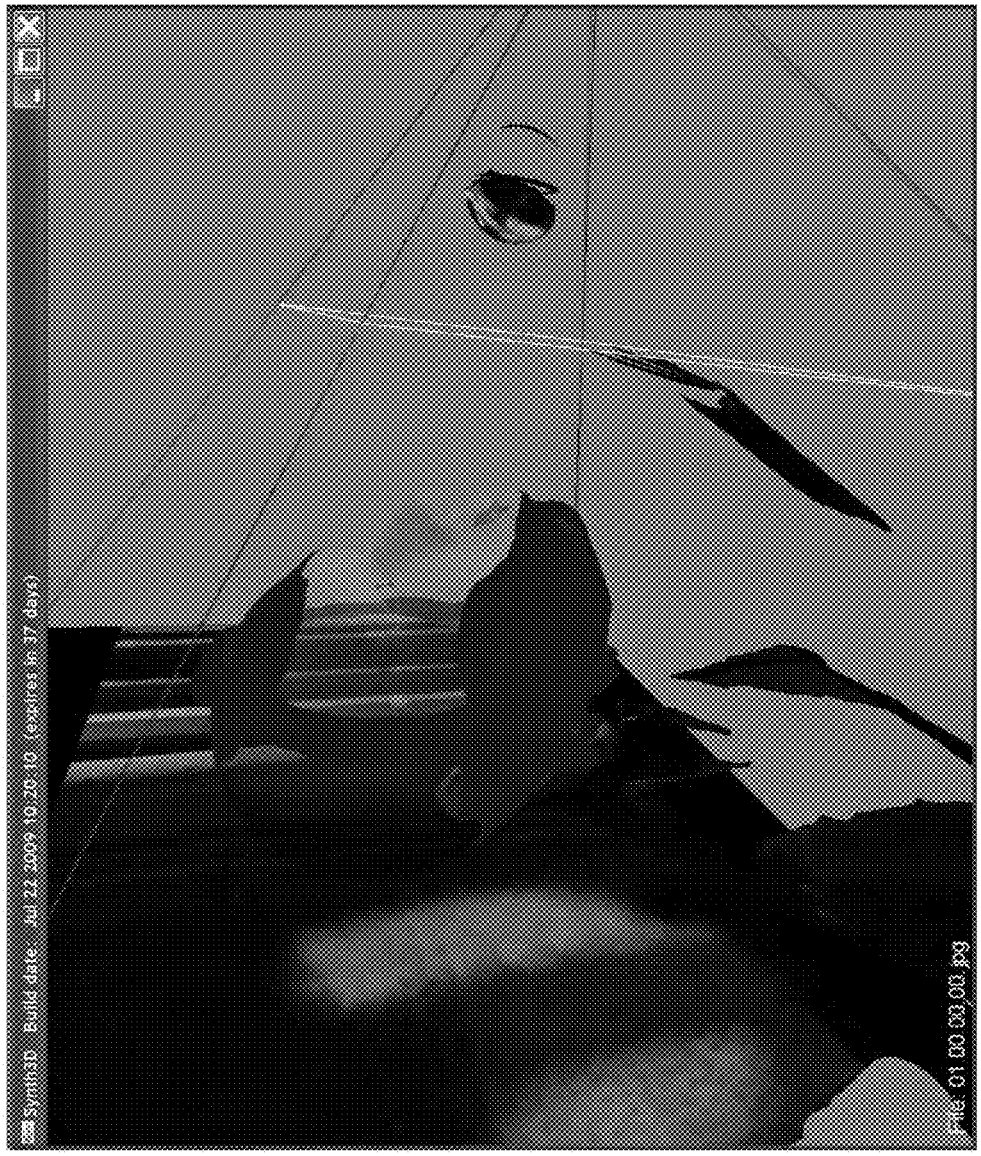

SynthEyes Build date: Jul 22 2009 10:20:10 (expires in 37 days)
File: 01 00 00_00.jpg

FIGURE 61

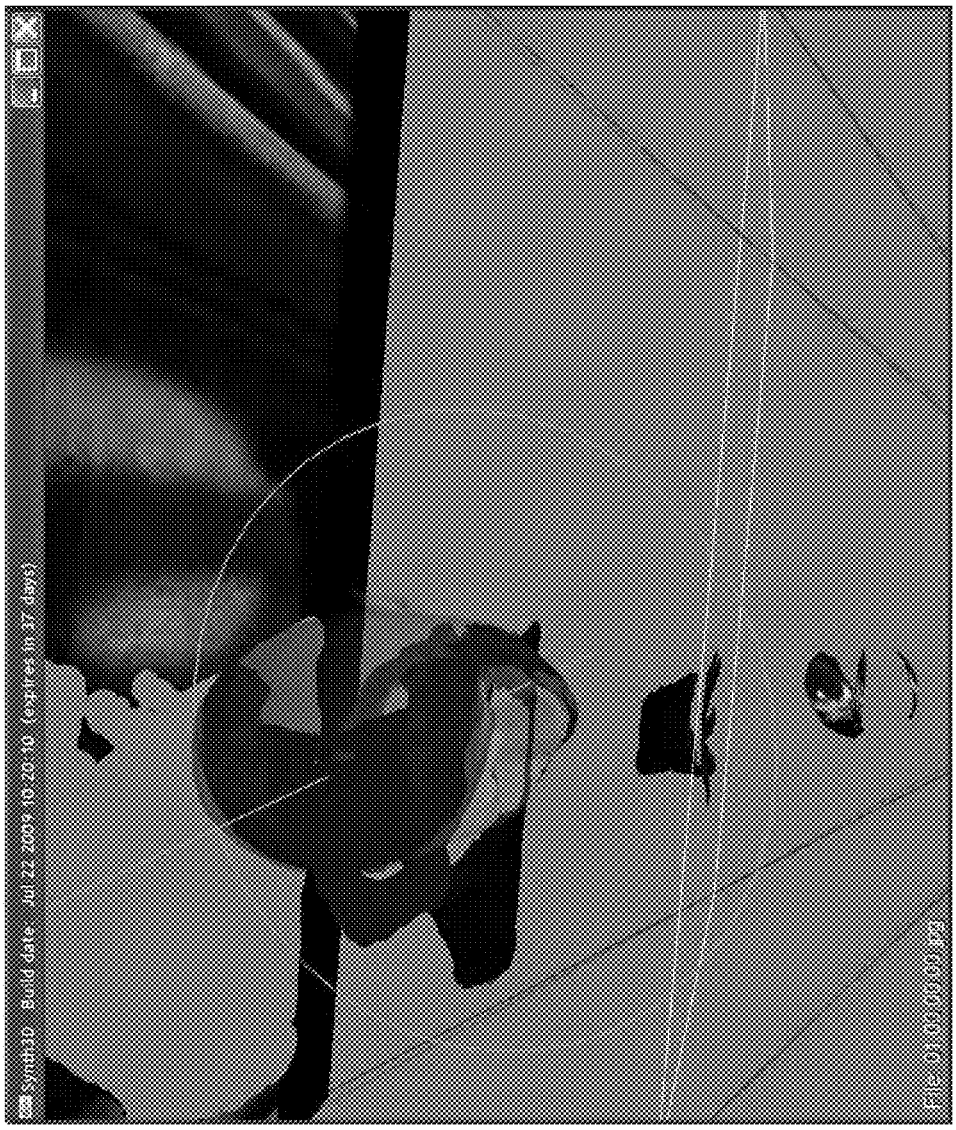


FIGURE 62

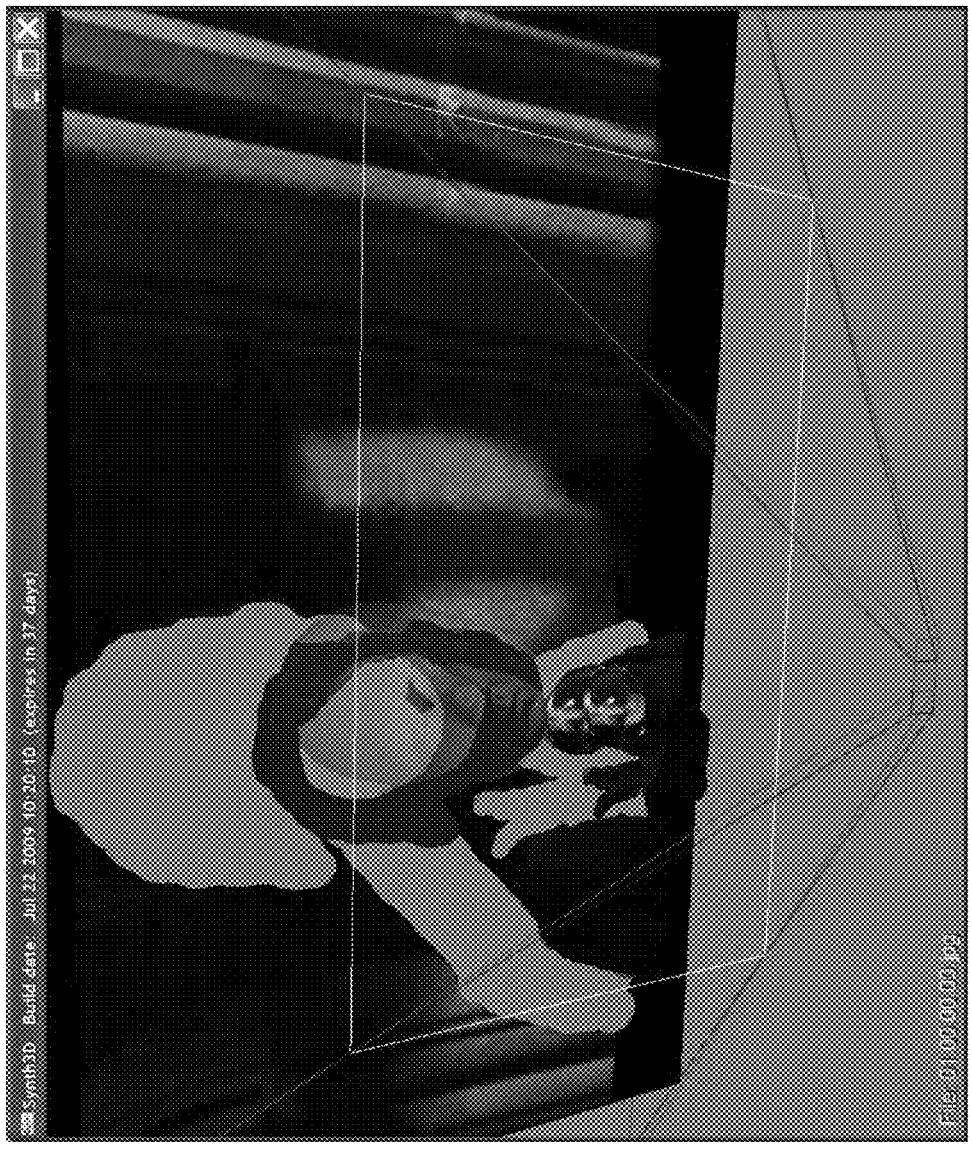

Synth3D Build date: Jul 22 2009 10:30:10 (expires in 37 days)
File: 01:00:00:00.pg

FIGURE 63

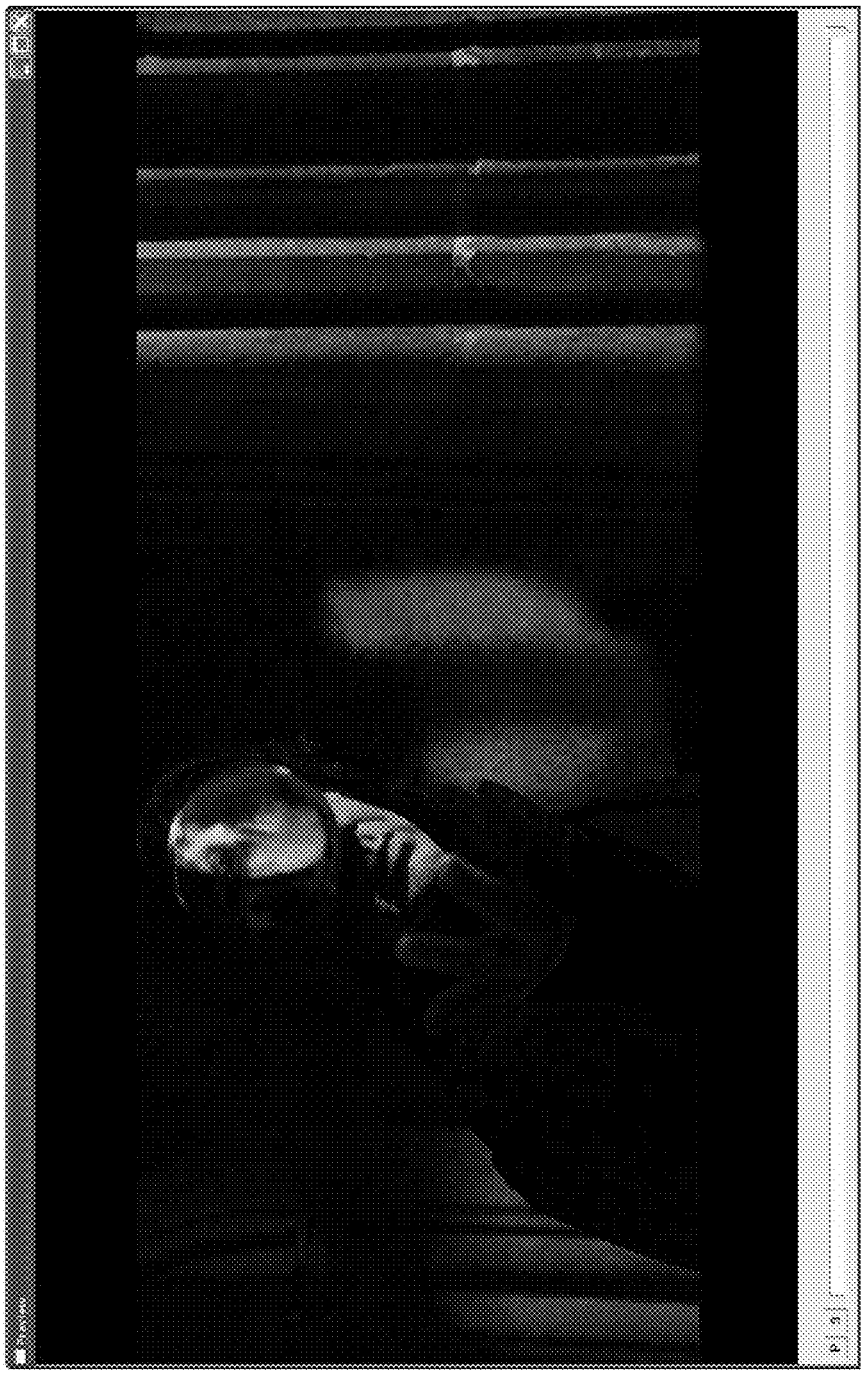


FIGURE 64

FIGURE 65

Square
Cursor
Round
Display
Color
Seed Fill
Light
Dark
Paint All
Undo
Fill Rect
Zoom
x1
x2
x8
Lock Colors
Auto Paint
Mask Fit
Region
Auto Prefit
Fit All
Anti-Splotter
Mask color
Reset All
Remnants
LUT
Remnants
Mask
Current Mask
All
Transparencies
Point
Show
Timer
Start
Clear

FIGURE 66

FIGURE 67

FIGURE 68

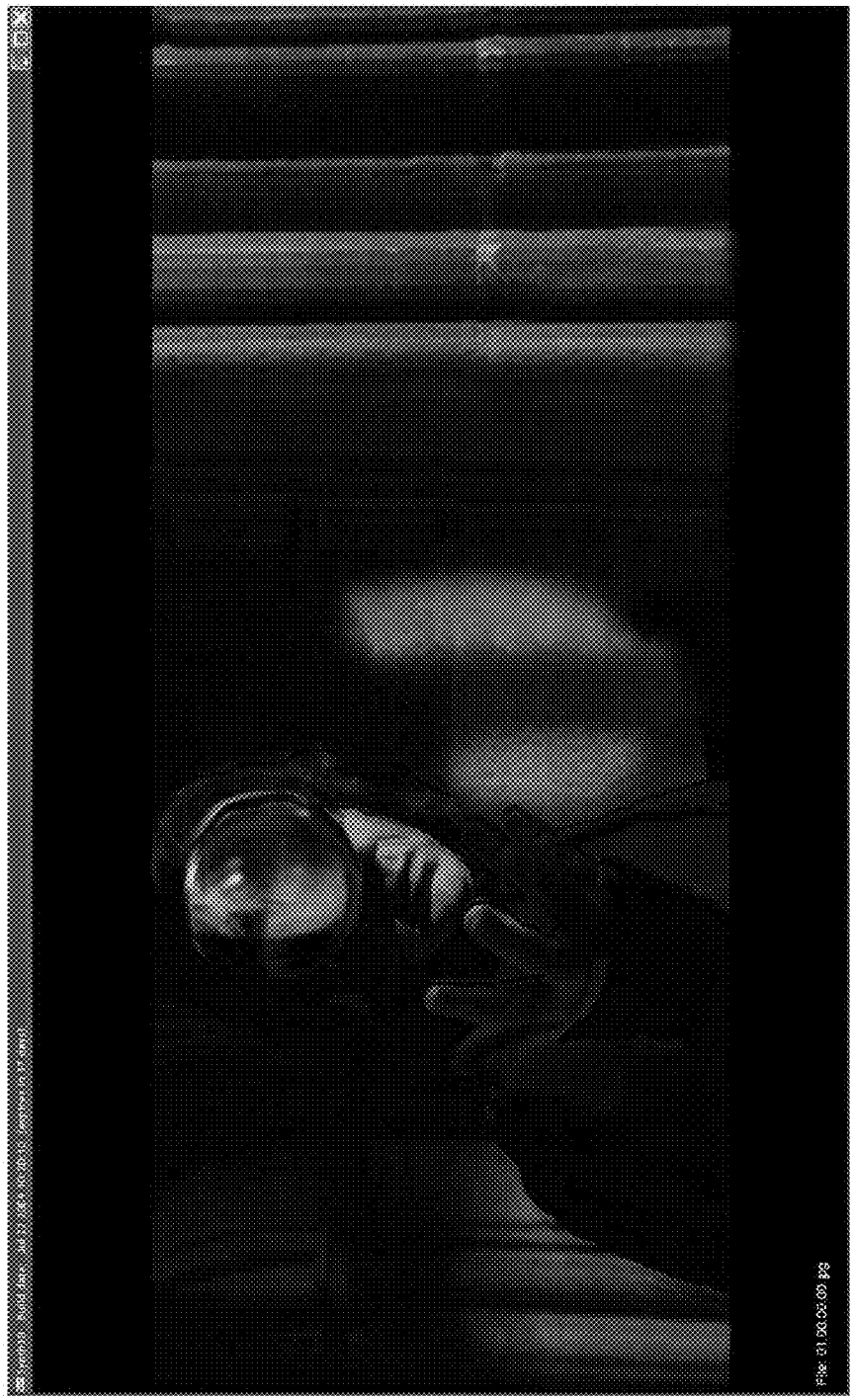


FIGURE 69

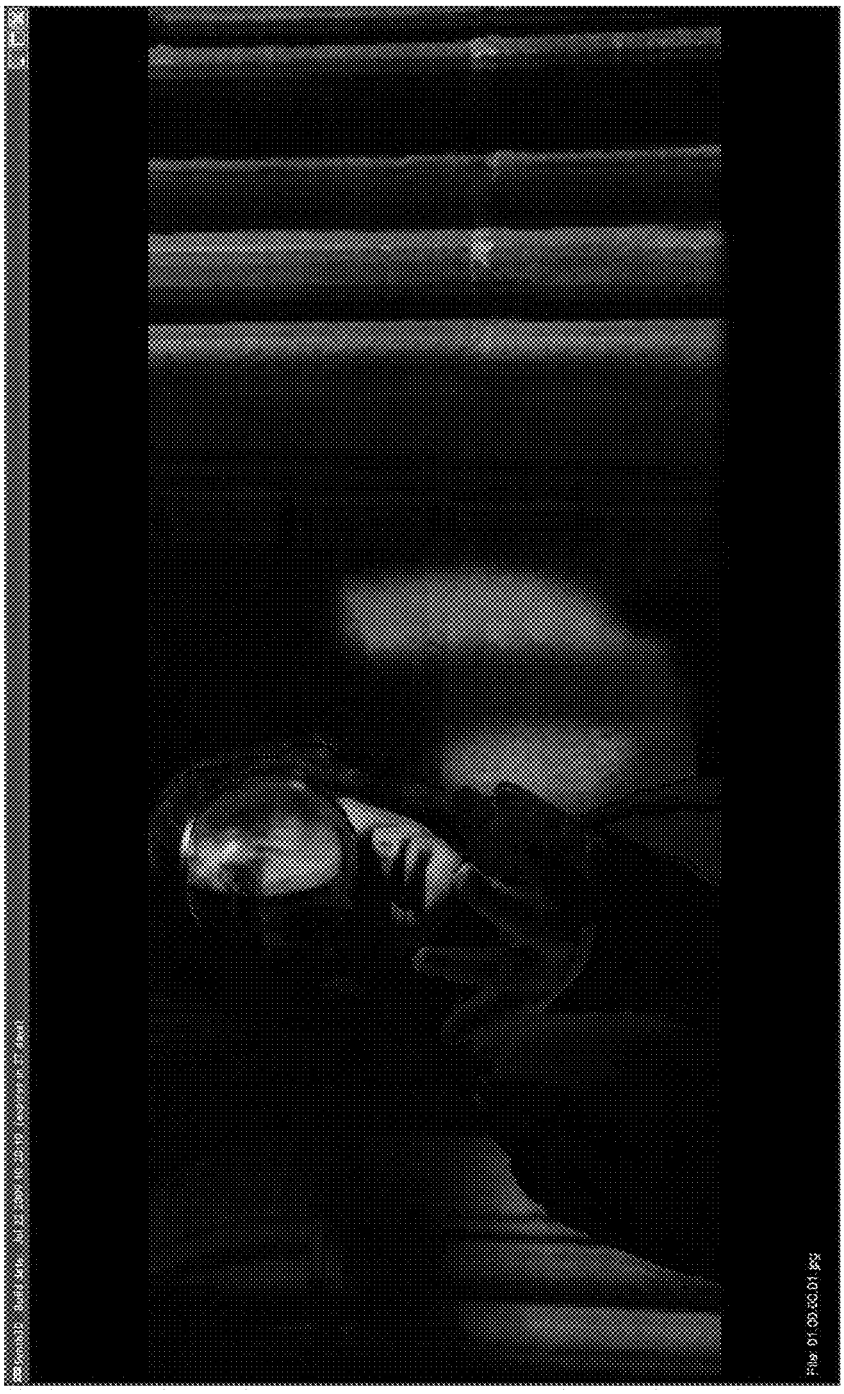


FIGURE 70

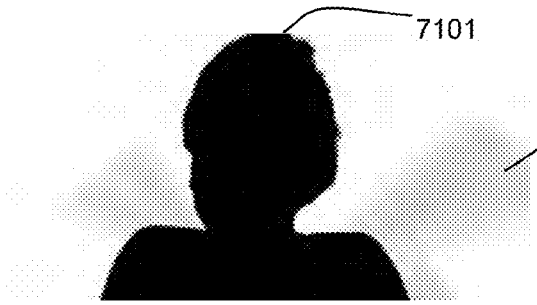


FIGURE 71

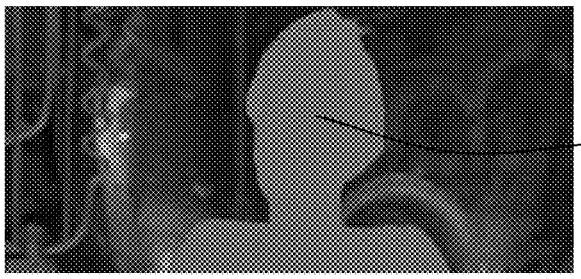


FIGURE 72

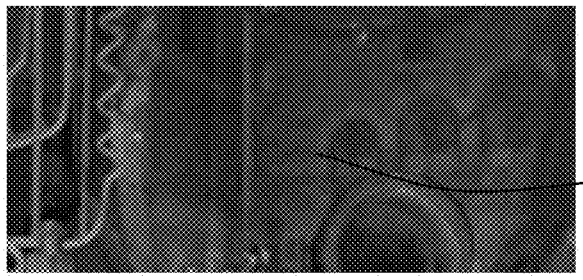


FIGURE 73

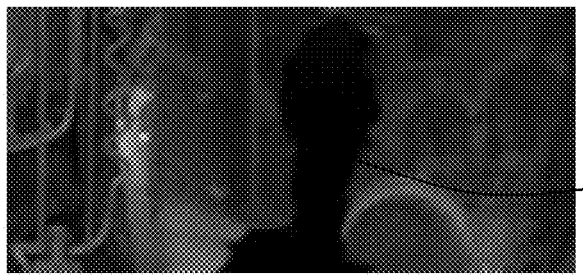


FIGURE 73A

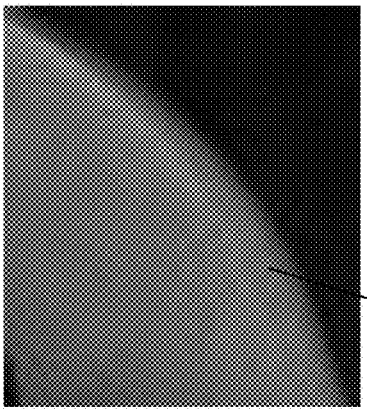


FIGURE 74

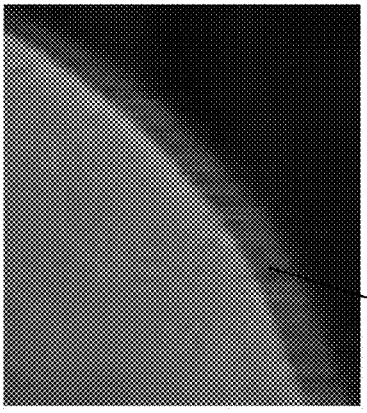


FIGURE 75

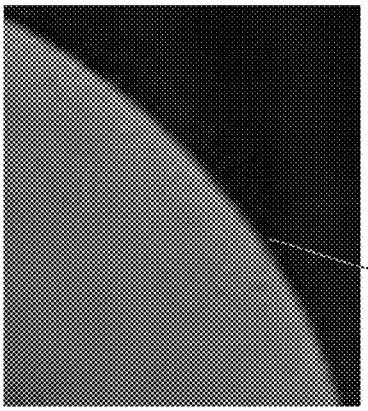


FIGURE 76

FIGURE 77

FIGURE 78

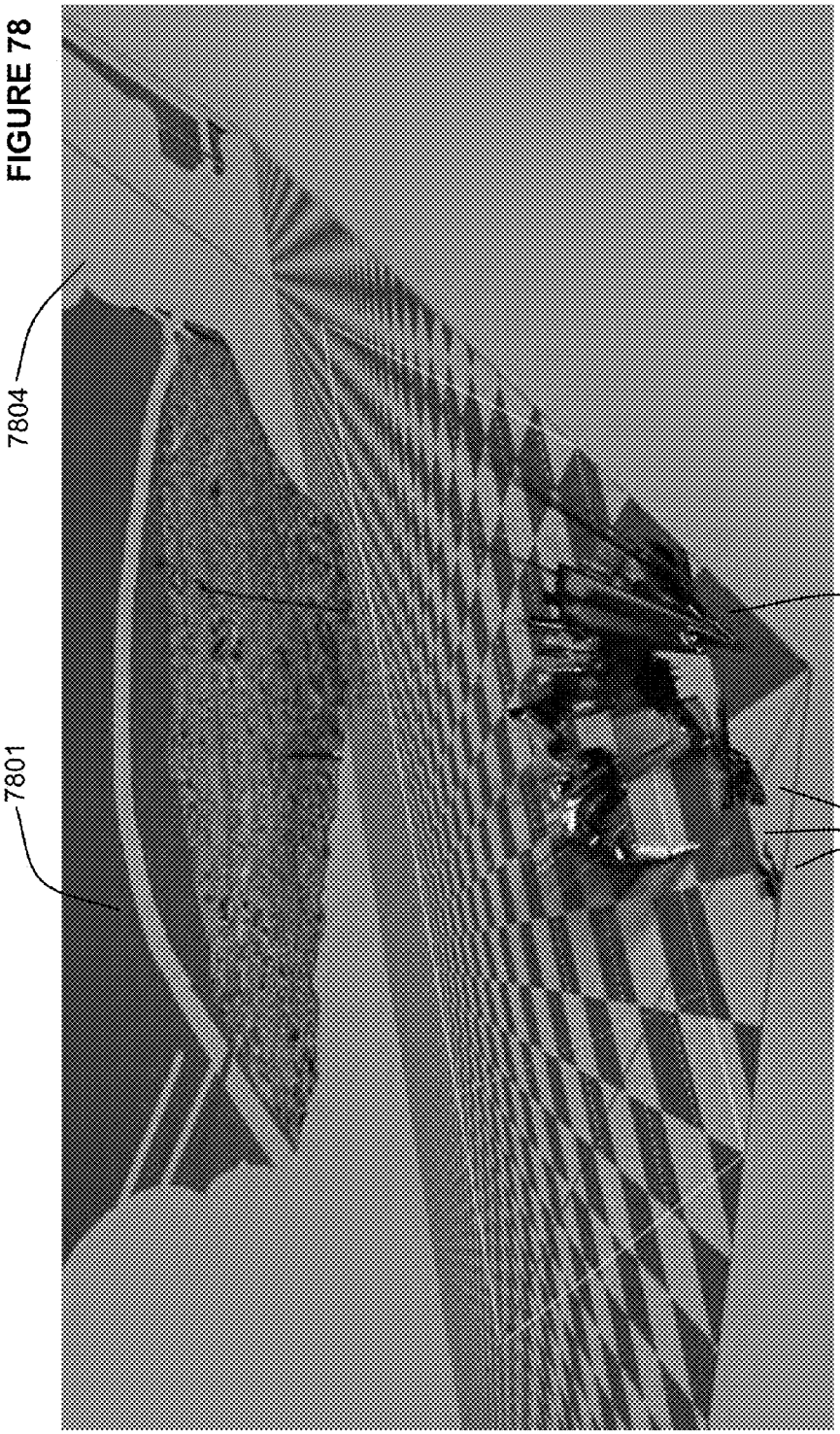


FIGURE 79

FIGURE 80

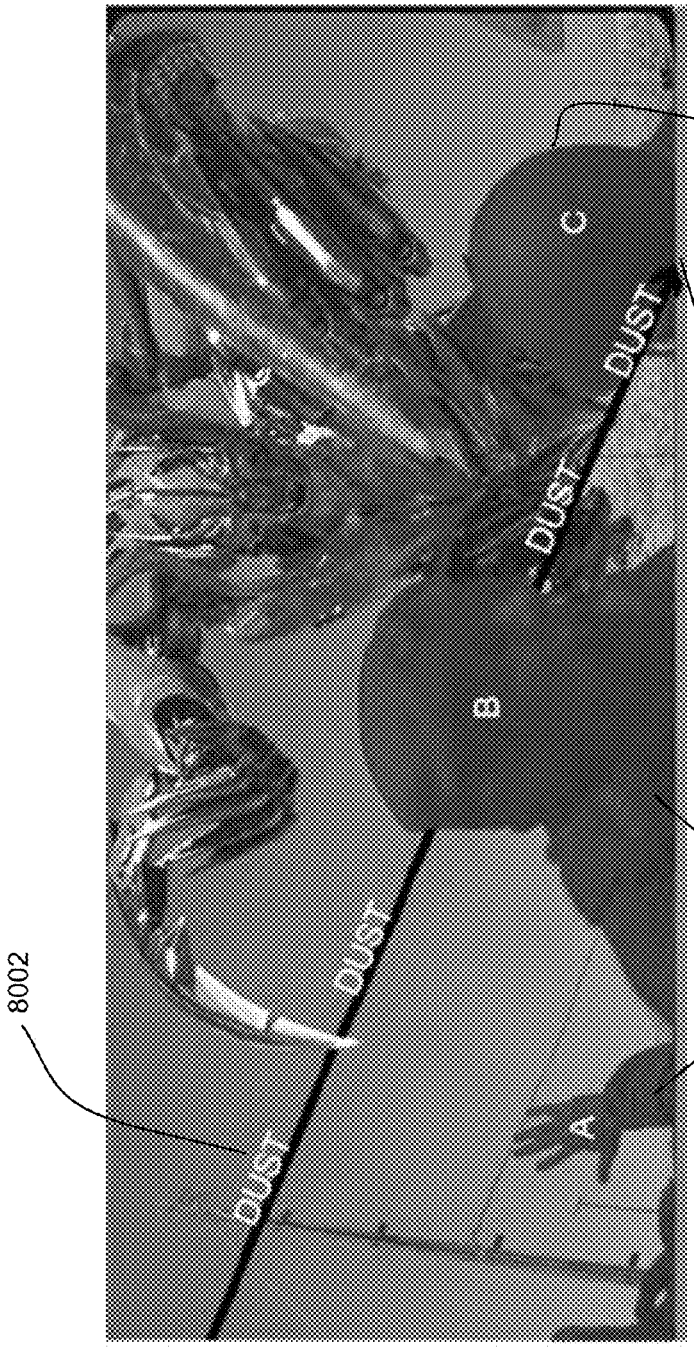


FIGURE 81

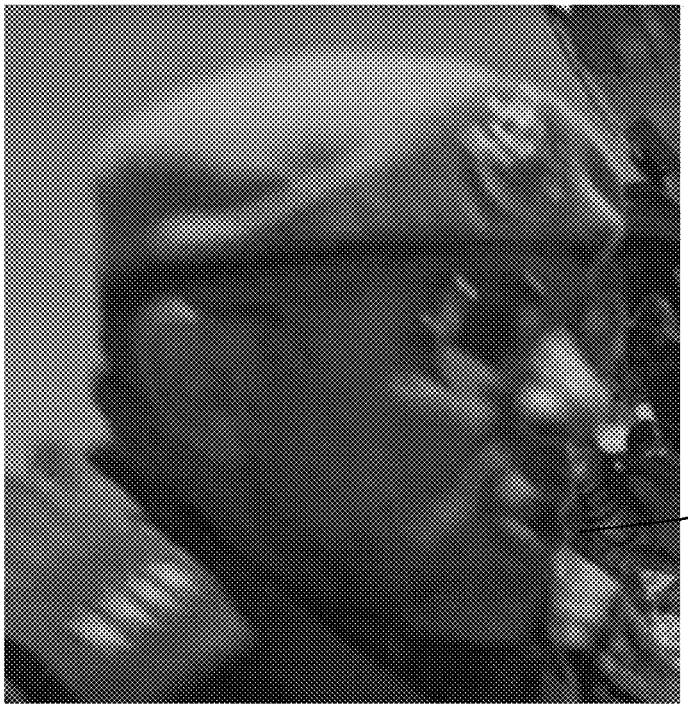


FIGURE 82

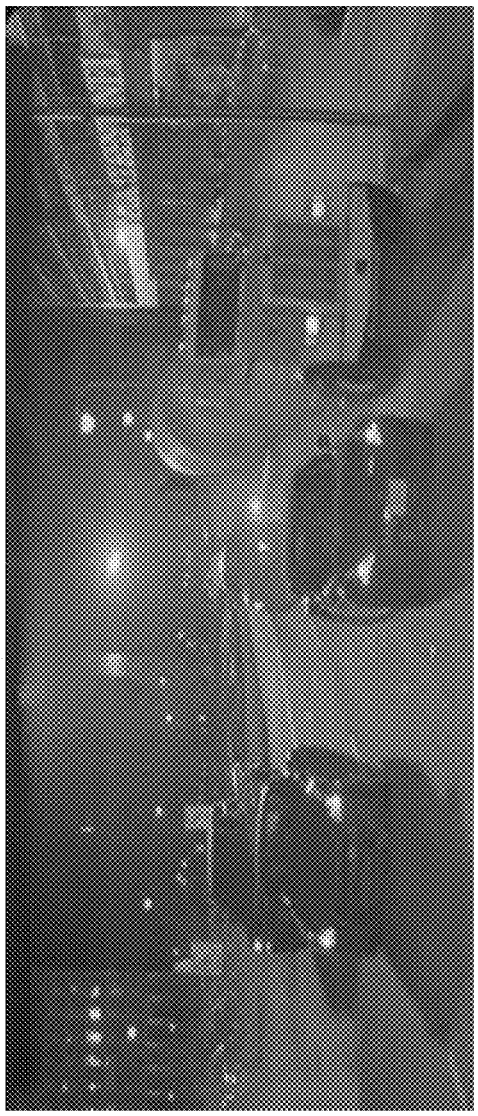


FIGURE 83

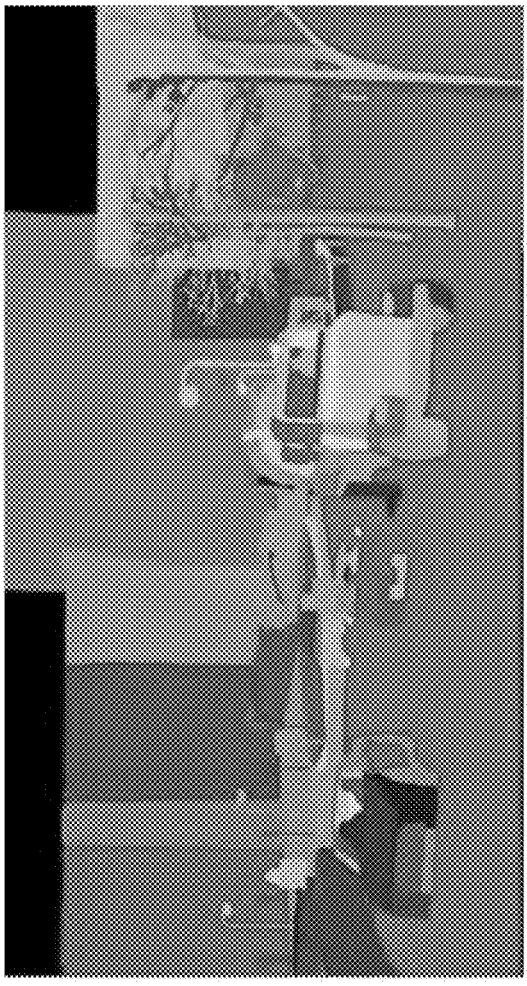


FIGURE 84

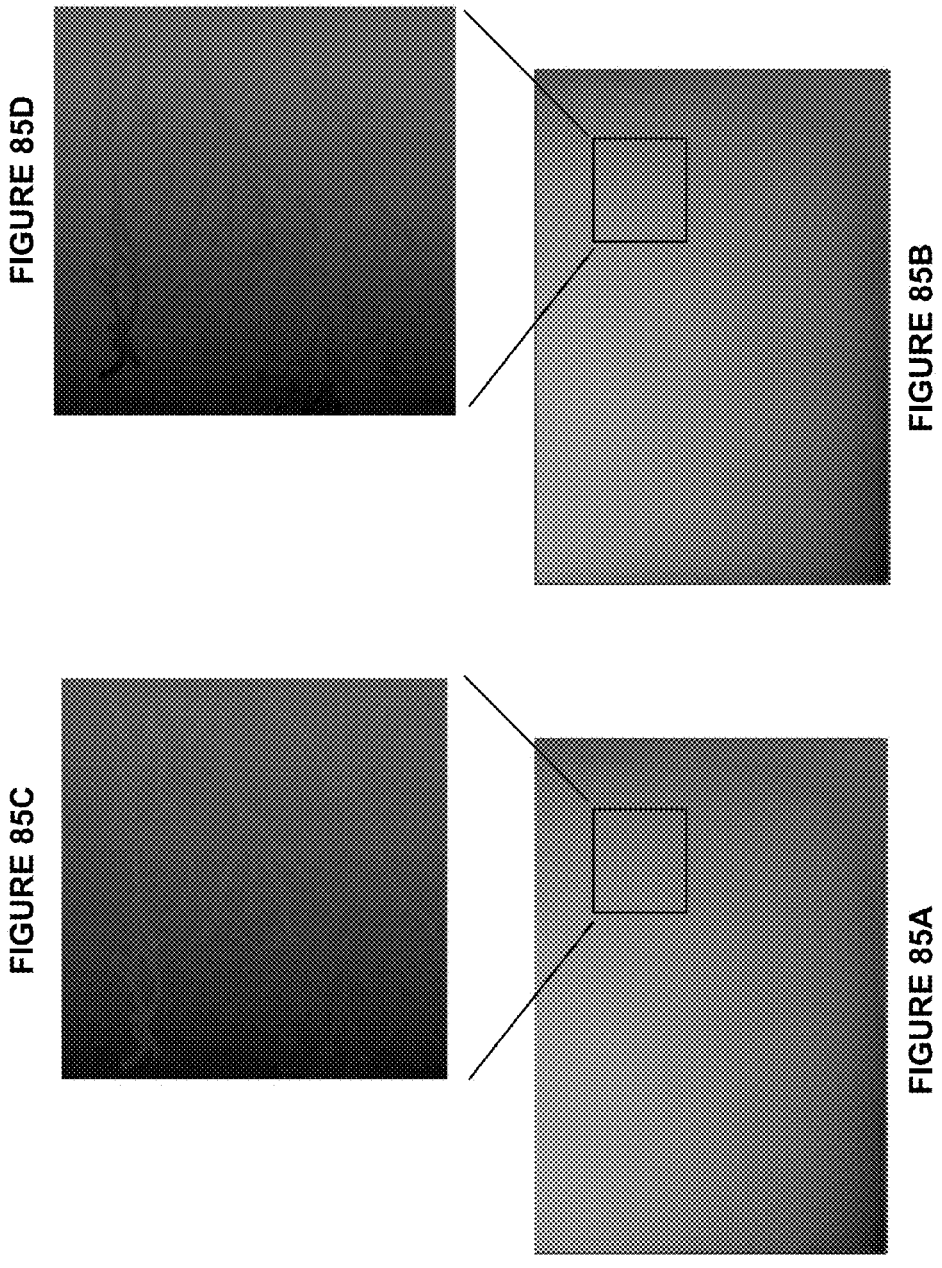


FIGURE 85D
FIGURE 85B
FIGURE 85C
FIGURE 85A

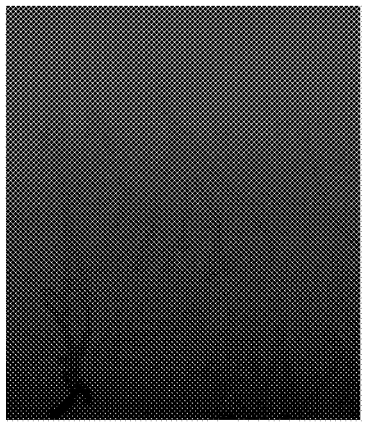


FIGURE 86D

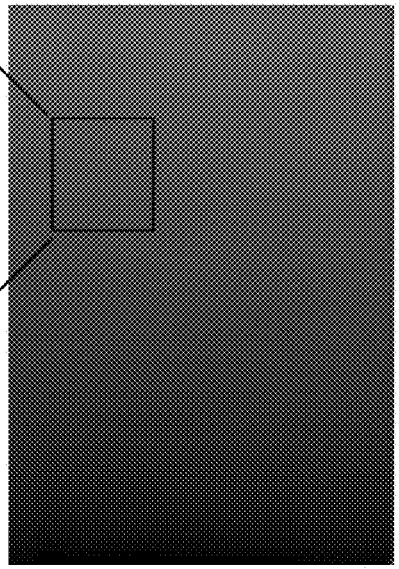


FIGURE 86B

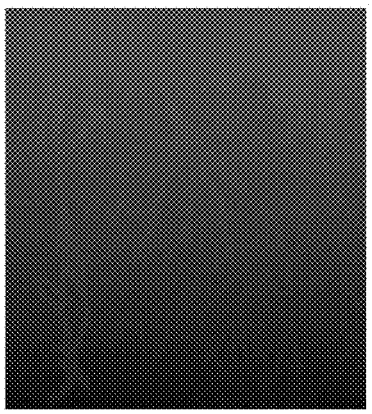


FIGURE 86C

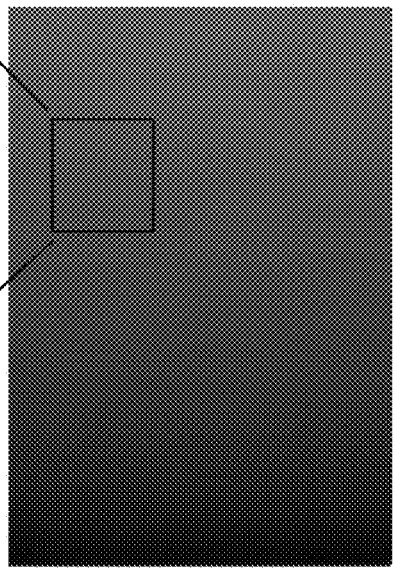


FIGURE 86A

FIGURE 87

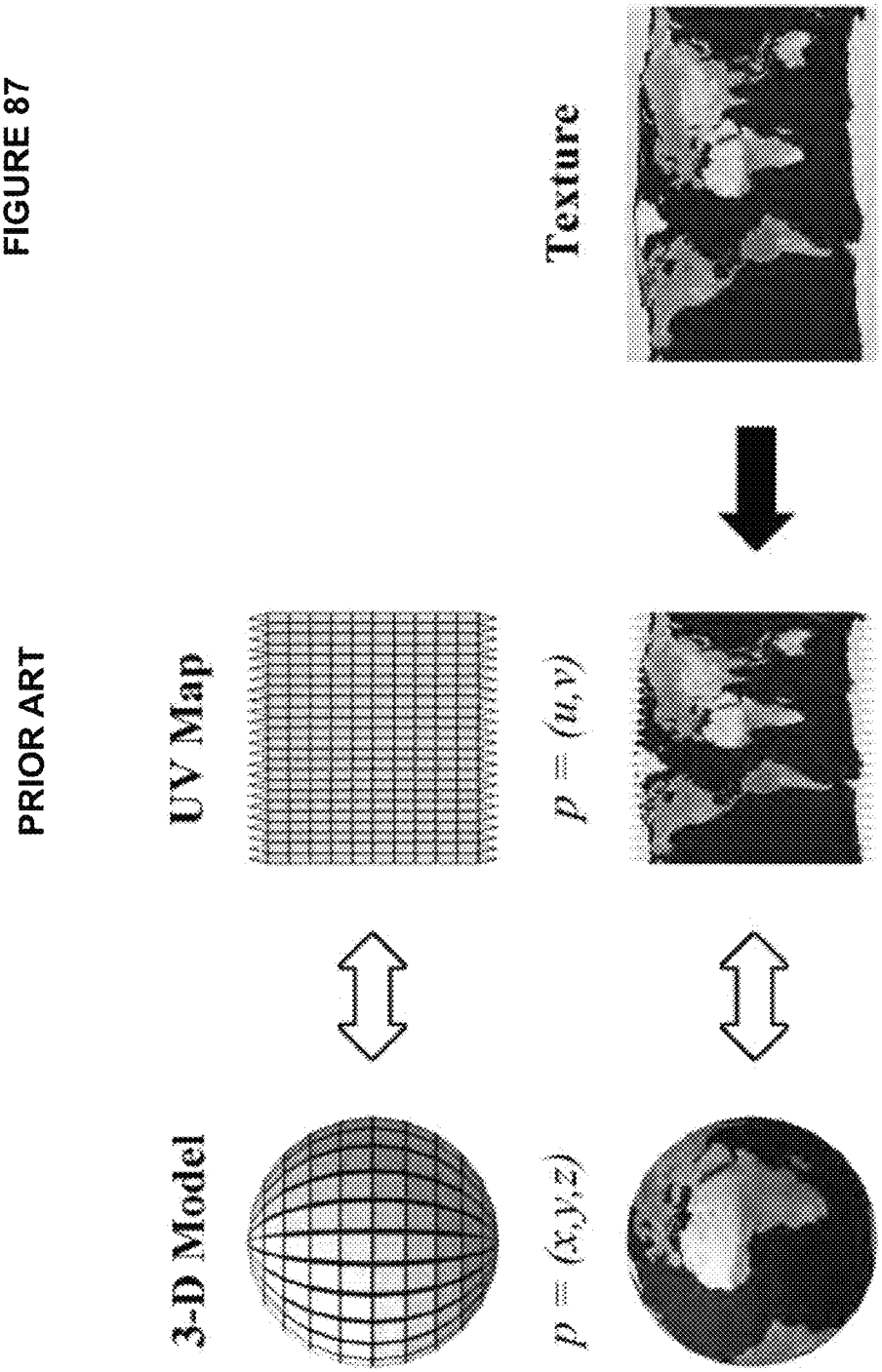


FIGURE 88

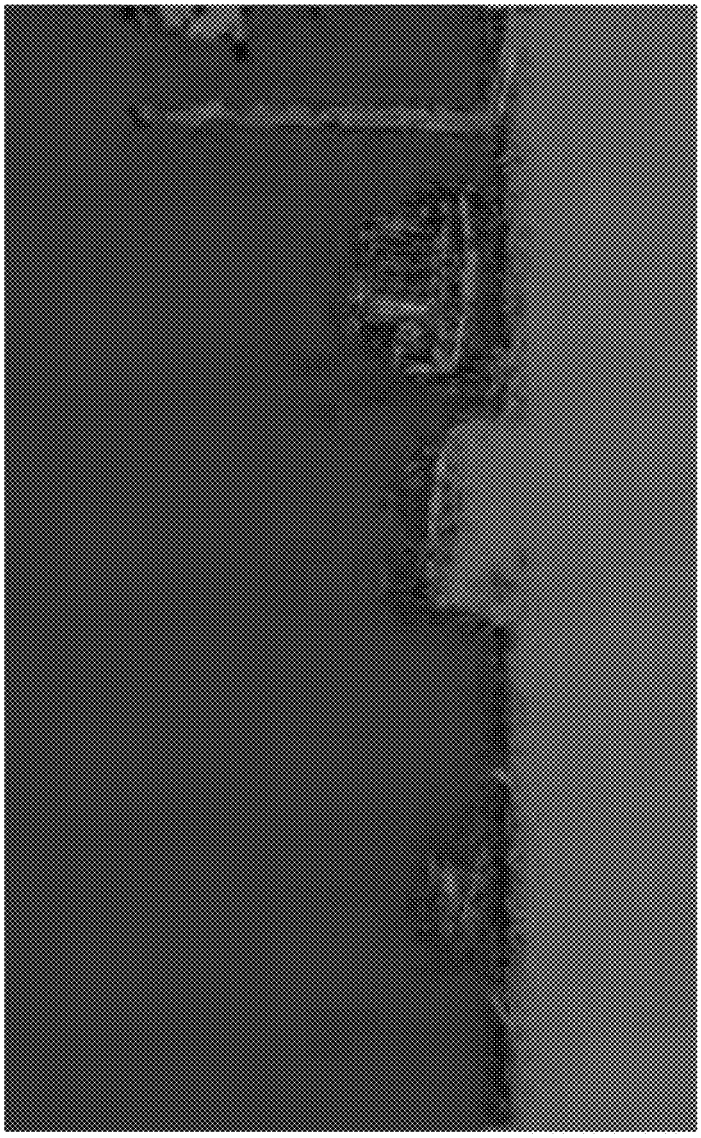


FIGURE 89

FIGURE 90

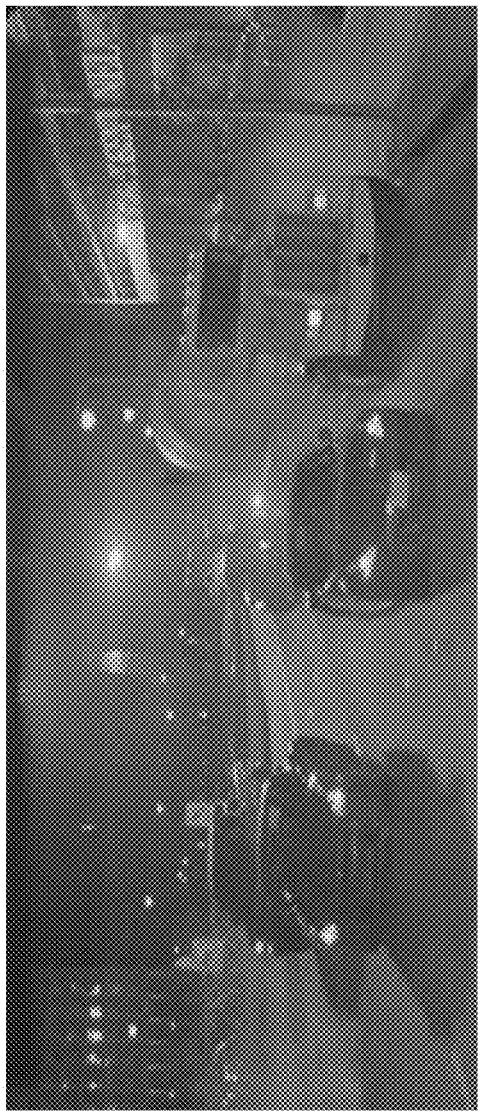


FIGURE 91

FIGURE 92

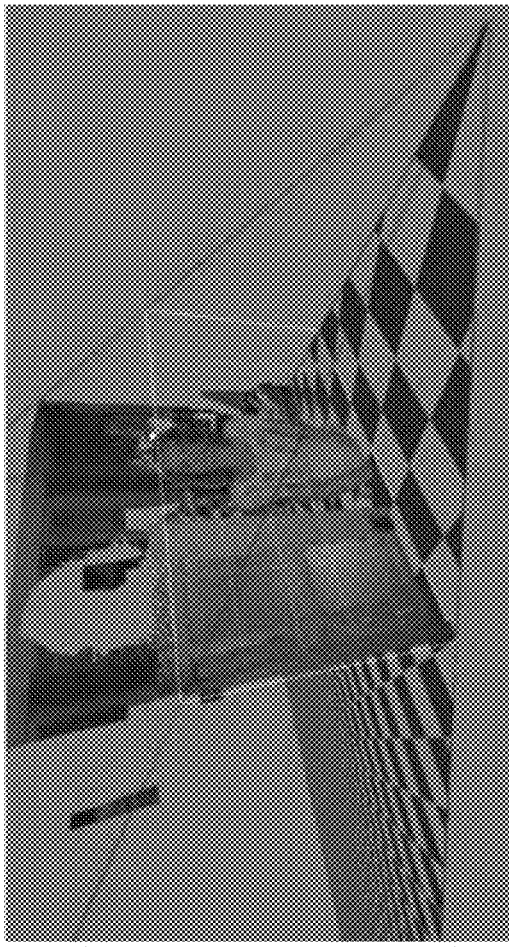


FIGURE 93

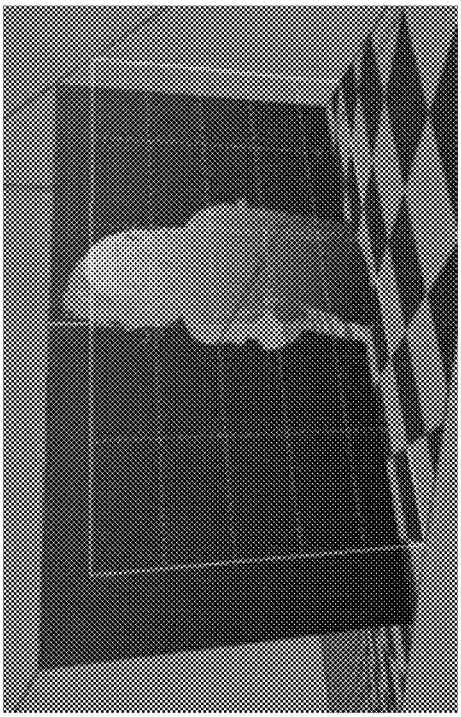


FIGURE 94

Figure 95

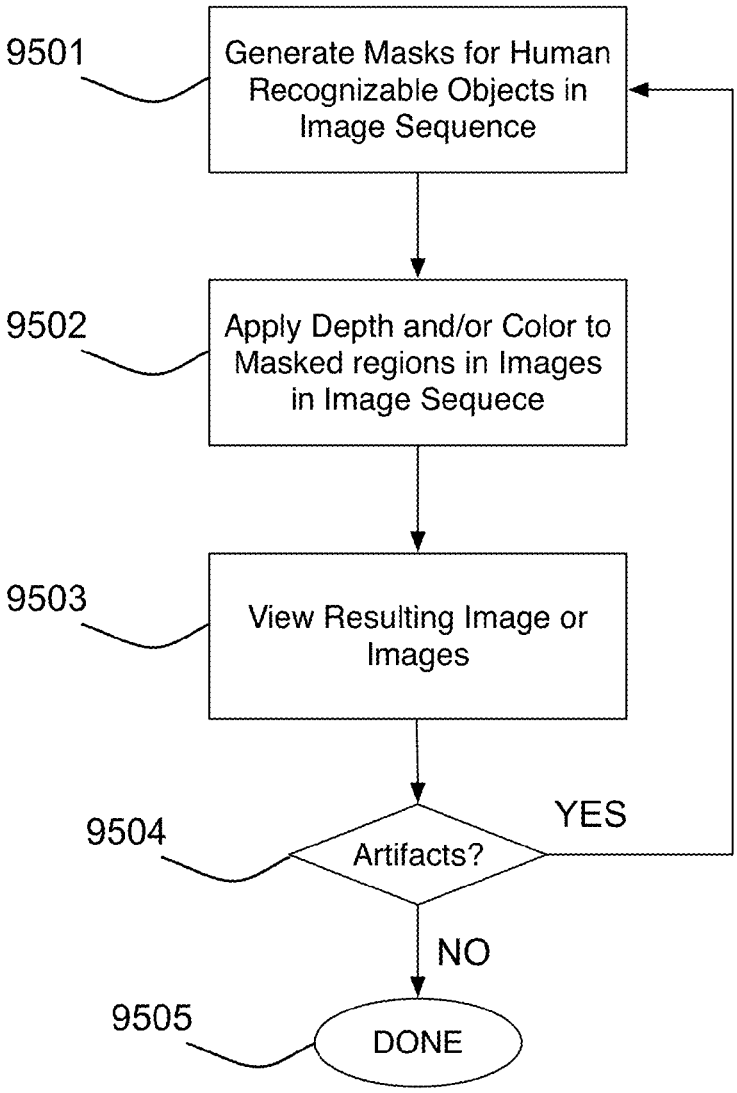

9501
Generate Masks for Human Recognizable Objects in Image Sequence
9502
Apply Depth and/or Color to Masked regions in Images in Image Sequece
9503
View Resulting Image or Images
9504
Artifacts?
YES
NO
9505
DONE

Figure 96

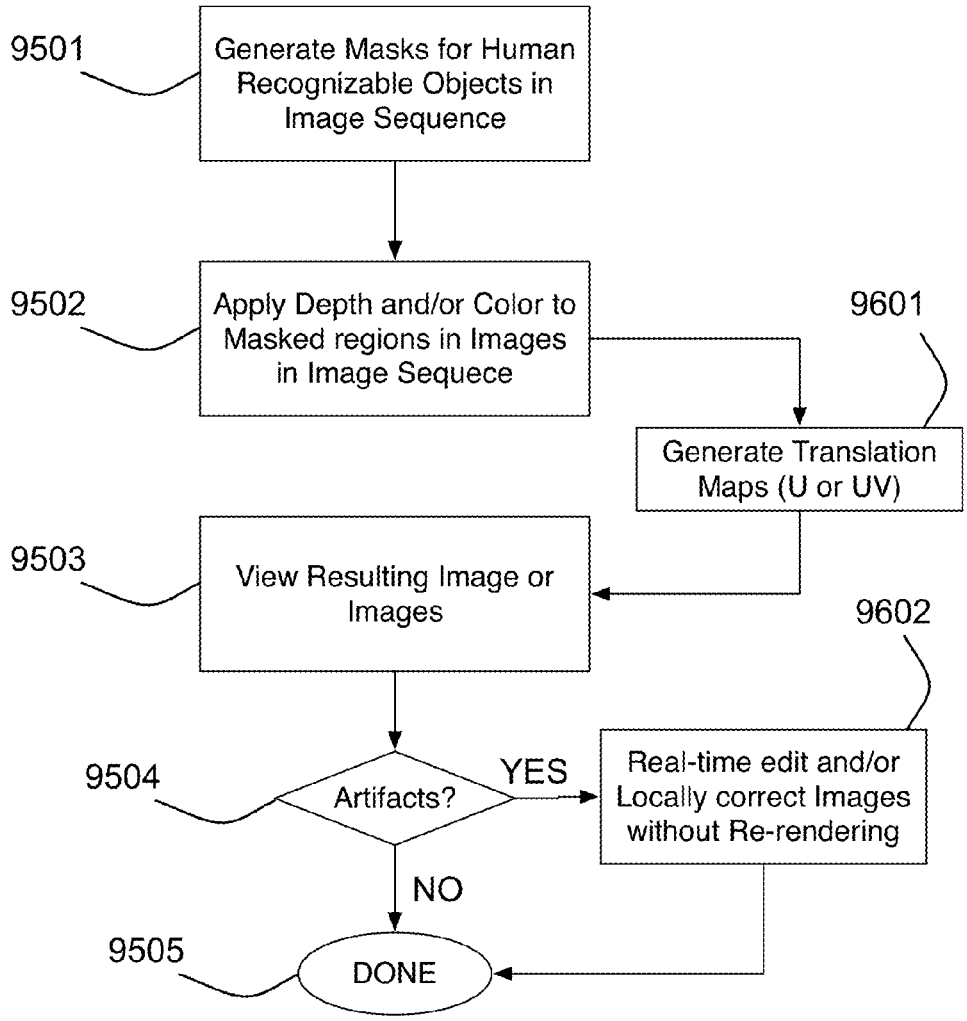

9501
Generate Masks for Human Recognizable Objects in Image Sequence
9502
Apply Depth and/or Color to Masked regions in Images in Image Sequece
9601
Generate Translation Maps (U or UV)
9503
View Resulting Image or Images
9602
9504
Artifacts?
YES
Real-time edit and/or Locally correct Images without Re-rendering
NO
9505
DONE

FIGURE 97

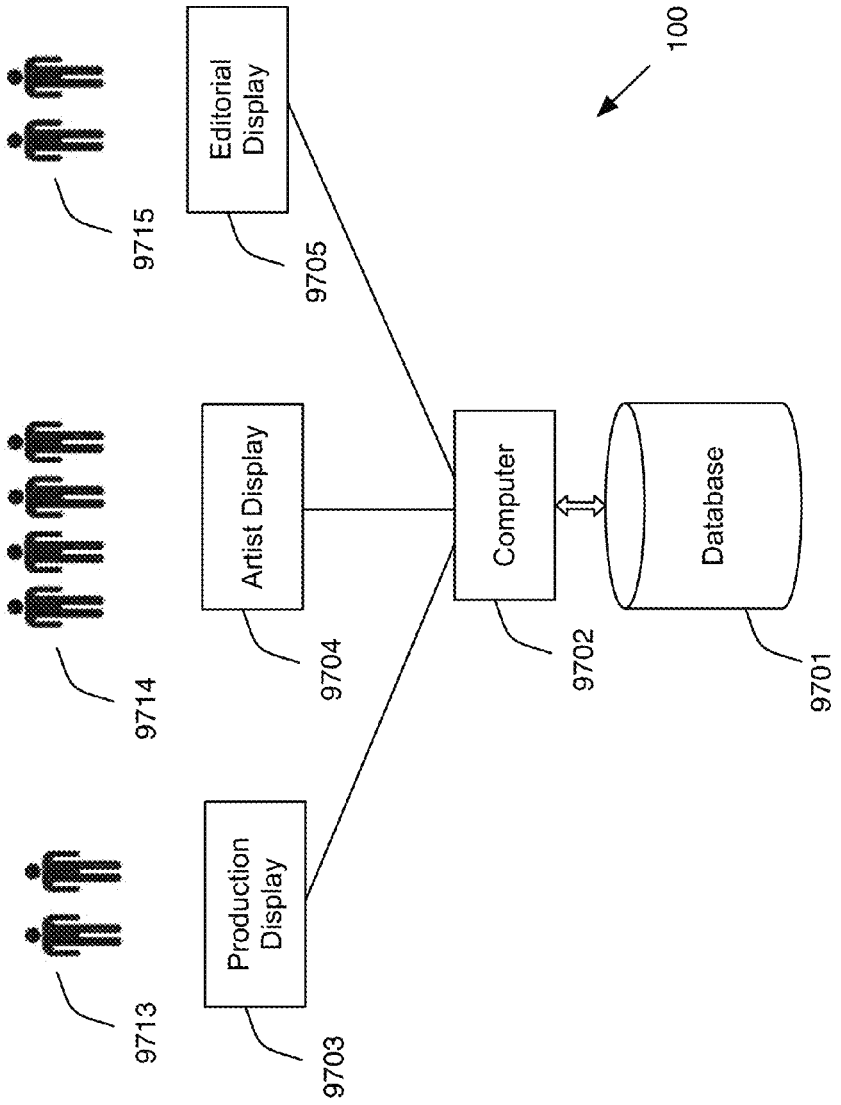

100
Editorial Display
9705
9715
Artist Display
9704
9714
Production Display
9703
9713
Computer
9702
Database
9701

FIGURE 98

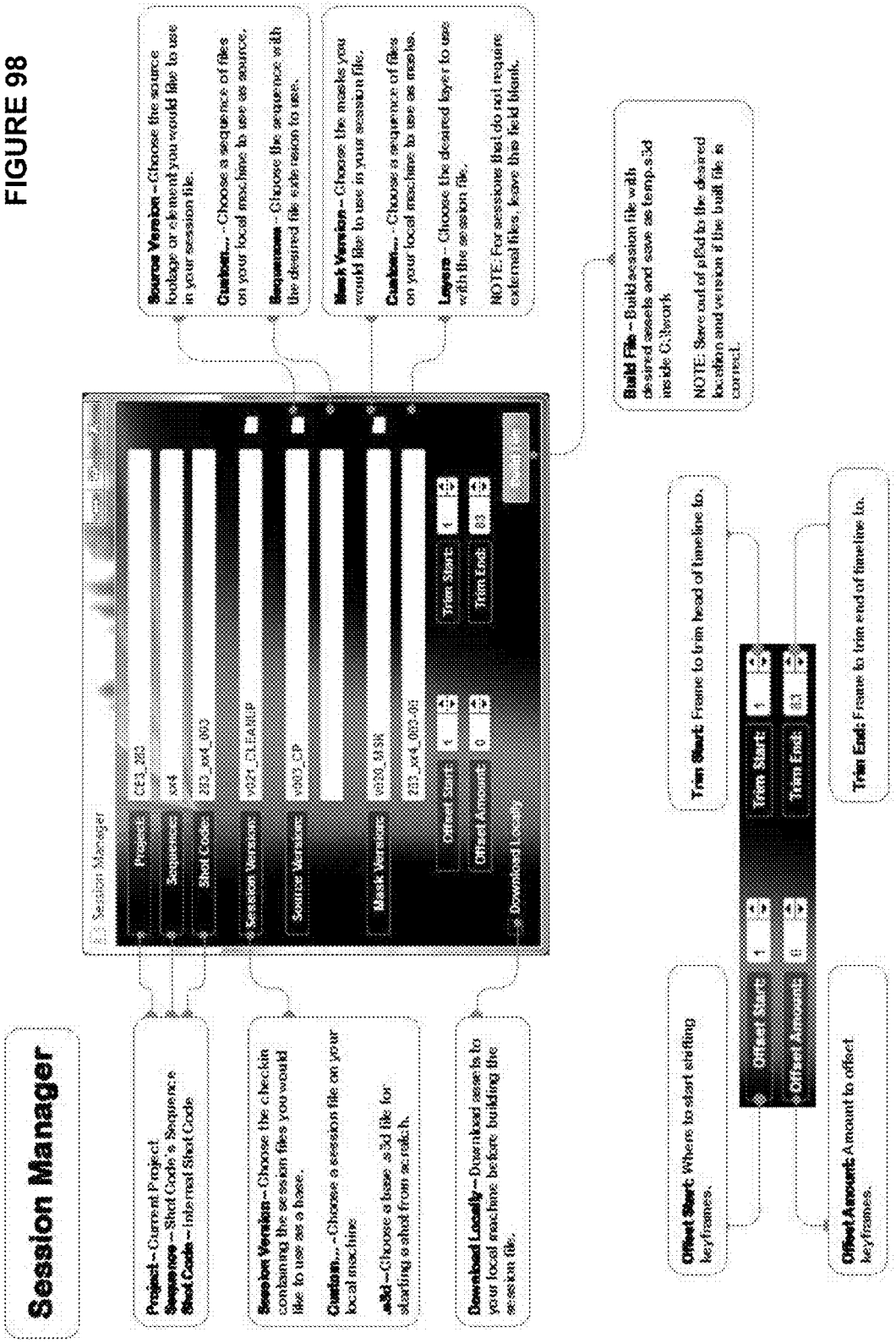

Session Manager
Project – Current Project
Sequence – Shot Code's Sequence
Shot Code – Internal Shot Code
Session Version – Choose the checkin containing the session files you would like to use as a base.
Custom... – Choose a session file on your local machine
.s3d – Choose a base .s3d file for starting a shot from scratch.
Download Locally – Download assets to your local machine before building the session file.
Session Manager
Project:
Sequence:
Shot Code:
Session Version:
Source Version:
Mask Version:
Offset Start:
Offset Amount:
Download Locally
Trim Start:
Trim End:
Source Version – Choose the source footage or element you would like to use in your session file.
Custom... – Choose a sequence of files on your local machine to use as source.
Sequence – Choose the sequence with the desired file extension to use.
Mask Version – Choose the masks you would like to use in your session file.
Custom... – Choose a sequence of files on your local machine to use as masks.
Layers – Choose the desired layer to use with the session file.
NOTE: For sessions that do not require external files, leave this field blank.
Build File – Build session file with desired assets and save as temp.s3d inside C:\Work
NOTE: Save out of p8d to the desired location and version if the built file is correct.
Offset Start: Where to start shifting keyframes.
Offset Amount: Amount to offset keyframes.
Trim Start: Frame to trim head of timeline to.
Trim End: Frame to trim end of timeline to.

FIGURE 99

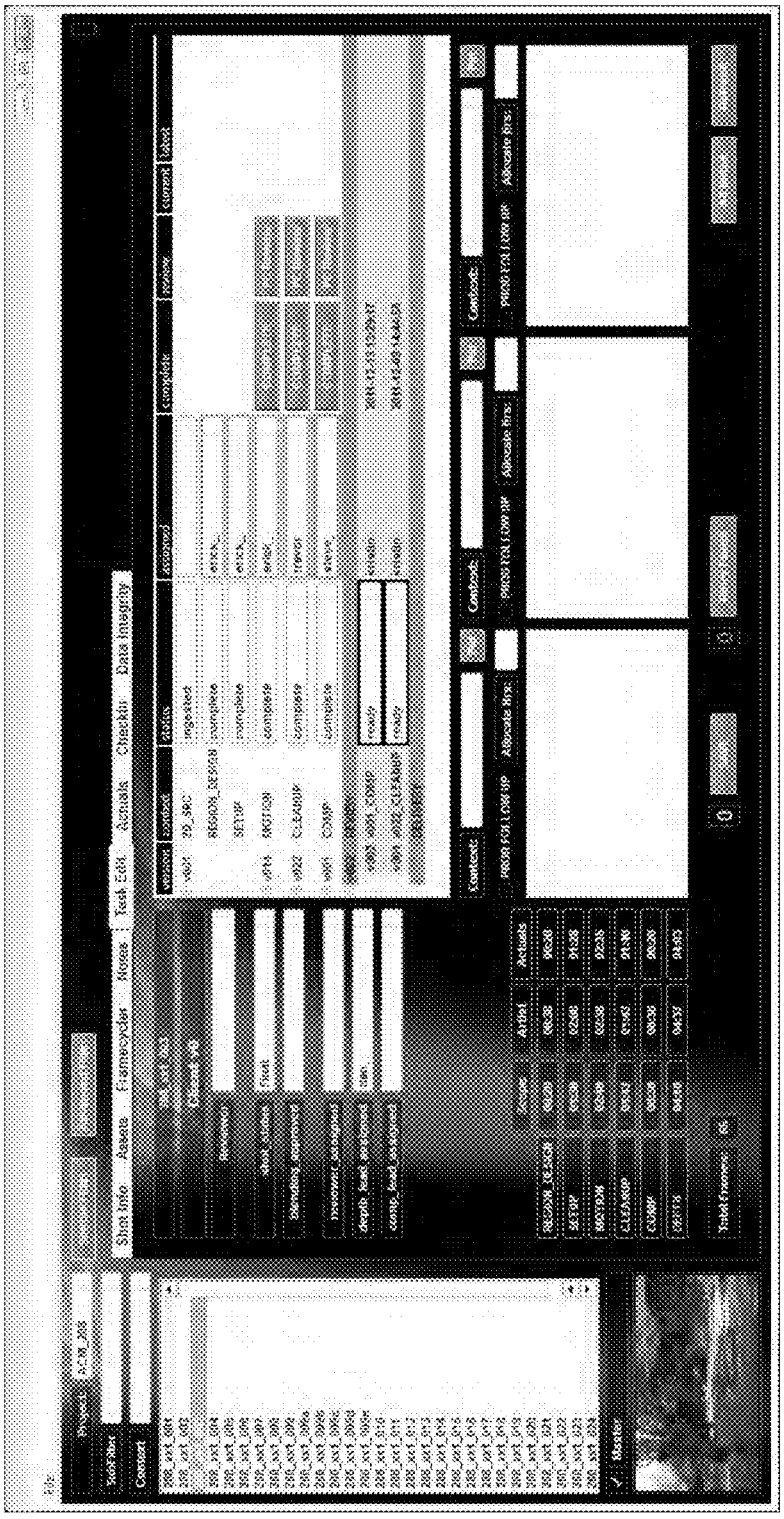


FIGURE 100

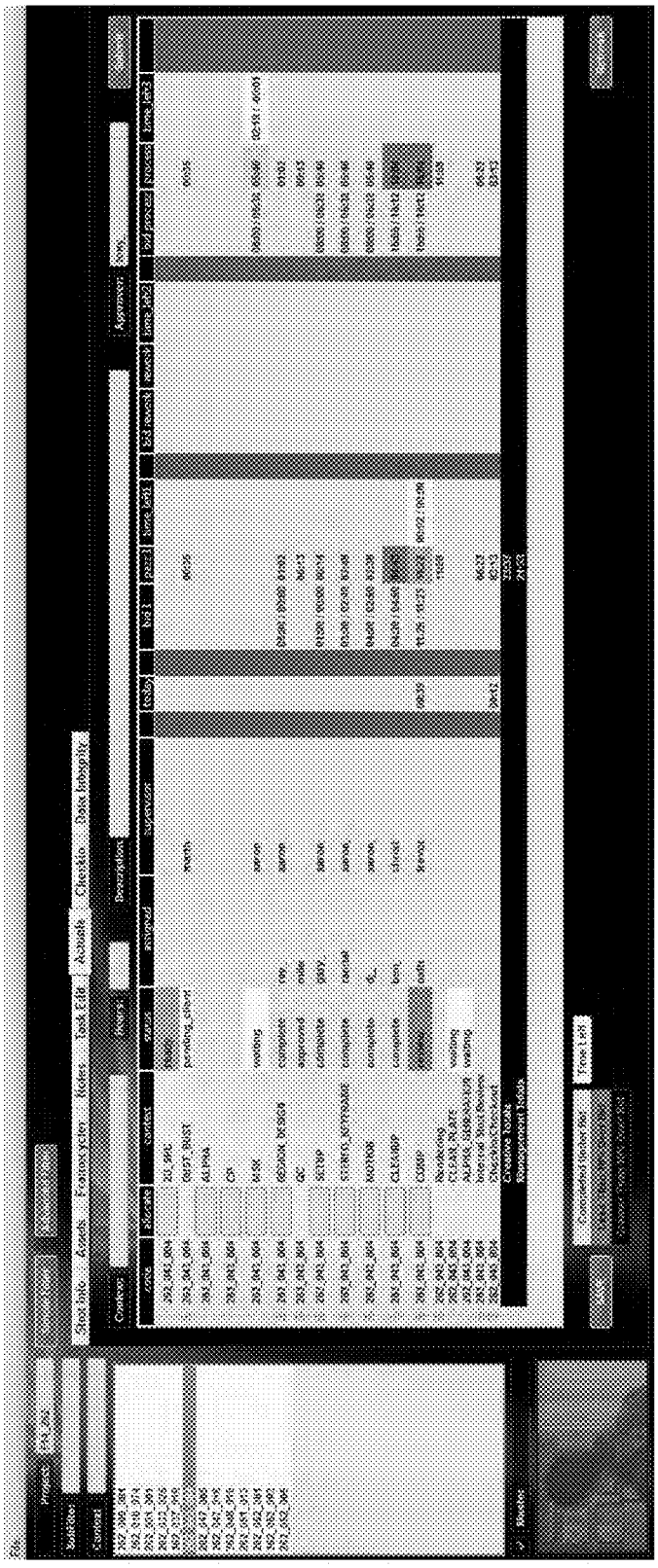


FIGURE 101

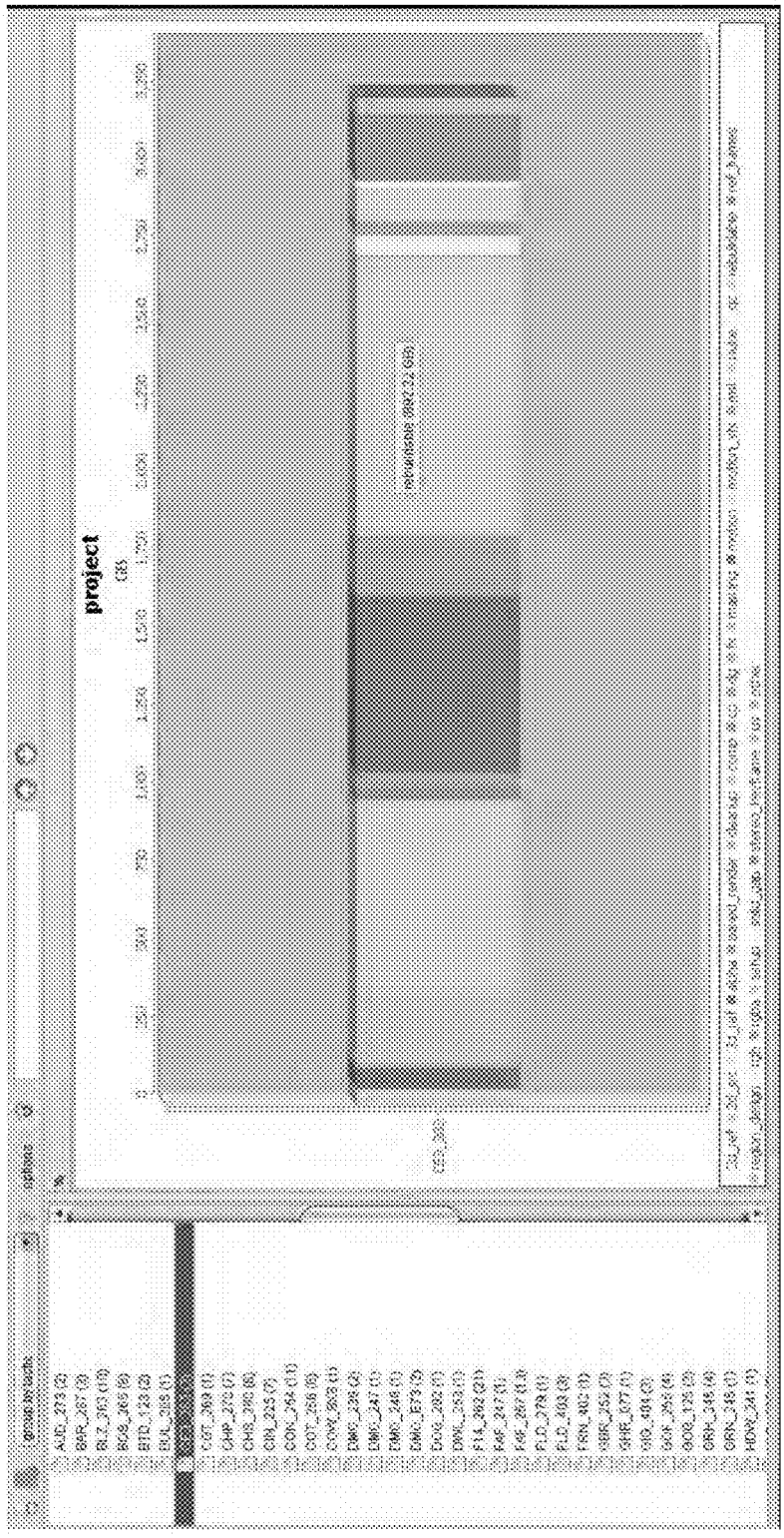


FIGURE 102

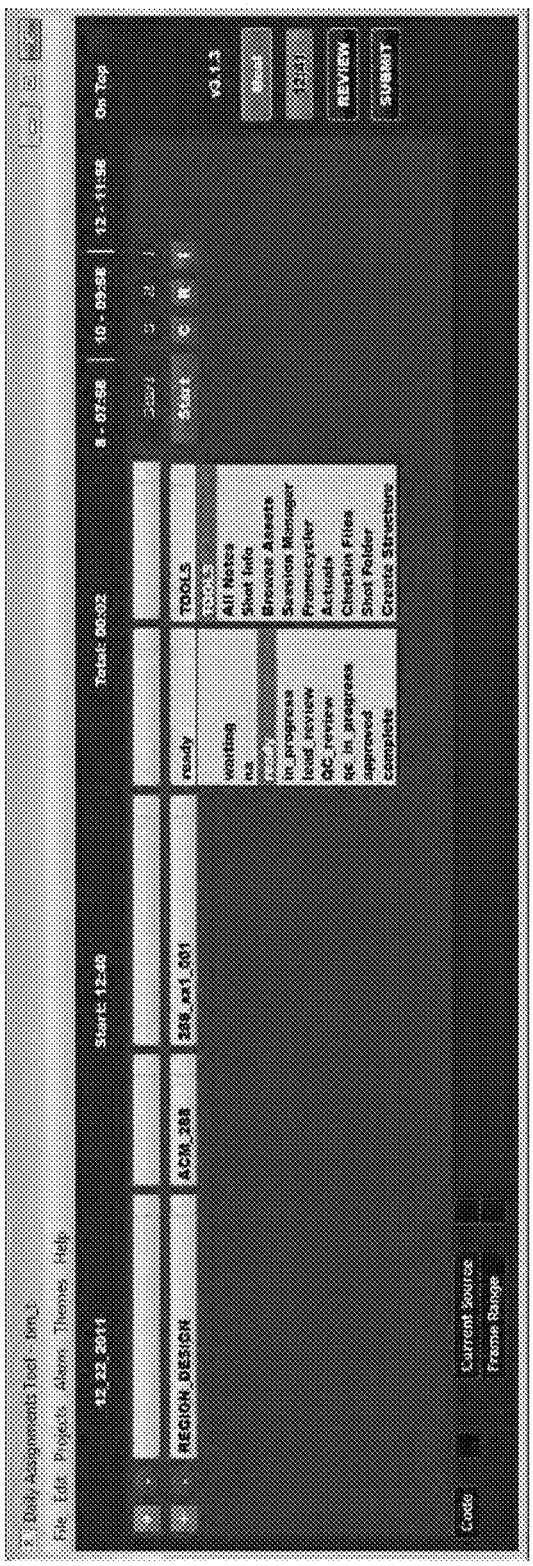

Daily Assignments Tool - tm_1
File Edit Projects Alarm Themes Help
12_22_2011
Start: 12:40
Total: 00:02
8 - 07:58
10 - 09:58
12 - 11:58
On Top
REGION_DESIGN
ACM_288
258_xx1_001
ready
waiting
na
in_progress
lead_review
QC_review
qc_in_progress
approved
complete
TOOLS
All Notes
Shot Info
Browse Assets
Session Manager
Framecycler
Actuals
Checkin Files
Shot Folder
Create Structure
Start
v3.1.3
Mod
REVIEW
SUBMIT
Code
Current Source
Frame Range

FIGURE 103

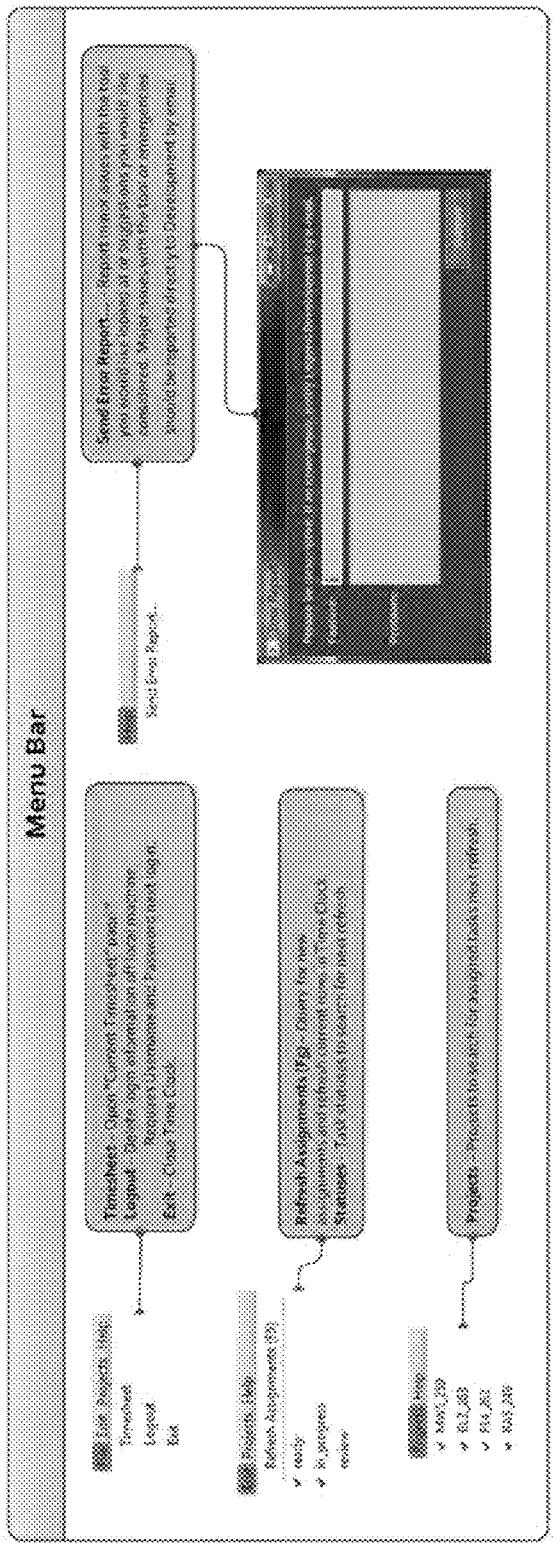


FIGURE 104

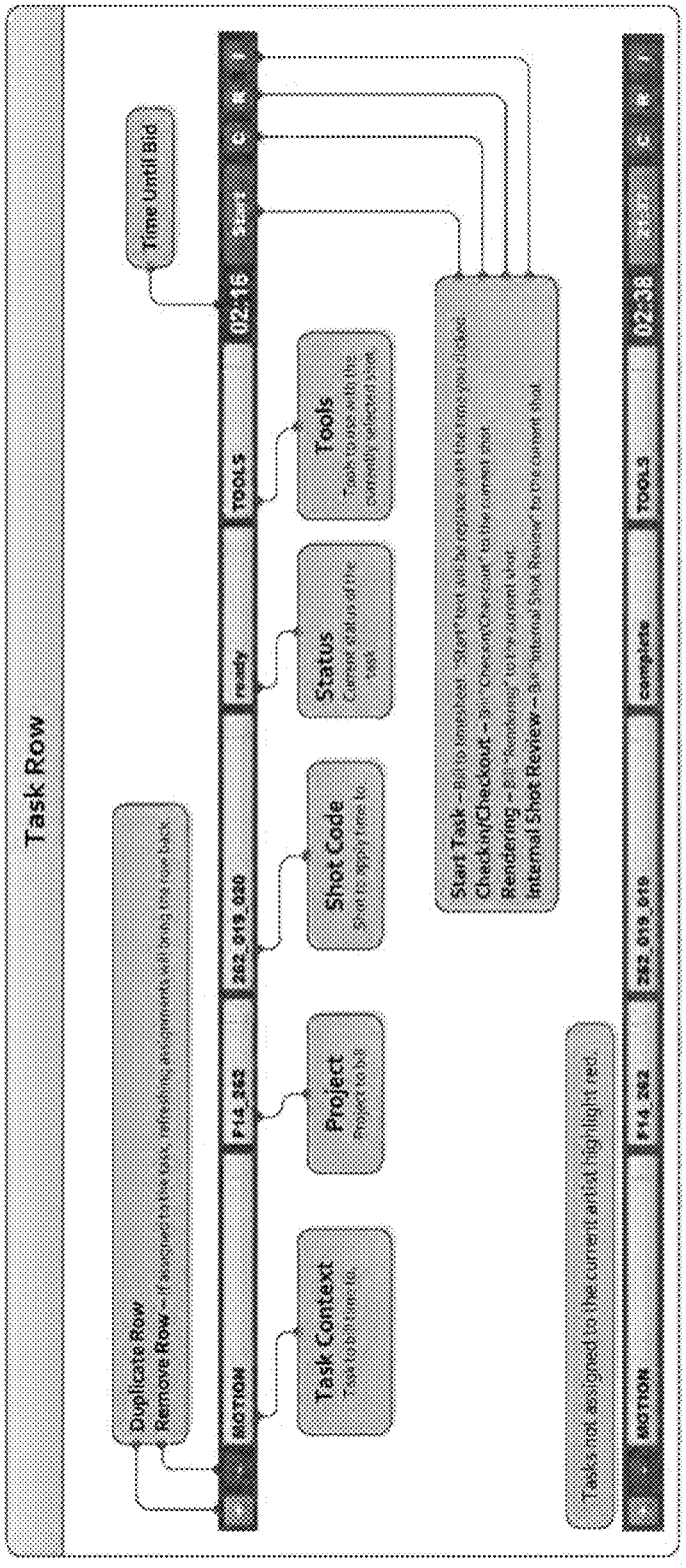

Task Row
Time Until Bid
02:16
02:38
Tools
Status
Shot Code
Project
Task Context
Duplicate Row
Remove Row
Start Task
Checkin/Checkout
Rendering
Internal Shot Review
Tasks not assigned to the current artist highlight red.

FIGURE 105

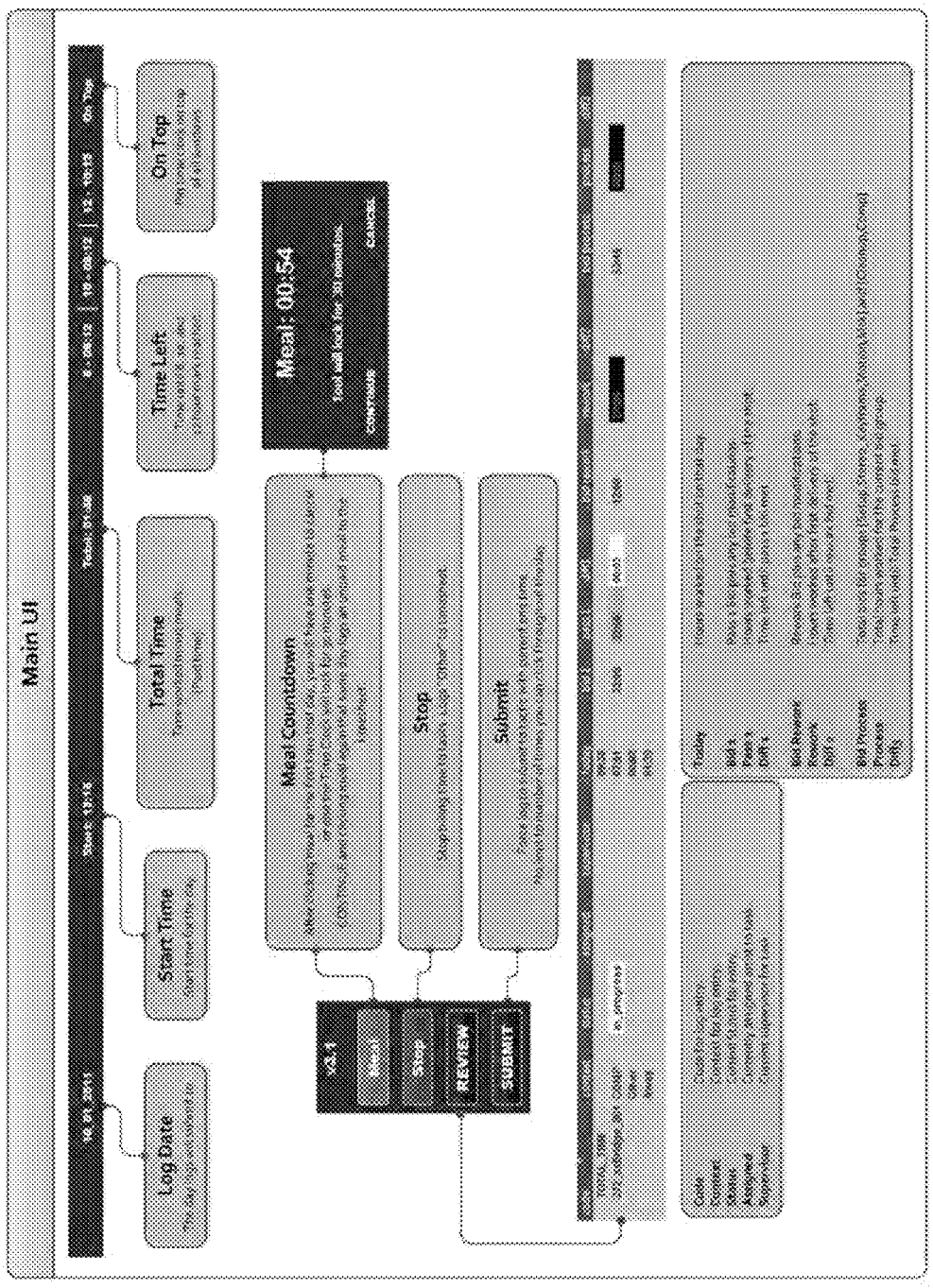

Main UI
On Top
Time Left
Total Time
Start Time
Log Date
Meal 00:54
Meal Countdown
Stop
Submit
Stop
REVIEW
SUBMIT

FIGURE 106

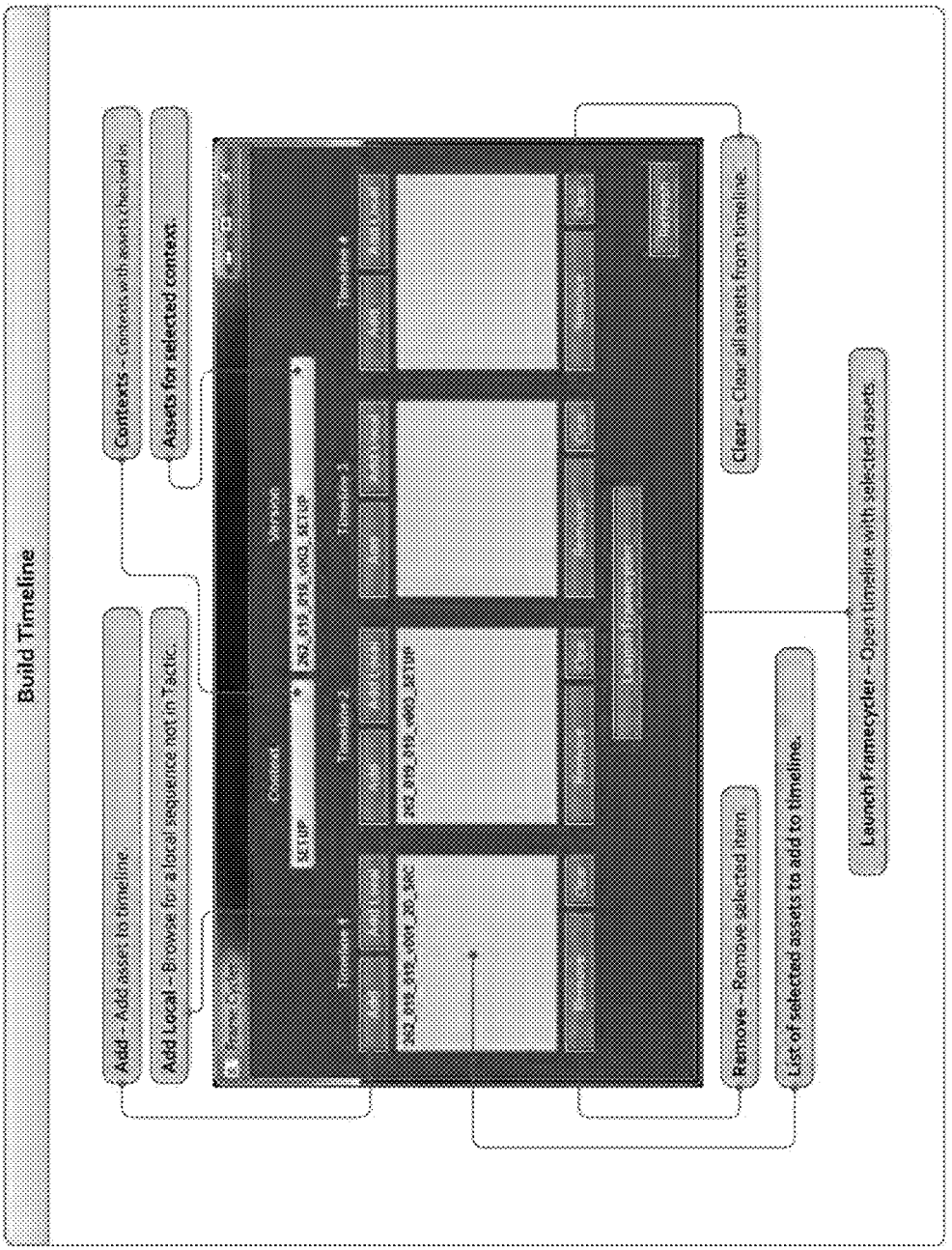

Build Timeline
Contents – Contents with assets checked in
Assets for selected context
Add – Add assets to timeline
Add Local – Browse for a local sequence not in Tactic
Clear – Clear all assets from timeline
Launch Framecycler – Open timeline with selected assets
Remove – Remove selected item
List of selected assets to add to timeline

FIGURE 107

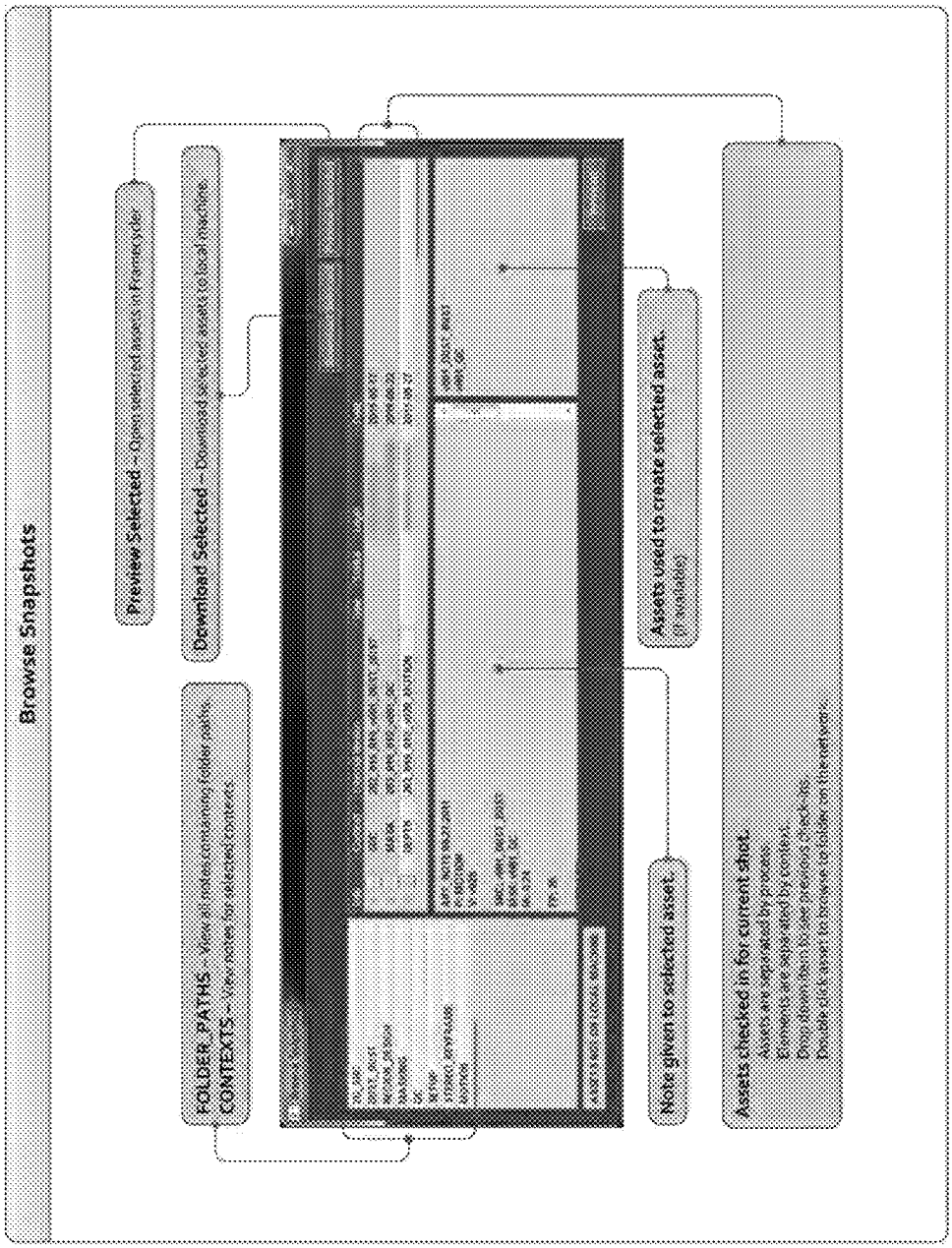

Browse Snapshots
Preview Selected
Download Selected
FOLDER_PATHS
CONTEXTS
Assets used to create selected asset.
Note given to selected asset
Assets checked in for current shot.

FIGURE 108

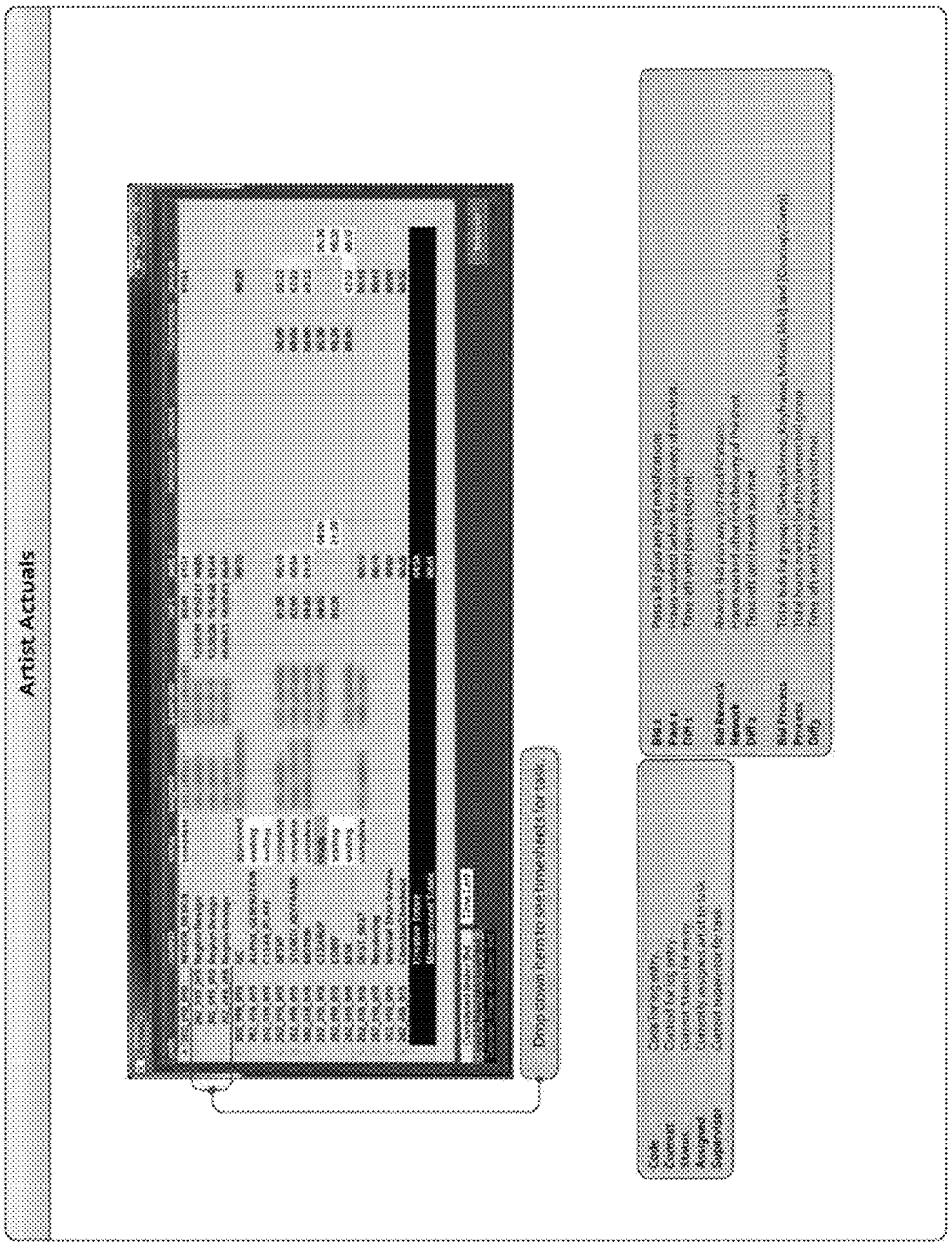


FIGURE 109

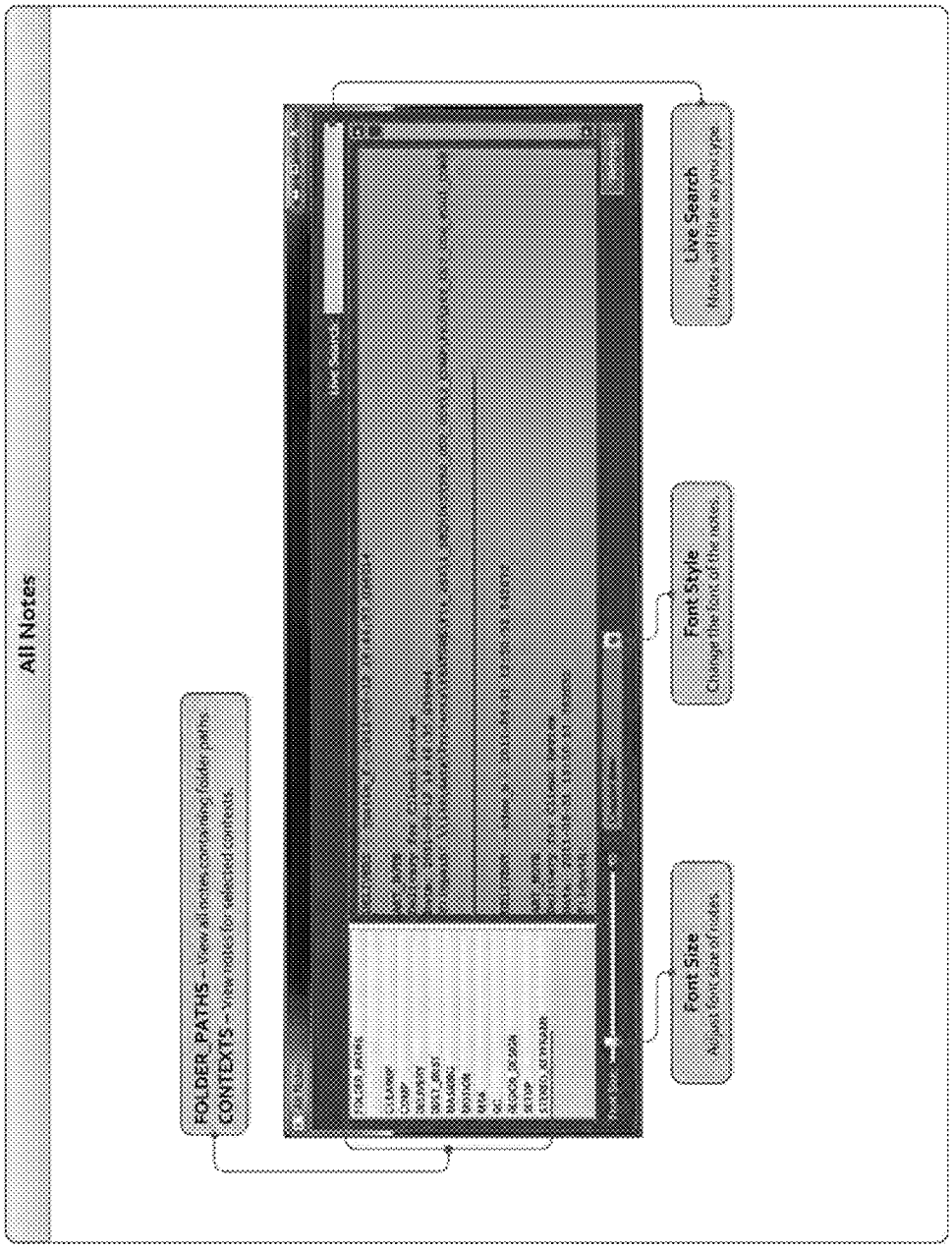

All Notes
Live Search
Font Style
Font Size
FOLDER PATHS
CONTEXTS

FIGURE 110

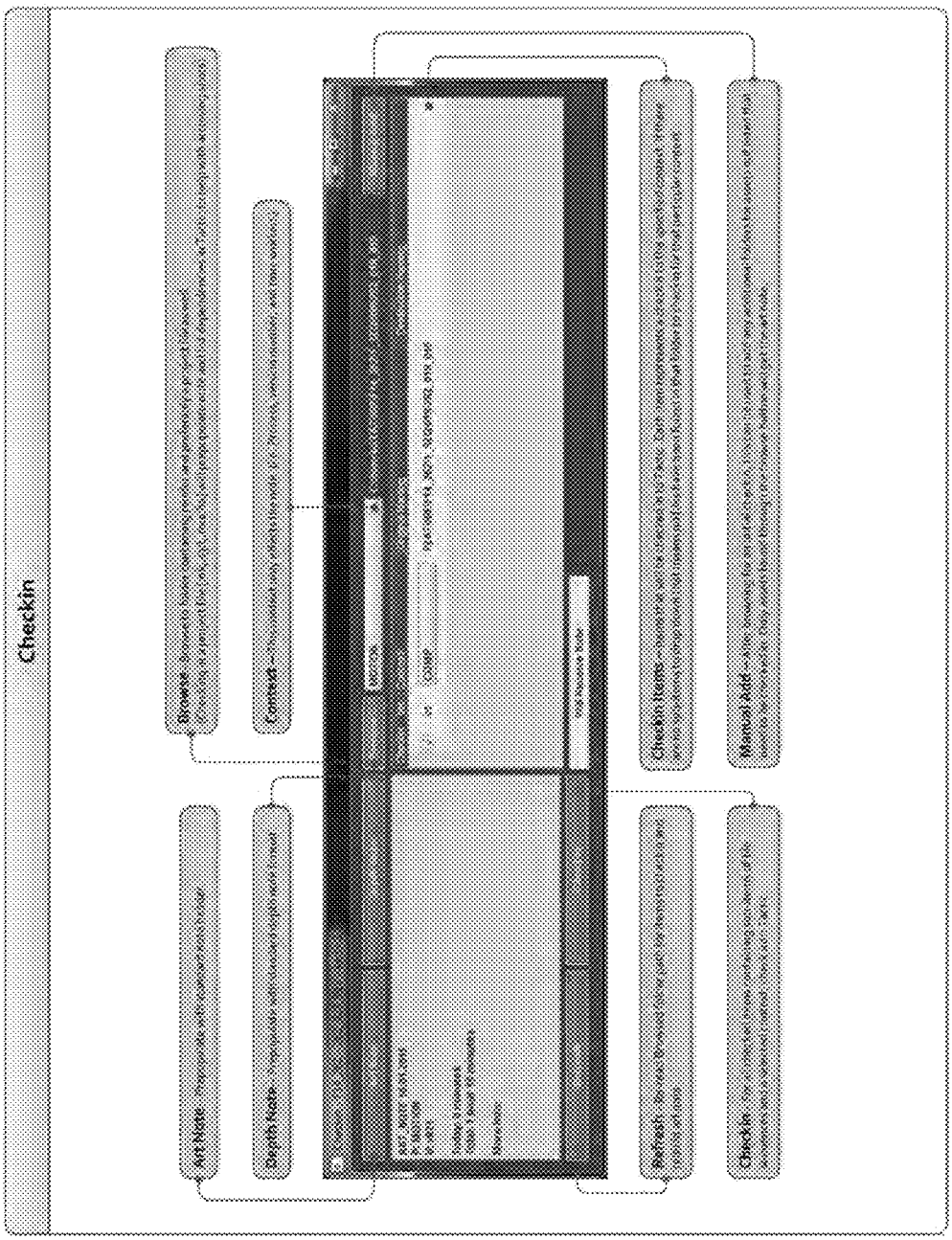

Checkin
Browse
Forced
Checkin Items
Manual Add
Art Note
Depth Note
Retreat
Checkin

FIGURE 111

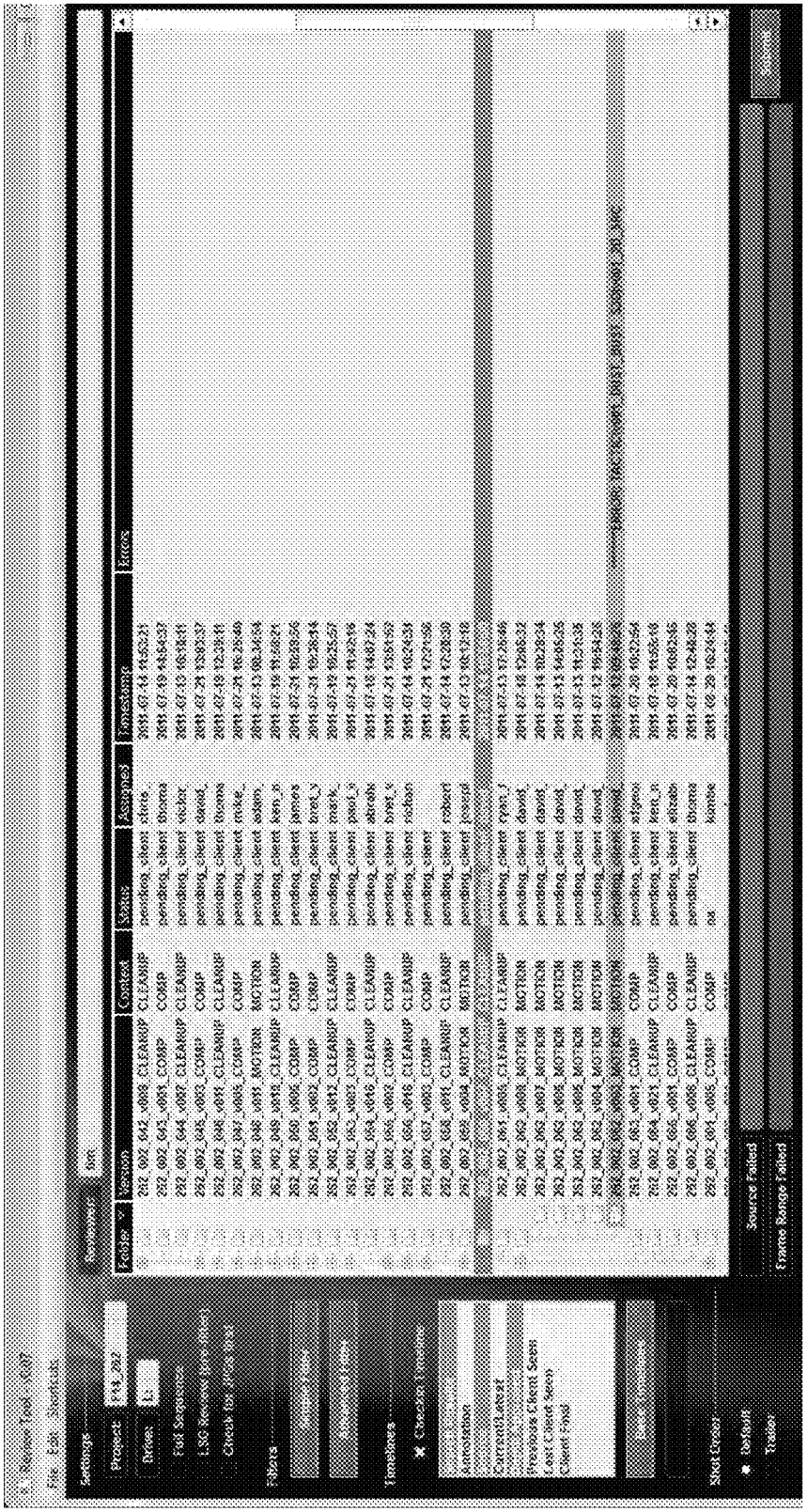


FIGURE 112

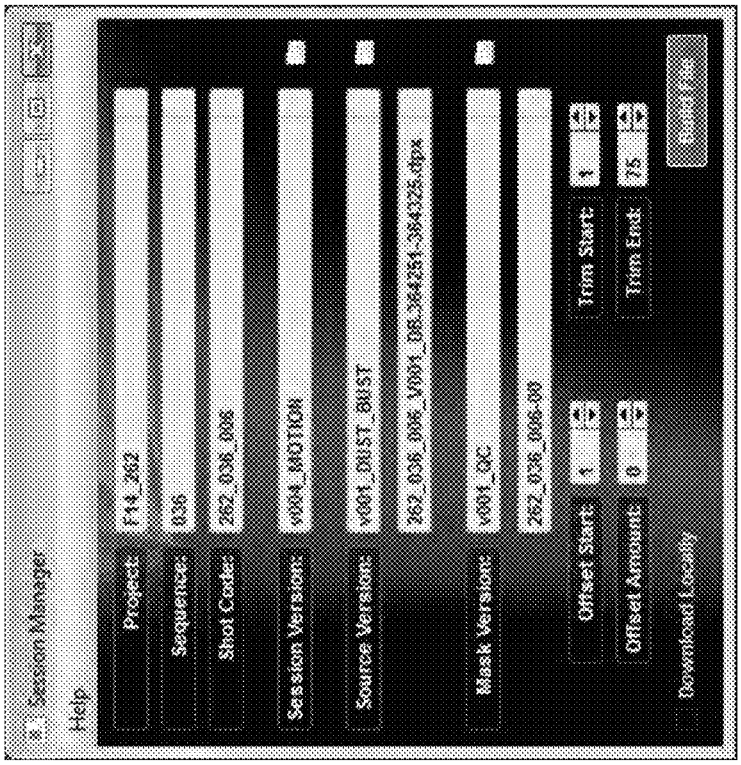

Session Manager
Help
Project:
F14_262
Sequence:
036
Shot Code:
262_036_006
Session Version:
v004_MOTION
Source Version:
v001_DUST_BUST
262_036_006_V001_DB_364251-364325.dpx
Mask Version:
v001_QC
262_036_006-00
Offset Start:
1
Offset Amount:
0
Trim Start:
1
Trim End:
75
Download Locally

FIGURE 113

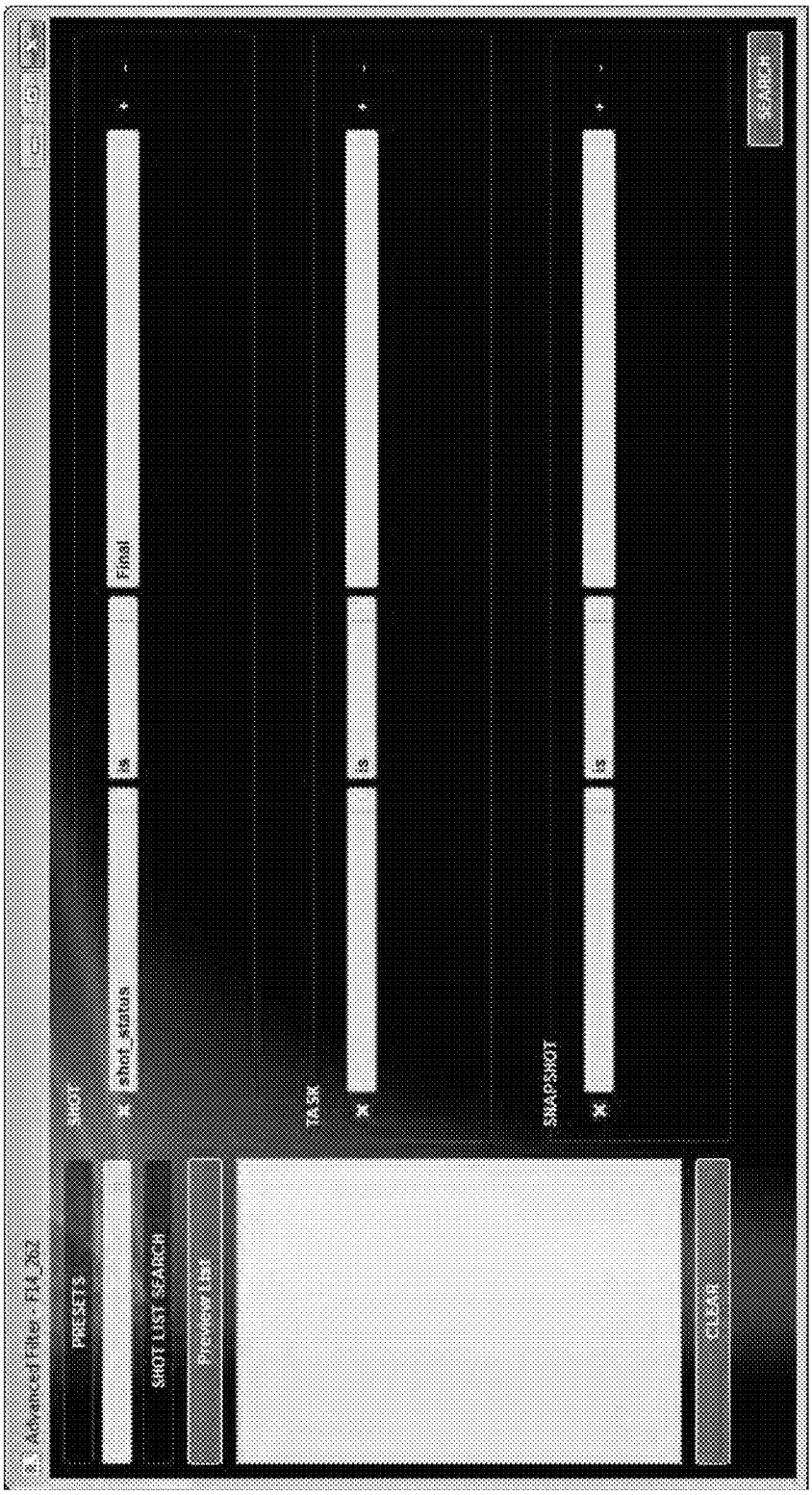

Advanced Filter - F14_262
PRESETS
SHOT LIST SEARCH
Preview List
CLEAR
SHOT
shot_status
is
Final
TASK
is
SNAPSHOT
is
SEARCH

FIGURE 114

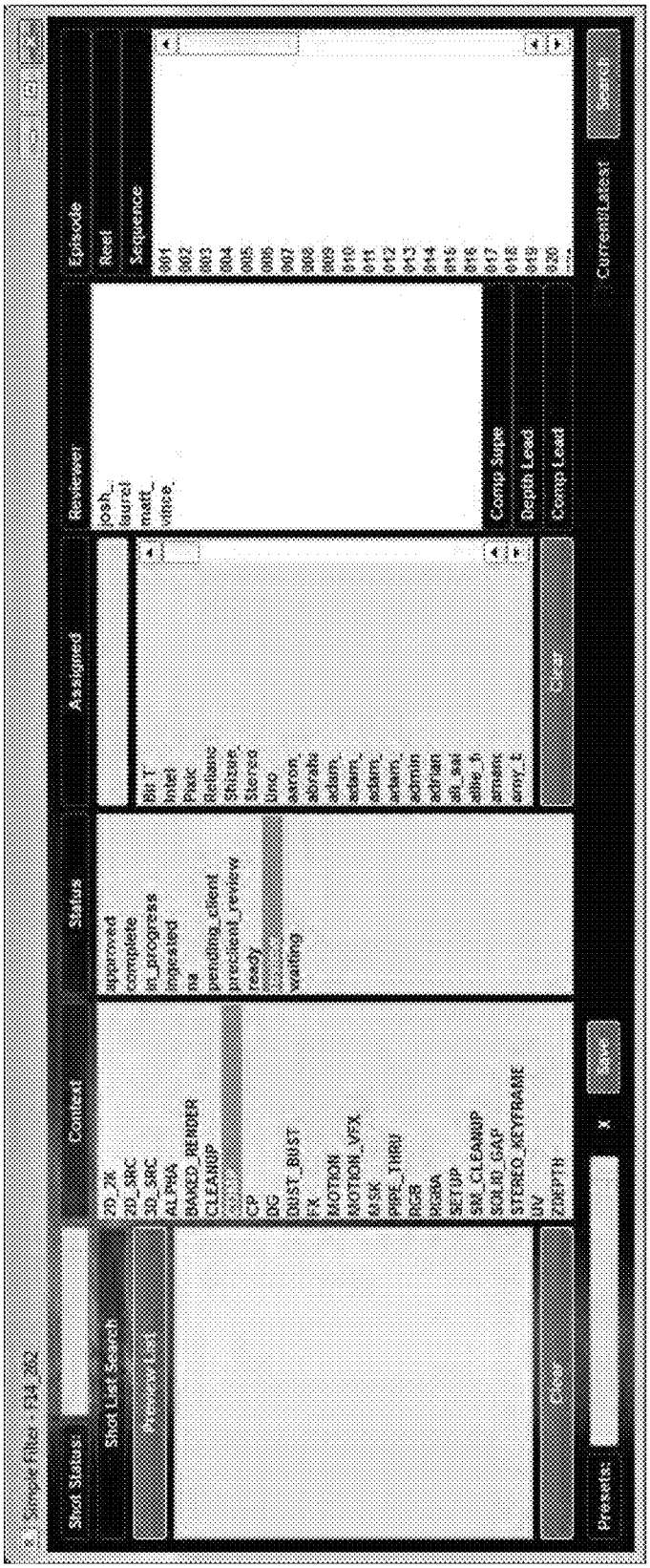

Shot Status
Shot List Search
Preview List
Clear
Context
2D_2K
2D_SRC
3D_SRC
ALPHA
BAKED_RENDER
CLEANUP
CP
DG
DUST_BUST
FX
MOTION
MOTION_VFX
MSK
PIPE_THRU
RGB
RGBA
SETUP
SM_CLEANUP
SOLID_GAP
STEREO_KEYFRAME
UV
ZDEPTH
Status
approved
complete
in_progress
ingested
na
pending_client
preclient_review
ready
waiting
Assigned
Clear
Reviewer
josh_
laurel
matt_
vince_
Comp Supe
Depth Lead
Comp Lead
Episode
Real
Sequence
Current Latest
Search
Presets

FIGURE 115

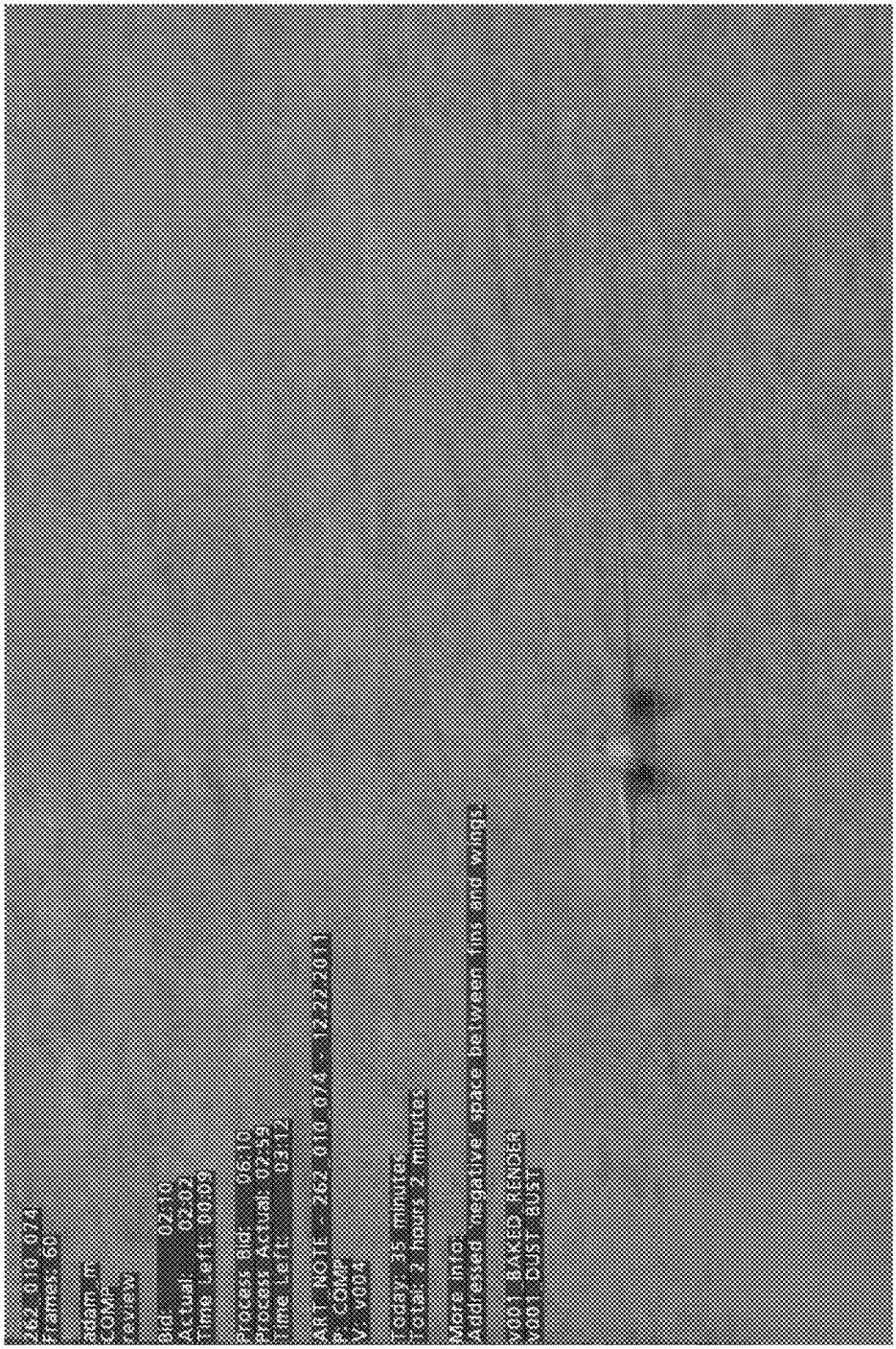


FIGURE 116

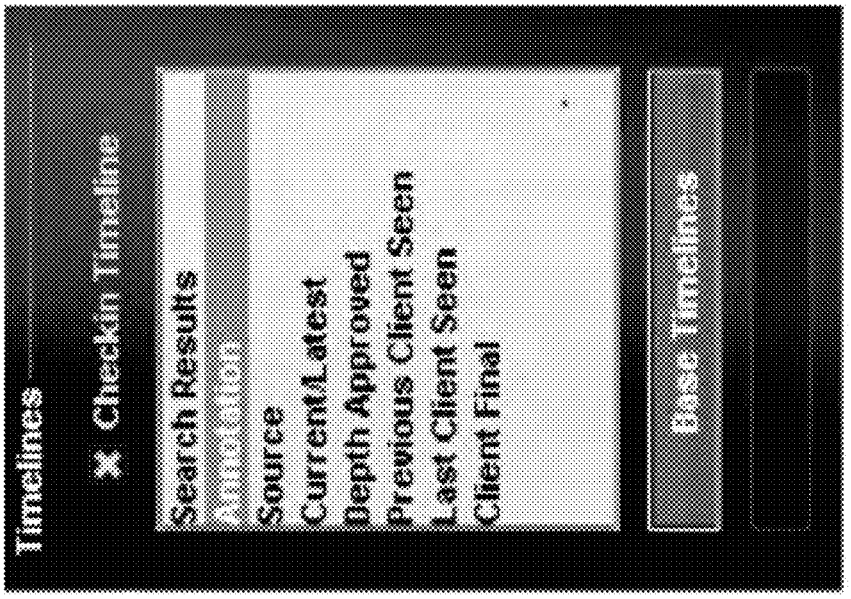

Timelines
Checkin Timeline
Search Results
Source
Current/Latest
Depth Approved
Previous Client Seen
Last Client Seen
Client Final
Base Timelines

FIGURE 117

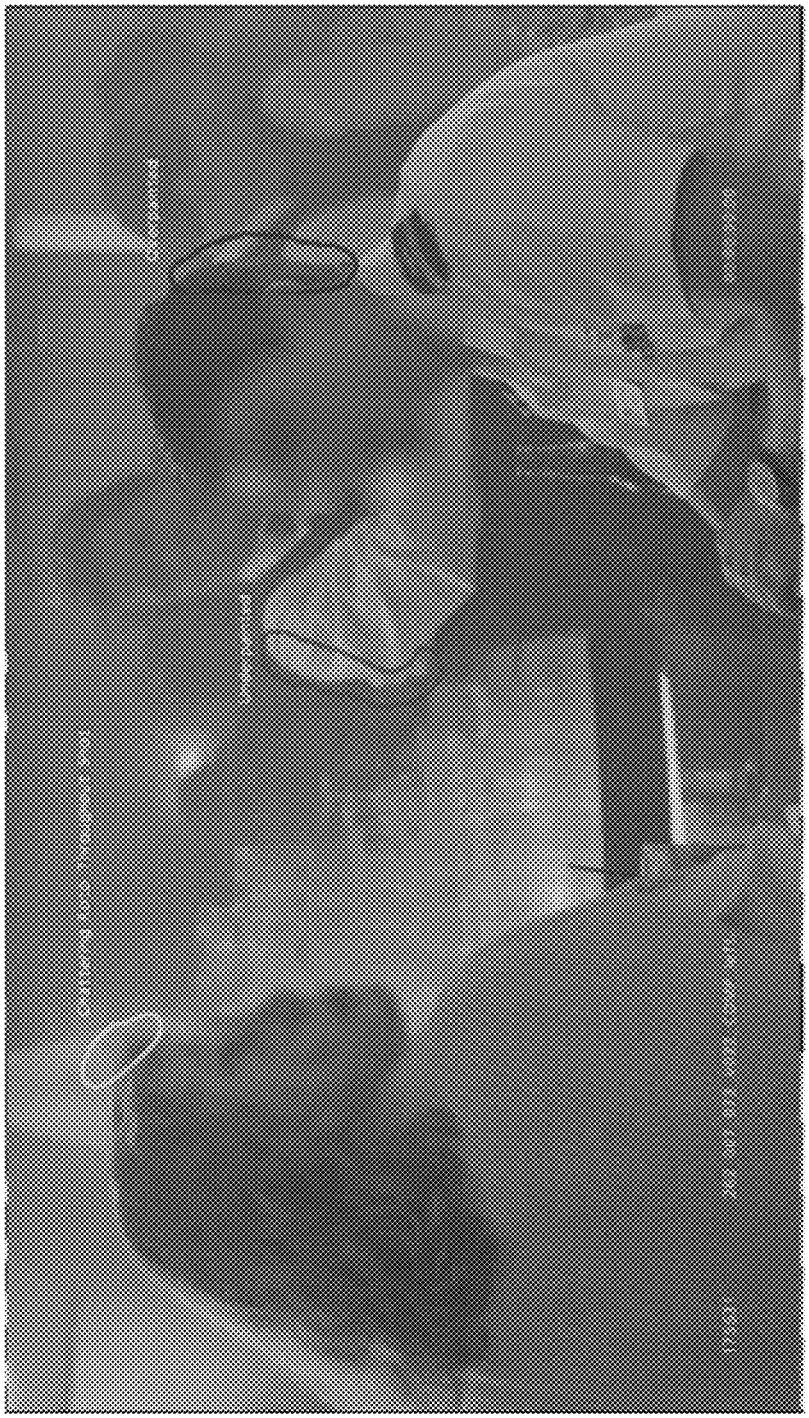


FIGURE 118

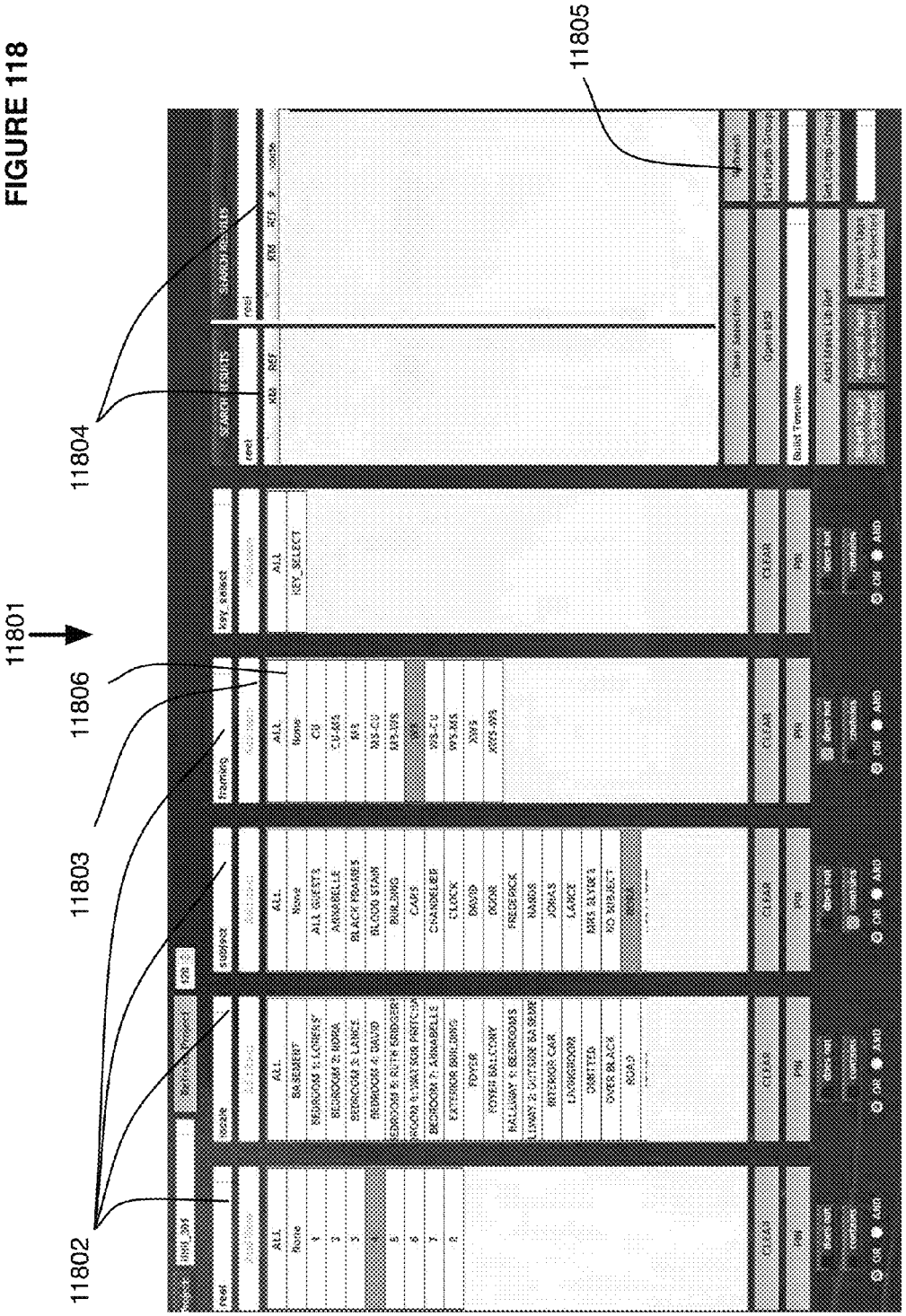


FIGURE 119

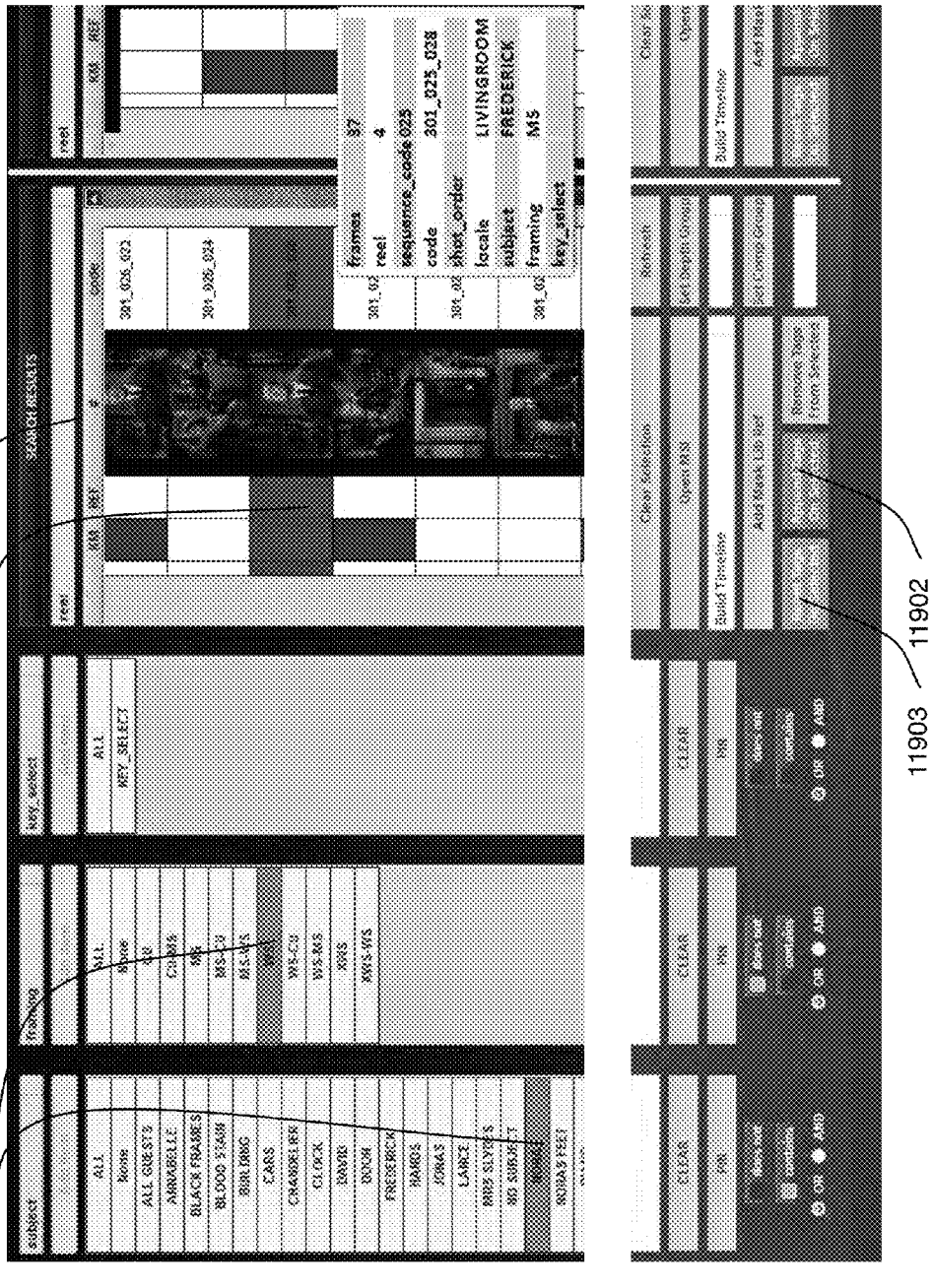


FIGURE 120

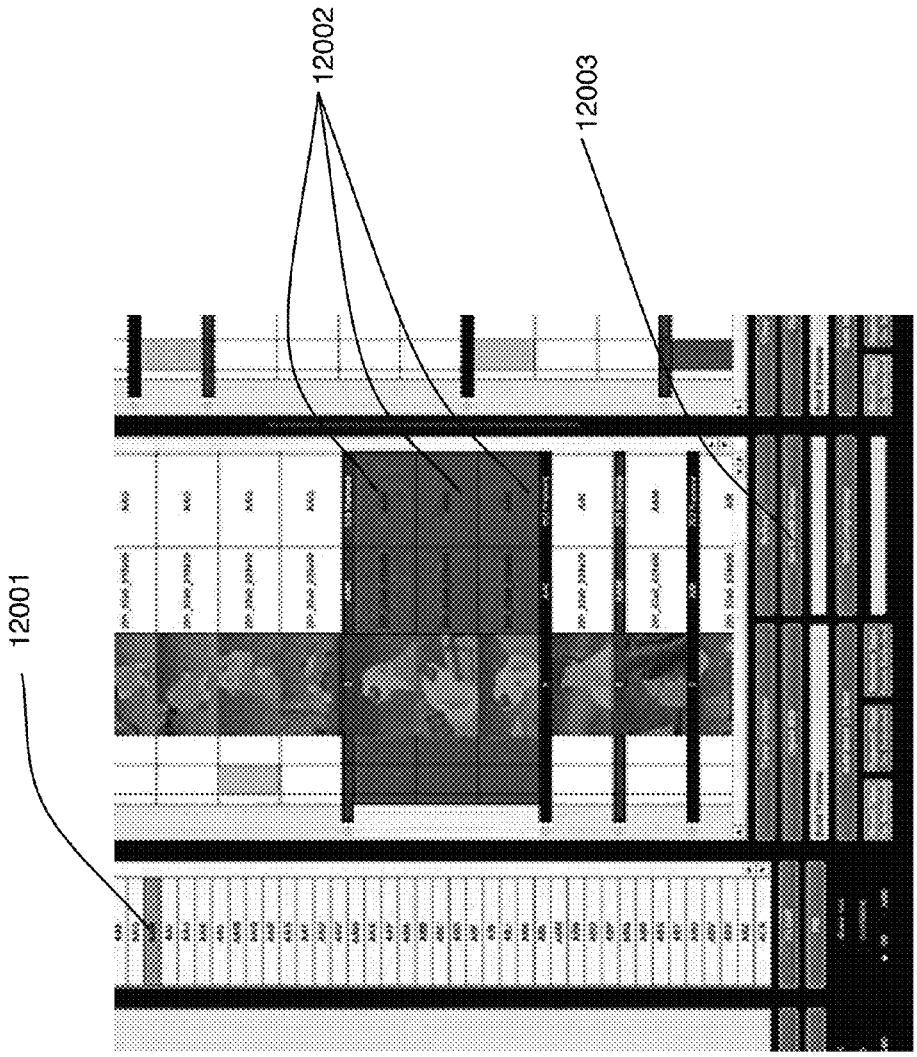


FIGURE 121

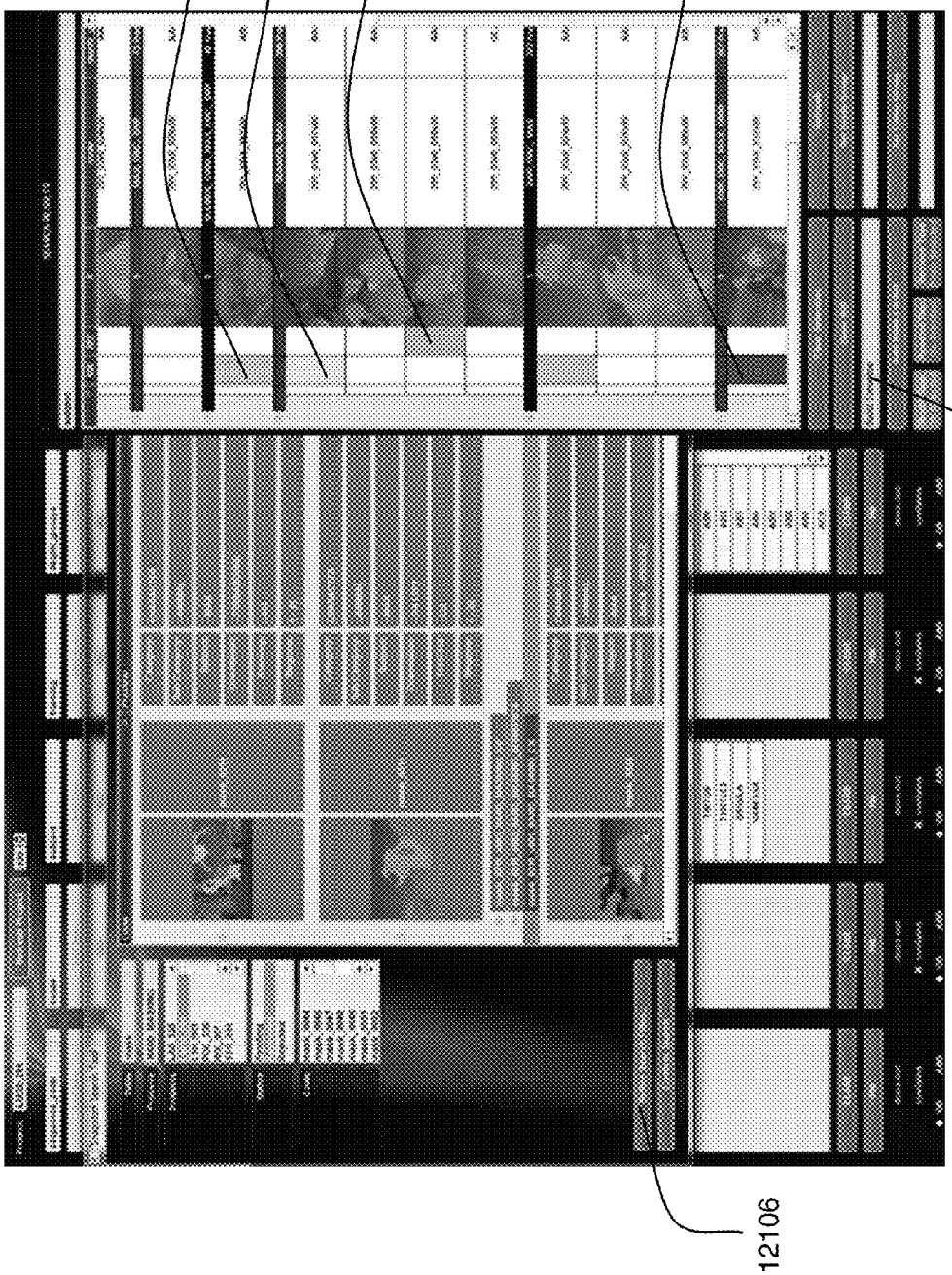


FIGURE 122

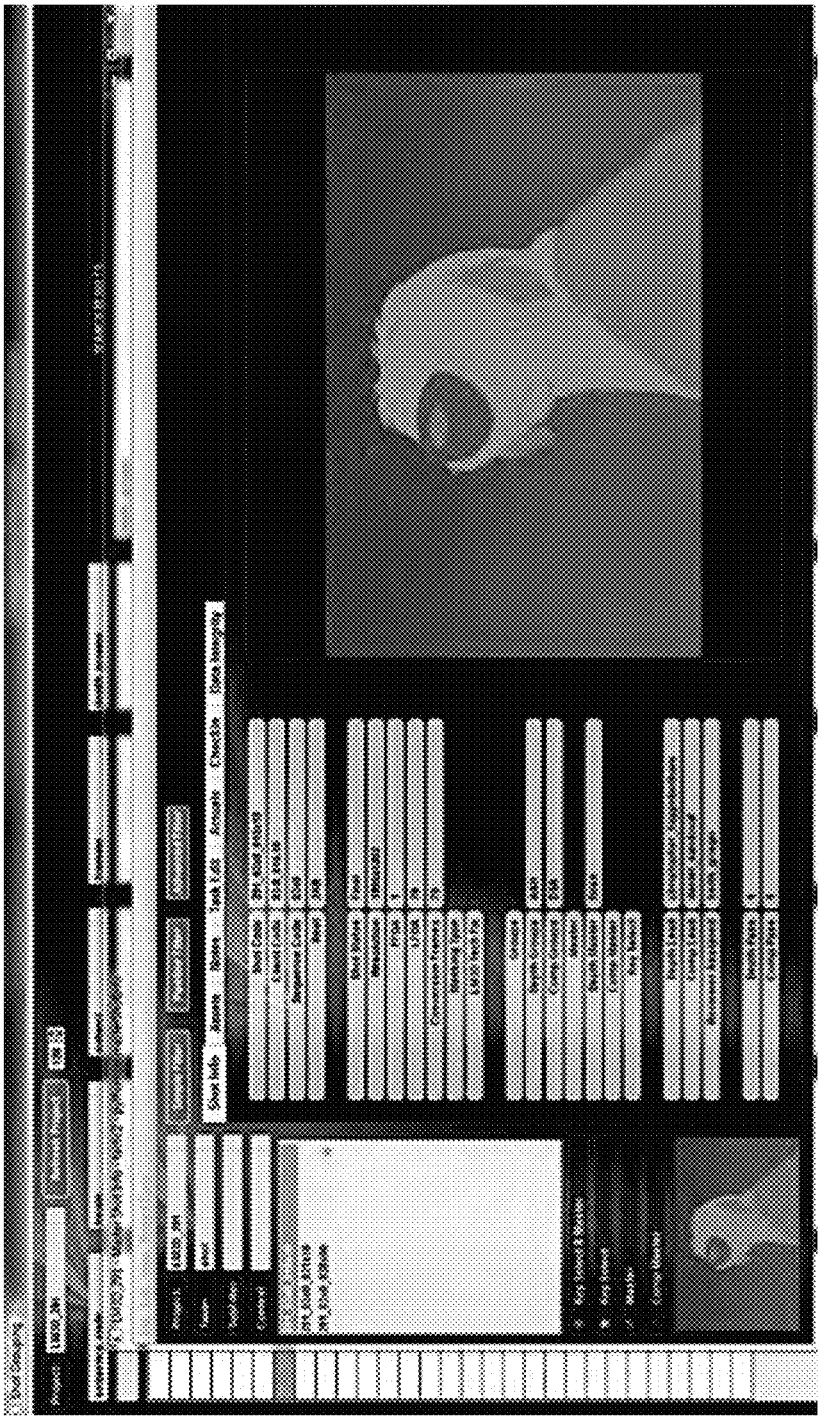


FIGURE 123

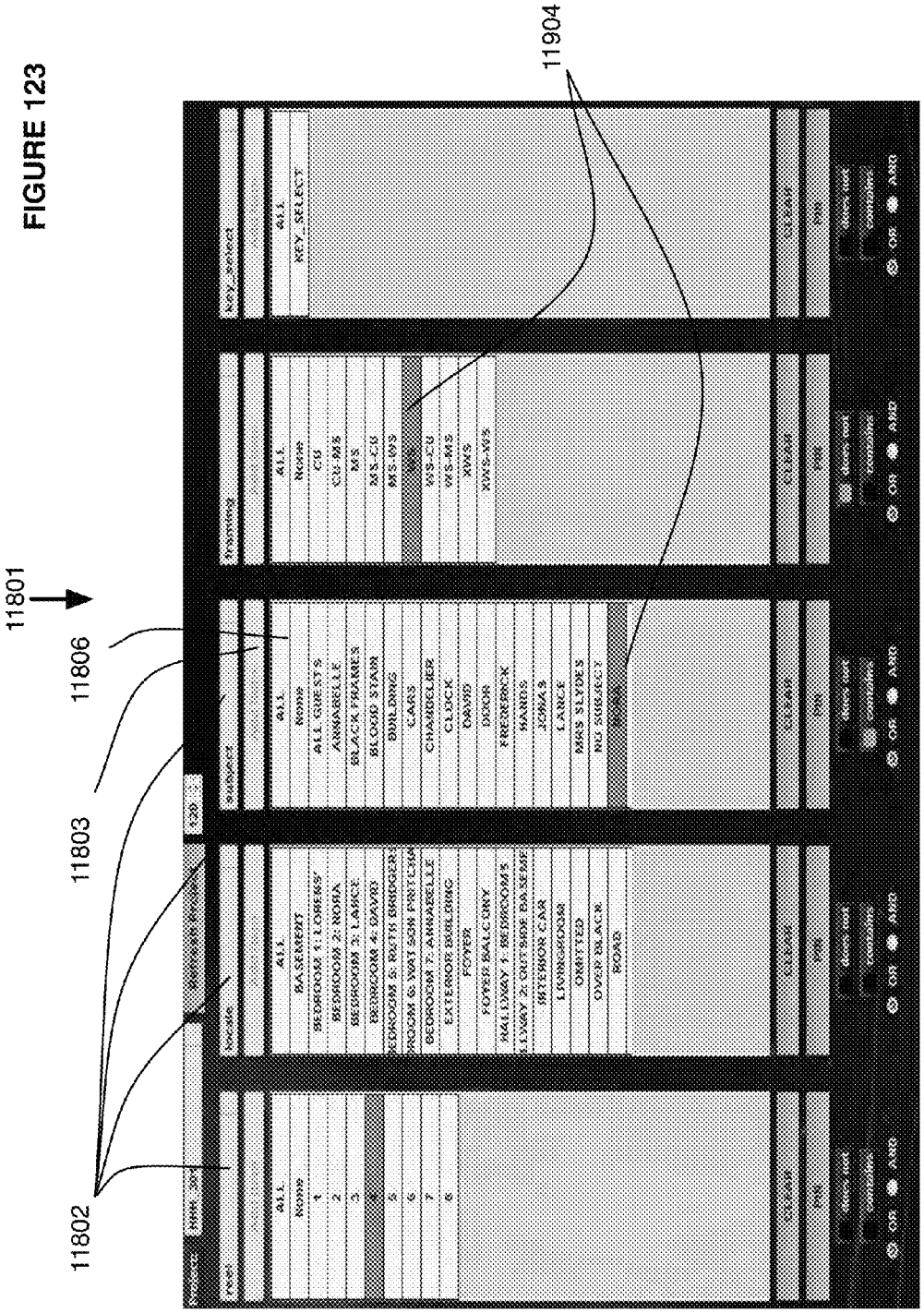


FIGURE 124

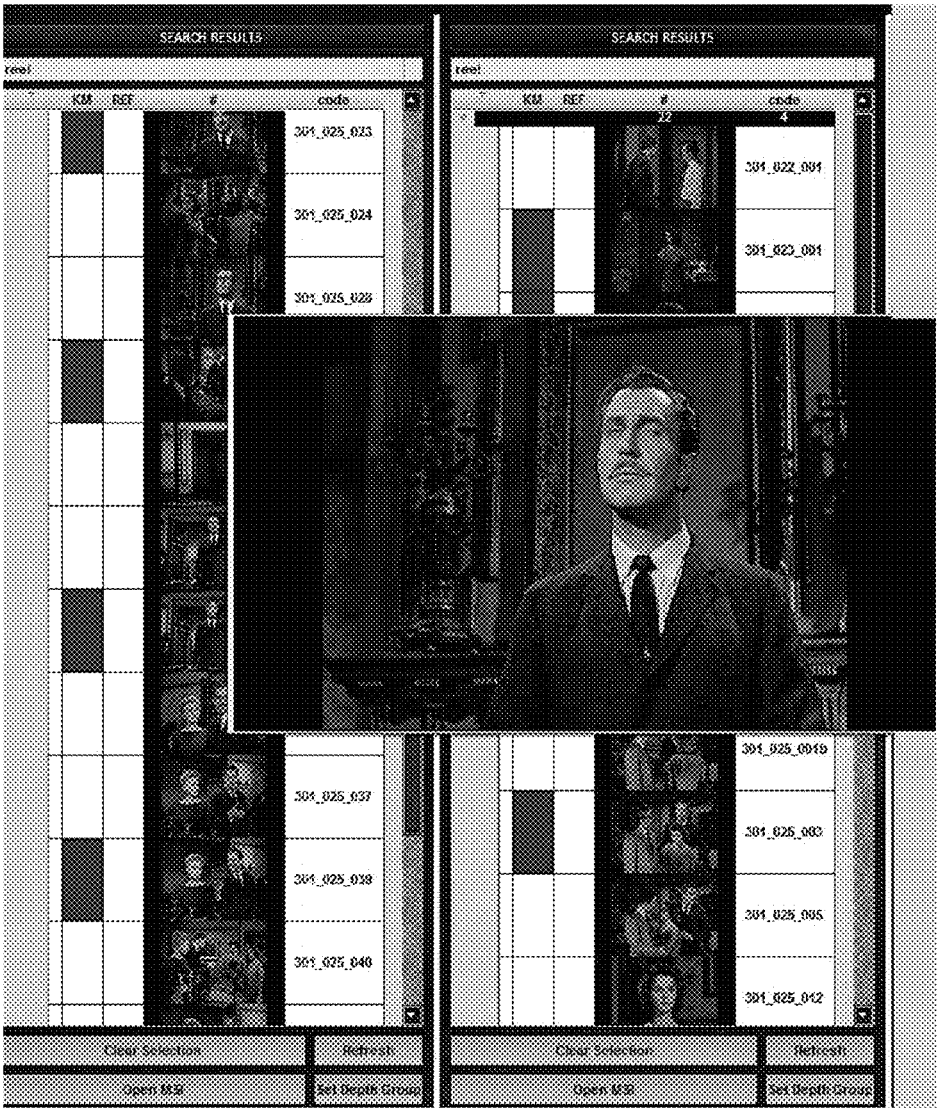


FIGURE 125

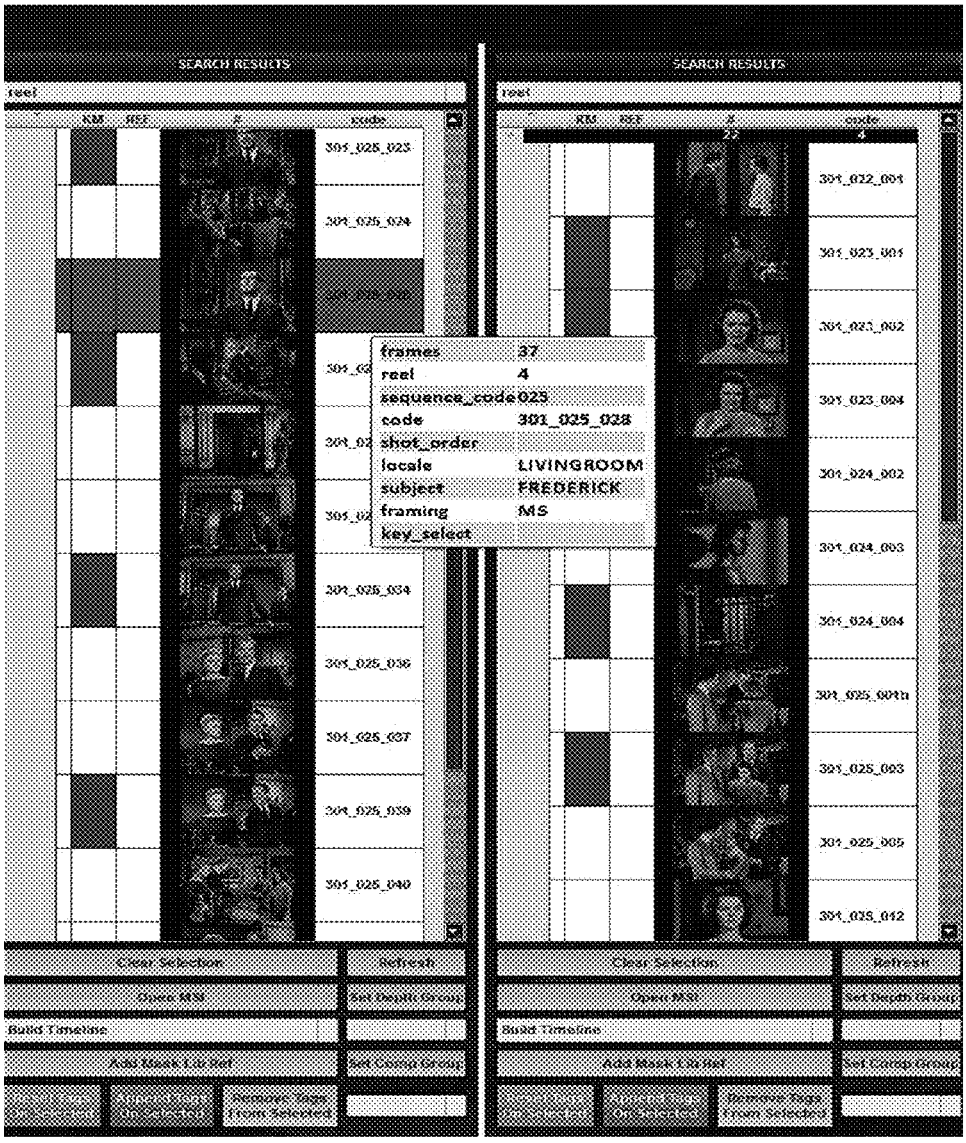
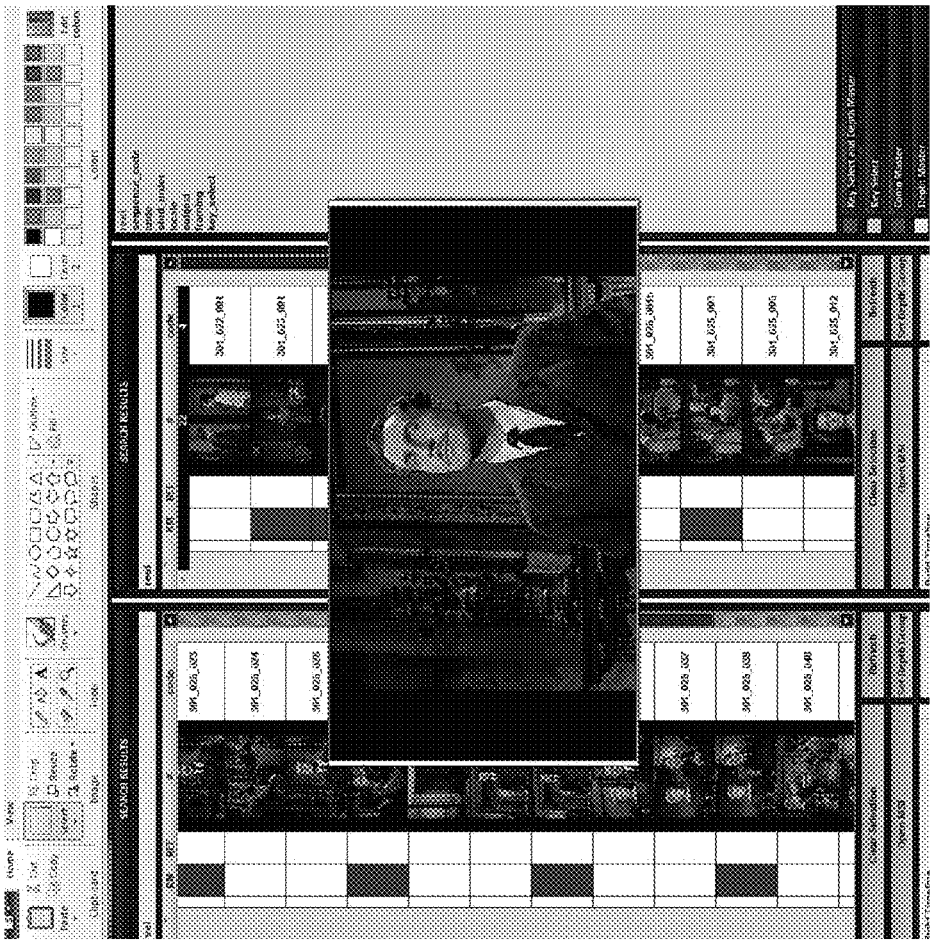

SEARCH RESULTS
reel
KM
REF
Z
code
301_025_023
301_025_024
301_025_034
301_025_036
301_025_037
301_025_039
301_025_040
frames 37
reel 4
sequence_code 025
code 301_025_028
shot_order
locale LIVINGROOM
subject FREDERICK
framing MS
key_select
Clear Selection
Refresh
Open MSI
Set Depth Group
Build Timeline
Add Mask To Set
Set Comp Group
SEARCH RESULTS
reel
KM
REF
Z
code
301_022_001
301_023_001
301_023_002
301_023_004
301_024_002
301_024_003
301_024_004
301_025_003
301_025_005
301_025_012
Clear Selection
Refresh
Open MSI
Set Depth Group
Build Timeline
Add Mask To Set
Set Comp Group

FIGURE 126



MULTI-STAGE PRODUCTION PIPELINE SYSTEM

This application is a continuation of U.S. Utility patent application Ser. No. 13/895,979, filed 16 May 2013, which is a continuation in part of U.S. Utility patent application Ser. No. 13/366,899, filed 6 Feb. 2012, the specifications of which are hereby incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

One or more embodiments of the invention are related to the field of motion picture production and project management in the motion picture industry and relates to reviewers, production managers who manage artists. Production managers are also known as "production" for short. Artists utilize image analysis and image enhancement and computer graphics processing for example to convert two-dimensional images into three-dimensional images associated with a motion picture or otherwise create or alter motion pictures. More particularly, but not by way of limitation, one or more embodiments of the invention enable a multi-stage production pipeline system that may be utilized in conjunction with a motion picture project management system that includes metadata associated with at least one shot or associated with regions within the plurality of images in the at least one shot, or both. The metadata is associated with the at least one shot or regions within the plurality of images in the at least one shot, or both. The system enables a large studio workforce to work non-linearly on a film while maintaining a unified vision driven by key creative figures, allowing for more consistent, higher quality, faster, less expensive work product and more efficient project management techniques. The system also enables reuse of project files, masks and other production elements across projects and efficiently manage projects related to motion pictures to manage assets, control costs, predict budgets and profit margins, reduce archival storage and otherwise provide displays tailored to specific roles to increase worker efficiency.

2. Description of the Related Art

Known methods for the colorizing of black and white feature films involves the identification of gray scale regions within a picture followed by the application of a pre-selected color transform or lookup tables for the gray scale within each region defined by a masking operation covering the extent of each selected region and the subsequent application of said masked regions from one frame to many subsequent frames. The primary difference between U.S. Pat. No. 4,984,072, System And Method For Color Image Enhancement, and U.S. Pat. No. 3,705,762, Method For Converting Black-And-White Films To Color Films, is the manner by which the regions of interest (ROIs) are isolated and masked, how that information is transferred to subsequent frames and how that mask information is modified to conform with changes in the underlying image data. In the U.S. Pat. No. 4,984,072 system, the region is masked by an operator via a one-bit painted overlay and operator manipulated using a digital paintbrush method frame by frame to match the movement. In the U.S. Pat. No. 3,705,762 process, each region is outlined or rotoscoped by an operator using vector polygons, which are then adjusted frame by frame by the operator, to create animated masked ROIs. Various masking technologies are generally also utilized in the conversion of 2D movies to 3D movies.

In both systems described above, the color transform lookup tables and regions selected are applied and modified manually to each frame in succession to compensate for changes in the image data that the operator detects visually. All changes and movement of the underlying luminance gray scale is subjectively detected by the operator and the masks are sequentially corrected manually by the use of an interface device such as a mouse for moving or adjusting mask shapes to compensate for the detected movement. In all cases the underlying gray scale is a passive recipient of the mask containing pre-selected color transforms with all modifications of the mask under operator detection and modification. In these prior inventions the mask information does not contain any information specific to the underlying luminance gray scale and therefore no automatic position and shape correction of the mask to correspond with image feature displacement and distortion from one frame to another is possible.

Existing systems that are utilized to convert two-dimensional images to three-dimensional images may also require the creation of wire frame models for objects in images that define the 3D shape of the masked objects. The creation of wire frame models is a large undertaking in terms of labor. These systems also do not utilize the underlying luminance gray scale of objects in the images to automatically position and correct the shape of the masks of the objects to correspond with image feature displacement and distortion from one frame to another. Hence, great amounts of labor are required to manually shape and reshape masks for applying depth or Z-dimension data to the objects. Motion objects that move from frame to frame thus require a great deal of human intervention. In addition, there are no known solutions for enhancing two-dimensional images into three-dimensional images that utilize composite backgrounds of multiple images in a frame for spreading depth information to background and masked objects. This includes data from background objects whether or not pre-existing or generated for an occluded area where missing data exists, i.e., where motion objects never uncover the background. In other words, known systems gap fill using algorithms for inserting image data where none exists, which causes artifacts.

Current methods for converting movies from 2D to 3D that include computer-generated elements or effects, generally utilize only the final sequence of 2D images that make up the movie. This is the current method used for conversion of all movies from two-dimensional data to left and right image pairs for three-dimensional viewing. There are no known current methods that obtain and make use of metadata associated with the computer-generated elements for a movie to be converted. This is the case since studios that own the older 2D movies may not have retained intermediate data for a movie, i.e., the metadata associated with computer generated elements, since the amount of data in the past was so large that the studios would only retain the final movie data with rendered computer graphics elements and discard the metadata. For movies having associated metadata that has been retained, (i.e., intermediate data associated with the computer-generated elements such as mask, or alpha and/or depth information), use of this metadata would greatly speed the depth conversion process.

In addition, typical methods for converting movies from 2D to 3D in an industrial setting capable of handling the conversion of hundreds of thousands of frames of a movie with large amounts of labor or computing power, make use of an iterative workflow in a linear manner that does not take into account common elements that exist for example if non-adjacent scenes. Since work is generated on a scene basis, the resulting work product is typically non-consistent, for example with colors and/or depths for a given object that appears in different scenes. The iterative workflow includes masking objects in each frame, adding depth and then ren-

dering the frame into left and right viewpoints forming an anaglyph image or a left and right image pair. If there are errors in the edges of the masked objects for example, then the typical workflow involves an "iteration", i.e., sending the frames back to the workgroup responsible for masking the objects, (which can be in a country with cheap unskilled labor half way around the world), after which the masks are sent to the workgroup responsible for rendering the images, (again potentially in another country), after which the rendered image pair is sent back to the quality assurance group. It is not uncommon in this workflow environment for many iterations of a complicated frame to take place. This is known as "throw it over the fence" workflow since different workgroups work independently to minimize their current work load and not as a team with overall efficiency in mind. With hundreds of thousands of frames in a movie, the amount of time that it takes to iterate back through frames containing artifacts can become high, causing delays in the overall project. Even if the re-rendering process takes place locally, the amount of time to re-render or ray-trace all of the images of a scene can cause significant processing and hence delays on the order of at least hours. Elimination of iterations such as this would provide a huge savings in wall-time, or end-to-end time that a conversion project takes, thereby increasing profits and minimizing the workforce needed to implement the workflow.

General simplistic project management concepts are known, however the formal and systematic application of project management in engineering projects of large complexity began in the mid-1900's. Project management in general involves at least planning and managing resources and workers to complete a temporary activity known as a project. Projects are generally time oriented and also constrained by scope and budget. Project management was first described in a systematic manner by Frederick Winslow Taylor and his students Henry Gantt and Henri Fayol. Work breakdown structure and Gantt charts were used initially and Critical Path Method "CPM" and Program Evaluation and Review Technique "PERT" were later developed in industrial and defense settings respectively. Project cost estimating followed these developments. Basic project management generally includes initiation, project planning, execution, monitor/control and completion. More complex project management techniques may attempt to achieve other goals, such as ensuring that the management process is defined, quantitatively managed and optimized for example as is described in the Capability Maturity Model Integration approach.

As described above, industrial based motion picture projects typically include hundreds of thousands of frames, however in addition, these types of projects may also utilize use gigantic amounts of storage including potentially hundreds of layers of masks and images per frame and hundreds of workers. These types of projects have been managed in a fairly ad hoc manner to date in which costs are difficult to predict, controlled feedback to redirect a project toward financial success, asset management and most other best practice project management techniques are minimally utilized. In addition, project management tools utilized include off the shelf project management tools that are not tailored for the specifics of project management in a unique vertical industry such as motion picture effects and conversion projects. Hence, predicting costs and quality and repeatedly performing projects in the film industry has been difficult to accomplish to date. For example, existing motion picture projects sometimes require three people to review an edited frame in some cases, e.g., a person to locate the resource amongst a large number of resources, a person to review the resource and another person to provide annotations for feedback and rework. Although standalone tools exist to perform these tasks, they are generally not integrated and are difficult for personnel in different roles to utilize. In addition, since the work is generally performed in a non-linear manner, there is no re-use of masks or other items in scenes that are not in linear time order and hence, the project costs are higher and resulting work product is inconsistent and of lower quality that would be achieved if the common elements in different scenes could be worked on and managed using metadata associated with the scenes or regions within the scenes.

Regardless of the known techniques, there are no known non-linear workflows for 2D to 3D conversion or special effects projects that enables reuse of masks and other items common to objects that appear in non-linear time sequence. In addition, there are no known optimizations or implementations of project management solutions that take into account the unique requirements of the motion picture industry. Hence there is a need for a motion picture project management system.

BRIEF SUMMARY OF THE INVENTION

Embodiments of the invention generally are directed at a multi-stage production pipeline system that may be utilized in conjunction with a motion picture project management system. One or more embodiments of the invention enables a computer and database to be configured to accept metadata associated with at least one shot or associated with regions within the plurality of images in the at least one shot, or both. The metadata is associated with the at least one shot or regions within the plurality of images in the at least one shot, or both. The system enables a large studio workforce to work non-linearly on a film while maintaining a unified vision driven by key creative figures, allowing for more consistent, higher quality, faster, less expensive work product and more efficient project management techniques. The system also enables reuse of project files, masks and other production elements across projects and greatly improves project management related to the production, processing or conversion of motion pictures. Large motion picture projects generally utilize workers of several roles to process each image that makes up a motion picture, which may number in the hundreds of thousands of image frames. One or more embodiments of the invention further enables a computer and database to be configured to accept the assignment of tasks related to artists, time entries for tasks by artists and review of time and actuals of artists by coordinators, a.k.a., "production" and the review of work product by editorial roles. The system thus enables artists working on shots made up of multiple images to be managed for successful on-budget completion of projects, along with minimization of generally vast storage requirements for motion picture assets and enable prediction of costs for future bidding on projects given quality, ratings of workers to use and schedule.

Tasks involved in a motion picture project generally include tasks related to assessment of a project, ingress of a project, assignment of tasks, performing the assigned tasks or project work, reviewing the work product and archiving and shipping the work product of the project. One or more embodiments of the invention enable workers of different "roles" to view project tasks in a manner consistent with and which aids their role. This is unknown in the motion picture industry. Roles may be utilized in one or more embodiments of the invention that include "editorial", "asset manager", "visual effects supervisor", "coordinator" or "production", "artist", "lead artist", "stereographer", "compositor", "reviewer", "production assistant". In a simpler sense for ease

of illustration, three general categories relate to production workers that manage artists, artist workers who perform the vast majority of work product related work and editorial workers who review and provide feedback based on the work product. Each of these roles may utilize a unique or shared view of the motion picture image frames and/or information related to each image or other asset that their role is assigned to work on.

General Workflow for the Assessment Phase

Generally, the editorial and/or asset manager and/or visual effects supervisor roles utilize a tool that shows the motion picture on a display of a computer. The tool, for example enables the various roles involved in this phase to break a motion picture down into scenes or shots to be worked on. One such tool includes "FRAME CYCLER®" commercially available from ADOBE®.

General Workflow for the Ingest Phase

Generally, the asset manager enters the various scene breaks and other resource such as alpha masks, computer generated element layers or any other resources associated with scenes in the motion picture into a database. Any type of database may be utilized in one or more embodiments of the invention. One such tool that may be utilized to store information related to the motion picture and the assets for project management includes the project management database "TACTIC™", which is commercially available from SOUTHPAW TECHNOLOGY Inc™ Any database may be utilized in one or more embodiments of the invention so long as the motion picture specific features are included in the project management database. One or more embodiments of the invention update the "snapshot" and "file" tables in the project management database. The schema of the project management database is briefly described in this section and described in more detail in the detailed description section below.

General Workflow for the Assignment Phase

Generally, production workers utilize an interface that couples with project management database to assign particular workers to particular tasks associated with their role and assign the workers images associated with shots or scenes in a given motion picture. One or more embodiments of the invention make use of basic project management database digital asset management tables and add additional fields that improve upon basic project management functionality to optimize the project management process for the motion picture industry. One or more embodiments of the invention update the "task" table in the project management database.

General Workflow for the Project Work Phase

Generally, artists, stereographers and compositors perform a large portion of the total work on a motion picture. These roles generally utilize a time clock tool to obtain their tasks and set task status and start and stop times for the task. Generally, artists perform mask and region design and initial depth augmentation of a frame. The artists generally utilize a ray tracing program that may include automated mask tracking capabilities for example, along with NUKE™ commercially available from "THE FOUNDRY™, for mask cleanup for example. Once a client approves the visual effects and/or depth work on a scene, then compositors finish the scene with the same tools that the artists use and generally with other tools such as AFTER EFFECTS® and PHOTOSHOP®, commercially available from ADOBE®. In one or more embodiments of the invention, the person who worked on a particular asset is stored in the project management database in custom fields for example.

In specific workflow scenarios, workers in region design for example classify elements in scenes into two separate categories. Scenes generally include two or more images in time sequence for example. The two categories include background elements (i.e. sets and foreground elements that are stationary) or motion elements (e.g., actors, automobiles, etc.) that move throughout the scene. These background elements and motion elements are treated separately in embodiments of the invention similar to the manner in which traditional animation is produced. In addition, many movies now include computer-generated elements (also known as computer graphics or CG, or also as computer-generated imagery or CGI) that include objects that do not exist in reality, such as robots or spaceships for example, or which are added as effects to movies, for example dust, fog, clouds, etc. Computer-generated elements may include background elements, or motion elements.

Motion Elements: The motion elements are displayed as a series of sequential tiled frame sets or thumbnail images complete with background elements. The motion elements are masked in a key frame using a multitude of operator interface tools common to paint systems as well as unique tools such as relative bimodal thresholding in which masks are applied selectively to contiguous light or dark areas bifurcated by a cursor brush. After the key frame is fully designed and masked, the mask information from the key frame is then applied to all frames in the display-using mask fitting techniques that include:

1. Automatic mask fitting using Fast Fourier Transform and Gradient Decent Calculations based on luminance and pattern matching which references the same masked area of the key frame followed by all prior subsequent frames in succession. Since the computer system implementing embodiments of the invention can reshape at least the outlines of masks from frame to frame, large amounts of labor can be saved from this process that traditionally has been done by hand. In 2D to 3D conversion projects, sub-masks can be adjusted manually within a region of interest when a human recognizable object rotates for example, and this process can be "tweened" such that the computer system automatically adjusts sub-masks from frame to frame between key frames to save additional labor.

2. Bezier curve animation with edge detection as an automatic animation guide

3. Polygon animation with edge detection as an automatic animation guide

In one or more embodiments of the invention, computer-generated elements are imported using RGBAZ files that include an optional alpha mask and/or depths on a pixel-by-pixel, or sub-pixel-by-sub-pixel basis for a computer-generated element. Examples of this type of file include the EXR file format. Any other file format capable of importing depth and/or alpha information is in keeping with the spirit of the invention. Embodiments of the invention import any type of file associated with a computer-generated element to provide instant depth values for a portion of an image associated with a computer-generated element. In this manner, no mask fitting or reshaping is required for any of the computer-generated elements from frame to frame since the alpha and depth on a pixel-by-pixel or sub-pixel-by-sub-pixel basis already exists, or is otherwise imported or obtained for the computer-generated element. For complicated movies with large amounts of computer-generated elements, the import and use of alpha and depth for computer-generated elements makes the conversion of a two-dimensional image to a pair of images for right and left eye viewing economically viable. One or more embodiments of the invention allow for the background elements and motion elements to have depths associated with them or otherwise set or adjusted, so that all objects other than

computer-generated objects are artistically depth adjusted. In addition, embodiments of the invention allow for the translation, scaling or normalization of the depths for example imported from an RGBAZ file that are associated with computer-generated objects so as to maintain the relative integrity of depth for all of the elements in a frame or sequence of frames. In addition, any other metadata such as character mattes or alphas or other masks that exist for elements of the images that make up a movie can also be imported and utilized to improve the operated-defined masks used for conversion. On format of a file that may be imported to obtain metadata for photographic elements in a scene includes the RGBA file format. By layering different objects from deepest to closest, i.e., "stacking" and applying any alpha or mask of each element, and translating the closest objects the most horizontally for left and right images, a final pair of depth enhanced images is thus created based on the input image and any computer-generated element metadata.

In another embodiment of this invention, these background elements and motion elements are combined separately into single frame representations of multiple frames, as tiled frame sets or as a single frame composite of all elements (i.e., including both motion and backgrounds/foregrounds) that then becomes a visual reference database for the computer controlled application of masks within a sequence composed of a multiplicity of frames. Each pixel address within the reference visual database corresponds to mask/lookup table address within the digital frame and X, Y, Z location of subsequent "raw" frames that were used to create the reference visual database. Masks are applied to subsequent frames based on various differentiating image processing methods such as edge detection combined with pattern recognition and other sub-mask analysis, aided by operator segmented regions of interest from reference objects or frames, and operator directed detection of subsequent regions corresponding to the original region of interest. In this manner, the gray scale actively determines the location and shape of each mask (and corresponding color lookup from frame to frame for colorization projects or depth information for two-dimensional to three-dimensional conversion projects) that is applied in a keying fashion within predetermined and operator-controlled regions of interest.

Camera Pan Background and Static Foreground Elements: Stationary foreground and background elements in a plurality of sequential images comprising a camera pan are combined and fitted together using a series of phase correlation, image fitting and focal length estimation techniques to create a composite single frame that represents the series of images used in its construction. During the process of this construction the motion elements are removed through operator adjusted global placement of overlapping sequential frames.

For colorization projects, the single background image representing the series of camera pan images is color designed using multiple color transform look up tables limited only by the number of pixels in the display. This allows the designer to include as much detail as desired including air brushing of mask information and other mask application techniques that provide maximum creative expression. For depth conversion projects, (i.e., two-dimensional to three-dimensional movie conversion for example), the single background image representing the series of camera pan images may be utilized to set depths of the various items in the background. Once the background color/depth design is completed the mask information is transferred automatically to all the frames that were used to create the single composited image. In this manner, color or depth is performed once per multiple images and/or scene instead of once per frame, with color/depth information automatically spread to individual frames via embodiments of the invention. Masks from colorization projects may be combined or grouped for depth conversion projects since the colorization masks may contain more sub-areas than a depth conversion mask. For example, for a coloration project, a person's face may have several masks applied to areas such as lips, eyes, hair, while a depth conversion project may only require an outline of the person's head or an outline of a person's nose, or a few geometric shape sub-masks to which to apply depth. Masks from a colorization project can be utilized as a starting point for a depth conversion project since defining the outlines of human recognizable objects by itself is time consuming and can be utilized to start the depth conversion masking process to save time. Any computer-generated elements at the background level may be applied to the single background image.

In one or more embodiments of the invention, image offset information relative to each frame is registered in a text file during the creation of the single composite image representing the pan and used to apply the single composite mask to frames used to create the composite image.

Since the foreground moving elements have been masked separately prior to the application of the background mask, the background mask information is applied wherever there is no pre-existing mask information.

Static Camera Scenes With and Without Film Weave, Minor Camera Following and Camera Drift: In scenes where there is minor camera movement or film weave resulting from the sprocket transfer from 35 mm or 16 mm film to digital format, the motion objects are first fully masked using the techniques listed above. All frames in the scene are then processed automatically to create a single image that represents both the static foreground elements and background elements, eliminating all masked moving objects where they both occlude and expose the background.

Wherever the masked moving object exposes the background or foreground, the instance of background and foreground previously occluded is copied into the single image with priority and proper offsets to compensate for camera movement. The offset information is included in a text file associated with each single representation of the background so that the resulting mask information can be applied to each frame in the scene with proper mask offsets.

The single background image representing the series of static camera frames is color designed using multiple color transform look up tables limited only by the number of pixels in the display. Where the motion elements occlude the background elements continuously within the series of sequential frames they are seen as black figure that are ignored and masked over. The black objects are ignored in colorization-only projects during the masking operation because the resulting background mask is later applied to all frames used to create the single representation of the background only where there is no pre-existing mask. If background information is created for areas that are never exposed, then this data is treated as any other background data that is spread through a series of images based on the composite background. This allows for minimization of artifacts or artifact-free two-dimensional to three-dimensional conversion since there is never any need to stretch objects or extend pixels as for missing data, since image data that has been generated to be believable to the human observer is generated for and then taken from the occluded areas when needed during the depth conversion process. Hence for motion elements and computer-generated elements, realistic looking data can be utilized for areas behind these elements when none exists. This allows the designer to include as much detail as desired

including air brushing of mask information and other mask application techniques that provide maximum creative expression. Once the background color design is completed the mask information is transferred automatically to all the frames that were used to create the single composited image. For depth projects, the distance from the camera to each item in the composite frame is automatically transferred to all the frames that were used to create the single composited image. By shifting masked background objects horizontally more or less, their perceived depth is thus set in a secondary viewpoint frame that corresponds to each frame in the scene. This horizontal shifting may utilize data generated by an artist for the occluded or alternatively, areas where no image data exists yet for a second viewpoint may be marked in one or more embodiments of the invention using a user defined color that allows for the creation missing data to ensure that no artifacts occur during the two-dimension to three-dimension conversion process. Any technique known may be utilized in embodiments of the invention to cover areas in the background where unknown data exists, i.e., (as displayed in some color that shows where the missing data exists) that may not be borrowed from another scene/frame for example by having artists create complete backgrounds or smaller occluded areas with artist drawn objects. After assigning depths to objects in the composite background, or by importing depths associated with computer-generated elements at the background depth, a second viewpoint image may be created for each image in a scene in order to produce a stereoscopic view of the movie, for example a left eye view where the original frames in the scene are assigned to the right eye viewpoint, for example by translating foreground objects horizontally for the second viewpoint, or alternatively by translating foreground objects horizontally left and right to create two viewpoints offset from the original viewpoint.

One or more tools employed by the system enable real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts and to minimize or eliminate iterative workflow paths back through different workgroups by generating translation files that can be utilized as portable pixel-wise editing files. For example, a mask group takes source images and creates masks for items, areas or human recognizable objects in each frame of a sequence of images that make up a movie. The depth augmentation group applies depths, and for example shapes, to the masks created by the mask group. When rendering an image pair, left and right viewpoint images and left and right translation files may be generated by one or more embodiments of the invention. The left and right viewpoint images allow 3D viewing of the original 2D image. The translation files specify the pixel offsets for each source pixel in the original 2D image, for example in the form of UV or U maps. These files are generally related to an alpha mask for each layer, for example a layer for an actress, a layer for a door, a layer for a background, etc. These translation files, or maps are passed from the depth augmentation group that renders 3D images, to the quality assurance workgroup. This allows the quality assurance workgroup (or other workgroup such as the depth augmentation group) to perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts such as masking errors without delays associated with processing time/re-rendering and/or iterative workflow that requires such re-rendering or sending the masks back to the mask group for rework, wherein the mask group may be in a third world country with unskilled labor on the other side of the globe. In addition, when rendering the left and right images, i.e., 3D images, the Z depth of regions within the image, such as actors for example, may also be passed along with the alpha mask to the quality assurance group, who may then adjust depth as well without re-rendering with the original rendering software. This may be performed for example with generated missing background data from any layer so as to allow "downstream" real-time editing without re-rendering or ray-tracing for example. Quality assurance may give feedback to the masking group or depth augmentation group for individuals so that these individuals may be instructed to produce work product as desired for the given project, without waiting for, or requiring the upstream groups to rework anything for the current project. This allows for feedback yet eliminates iterative delays involved with sending work product back for rework and the associated delay for waiting for the reworked work product. Elimination of iterations such as this provide a huge savings in wall-time, or end-to-end time that a conversion project takes, thereby increasing profits and minimizing the workforce needed to implement the workflow.

General Workflow for the Review Phase

Regardless of the type of project work performed on a given asset, the asset is reviewed for example using an interface that couples with the project management database to enable the viewing of work product. Generally, editorial role based users use the interface most, artists and stereographers less and lead artists the least. The review notes and images may be viewed simultaneously, for example with a clear background surrounding text that is overlaid on the image or scene to enable rapid review and feedback by a given worker having a particular role. Other improvements to the project management database include ratings or artists and difficulty of the asset. These fields enable workers to be rated and projected costs to be forecast when bidding projects, which is unknown in the field of motion picture project planning.

General Workflow for the Archive and Shipping Phase

Asset managers may delete and/or compress all assets that may be regenerated, which can save hundreds of terabytes of disk space for a typical motion picture. This enables an enormous savings in disk drive hardware purchases and is unknown in the art.

One or more embodiments of the system may be implemented with a computer and a database coupled with the computer. Any computer architecture having any number of computers, for example coupled via a computer communication network is in keeping with the spirit of the invention. The database coupled with the computer includes at least a project table, shot table, task table and timesheet table. The project table generally includes project identifier and description of a project related to a motion picture. The shot table generally includes a shot identifier and references a plurality of images with a starting frame value and an ending frame value wherein the plurality of images are associated with the motion picture that is associated with the project. The shot table generally includes at least one shot having status related to progress of work performed on the shot. The task table generally references the project using a project identifier in also located in the project table. The task table generally includes at least one task which generally includes a task identifier and an assigned worker, e.g., artist, and which may also include a context setting associated with a type of task related to motion picture work selected from region design, setup, motion, composite, and review. The at least one task generally includes a time allocated to complete the at least one task. The timesheet item table generally references the project identifier in the project table and the task identifier in the task table. The task table generally includes at least one timesheet item that includes a start time and an end time. In one or more embodiments of the invention, the computer is configured to present a first display

configured to be viewed by an artist that includes at least one daily assignment having a context, project, shot and a status input that is configured to update the status in the task table and a timer input that is configured to update the start time and the end time in the timesheet item table. The computer is generally configured to present a second display configured to be viewed by a coordinator or "production" worker, i.e., production that includes a search display having a context, project, shot, status and artist and wherein the second display further includes a list of a plurality of artists and respective status and actuals based on time spent in the at least one timesheet item versus the time allocated per the at least one task associated with the at least one shot. The computer is generally also configured to present a third display configured to be viewed by an editor that includes an annotation frame configured to accept commentary or drawing or both commentary and drawing on the at least one of said plurality of images associated with the at least one shot. One or more embodiments of the computer may be configured to provide the third display configured to be viewed by an editor that includes an annotation overlaid on at least one of the plurality of images. This capability provides information on one display that has generally required three workers to integrate in known systems, and which is novel in and of itself.

Embodiments of the database may also include a snapshot table which includes an snapshot identifier and search type and which includes a snapshot of the at least one shot, for example that includes a subset of the at least one shot wherein the snapshot is cached on the computer to reduce access to the shot table. Embodiments may also include other context settings for other types of task categories, for example source and cleanup related tasks. Any other context settings or values that are related to motion picture work may also be included in keeping with the spirit of the invention. Embodiments of the database may also include an asset request table that includes an asset request identifier and shot identifier that may be utilized to request work on assets or assets themselves to be worked on or created by other workers for example. Embodiments of the database may also include a request table that includes an mask request identifier and shot identifier and that may be utilized to request any type of action by another worker for example. Embodiments of the database may also include a note table which includes a note identifier and that references the project identifier and that includes at least one note related to at least one of the plurality of images from the motion picture. Embodiments of the database may also include a delivery table that includes a delivery identifier that references the project identifier and which includes information related to delivery of the motion picture.

One or more embodiments of the computer are configured to accept a rating input from production or the editor based on work performed by the artist, optionally in a blind manner in which the reviewer does not know the identity of the artist in order to prevent favoritism for example. One or more embodiments of the computer are configured to accept a difficulty of the at least one shot and calculate a rating based on work performed by the artist and based on the difficulty of the shot and time spent on the shot. One or more embodiments of the computer are configured to accept a rating input from production or editorial, (i.e., an editor worker) based on work performed by the artist, or, accept a difficulty of the at least one shot and calculate a rating based on work performed by the artist and based on the difficulty of the shot and time spent on the shot, and, signify an incentive with respect to the artist based on the rating accepted by the computer or calculated by the computer. One or more embodiments of the computer are configured to estimate remaining cost based on the actuals that are based on total time spent for all of the at least one tasks associated with all of the at least one shot in the project versus time allocated for all of the at least one tasks associated with all of the at least one shot in the project. One or more embodiments of the computer are configured to compare the actuals associated with a first project with actuals associated with a second project and signify at least one worker to be assigned from the first project to the second project based on at least one rating of the first worker that is assigned to the first project. One or more embodiments of the computer are configured to analyze a prospective project having a number of shots and estimated difficulty per shot and based on actuals associated with the project, calculate a predicted cost for the prospective project. One or more embodiments of the computer are configured to analyze a prospective project having a number of shots and estimated difficulty per shot and based on the actuals associated with a first previously performed project and a second previously performed project that completed after the first previously performed project, calculate a derivate of the actuals, calculate a predicted cost for the prospective project based on the derivative of the actuals. For example, as the process improves, tools improve and workers improve, the efficiency of work improves and the budgeting and bid processes can take this into account by calculating how efficiency is changing versus time and use this rate of change to predict costs for a prospective project. One or more embodiments of the computer are configured to analyze the actuals associated with said project and divide completed shots by total shots associated with said project and present a time of completion of the project. One or more embodiments of the computer are configured to analyze the actuals associated with the project and divide completed shots by total shots associated with the project, present a time of completion of the project, accept an input of at least one additional artist having a rating, accept a number of shots in which to use the additional artist, calculate a time savings based on the at least one additional artist and the number of shots, subtract the time savings from the time of completion of the project and present an updated time of completion of the project. One or more embodiments of the computer are configured to calculate amount of disk space that may be utilized to archive the project and signify at least one asset that may be rebuilt from other assets to avoid archival of the at least one asset. One or more embodiments of the computer are configured to display an error message if the artist is working with a frame number that is not current in the at least one shot. This may occur when fades, dissolves or other effects lengthen a particular shot for example wherein the shot contains frames not in the original source assets.

Metadata Grouping Tool

In one or more embodiments of the invention, the motion picture project management system includes a multi-stage production pipeline system for the motion picture projects. According to at least one embodiment as previously stated, the multi-stage production pipeline system includes a computer and a database that includes a shot table with a shot identifier associated with a plurality of images that are ordered in time and that make up a shot such that the shot table has a starting frame value and an ending frame value associated with each shot. In at least one embodiment, the plurality of images are associated with a motion picture and wherein the database further includes metadata associated with at least one shot or associated with regions within the plurality of images in the at least one shot, or both. In at least one embodiment of the invention, the multi-stage production pipeline

system includes a project table, such that the project table includes a project identifier and description of a project related to the motion picture.

In one or more embodiments, the computer may present a grouping tool interface coupled with the computer and the database, wherein the grouping tool may present user interface elements, accept input of the metadata and accept selected shots associated with the metadata via the user interface elements. In at least one embodiment of the invention, the computer may one or more of store the metadata associated with the selected shots in the shot table, accept selected metadata to search the at least one shot, query the shot table with the selected metadata associated with the at least one shot or the regions within the plurality of images in the at least one shot or both, and display a list of shots having the selected metadata. In one or more embodiments, the list of shots may include at least one shot that is non-sequential in time in the motion picture with respect to another shot in the list of shots. The computer may assign work tasks based on the list of shots wherein the list of shots includes the at least one shot that is non-sequential in time in the motion picture with respect to another shot in the list of shots.

In one or more embodiments, the computer may present a first display to be viewed by a production worker that includes a search display with one or more of a context, a project, a shot, the list of shots, a status and an artist, present a second display to be viewed by an artist that includes at least one daily assignment having a context, project and shot or the list of shots or both, and present a third display to be viewed by an editorial worker that includes an annotation frame to accept commentary or drawing or both commentary and drawing on at least one of the plurality of images associated with the at least one shot or the list of shots or both. In one or more embodiments of the invention, the at least one shot or the list of shots, or both, include status related to progress of work performed.

By way of one or more embodiments, the metadata associated with the at least one shot is associated with at least one metadata category. In one or more embodiments, the metadata category comprises one or more of a locale or location at which the shot was obtained, a subject that appears in the shot wherein the subject is a person, place or thing, a shot framing associated with the shot wherein the shot framing includes one or more of a close up or CU, a mid shot or MS, wide shot or WS, and an extreme wide shot or XWS. In at least one embodiment, the grouping tool interface and the metadata categories may transition shot framing including any combination of CU, MS, SW and XWS, such that a CU may transition to a MS, WS or XWS, and/or a MS may transition to a CU, WS or XWS, and/or a WS may transition to a CU, MS or XWS, and/or a XWS may transition to a CU, MS or WS.

In one or more embodiments, the metadata associated with the at least one shot is associated with a metadata category, wherein the metadata category includes one or more of a depth complexity associated with the shot and a clean plate complexity associated with the shot. In at least one embodiment of the invention, the grouping tool interface may accept at least one additional metadata category and additional metadata values associated the metadata category.

By way one or more embodiments, the grouping tool interface may accept an input to designate the at least one shot as a master shot associated with depth, key selects or clean plate or any combination thereof. This enables the at least one shot to be utilized as a benchmark for quality or volume or to improve efficiency or both.

Embodiments enable a large studio workforce to work non-linearly on a film while maintaining a unified vision driven by key creative figures. Thus, work product is more consistent, higher quality, faster, less expensive and enables reuse of project files, masks and other production elements across projects since work is no longer constrained by shot order when using embodiments of the invention.

In at least one embodiment of the invention, the grouping tool interface includes a reference mask library tool with a plurality of reference masks, wherein each of the list of shots share selected metadata. In one or more embodiments, the reference mask library tool may present an interface to accept a selection of one or more of the plurality of reference masks to be utilized in shots in the list of shots that do not already utilize the reference mask associated with the subject. This may enable a worker to locate similar subjects in one or more additional shots and use reference masks associated with the subject located to add metadata to the one or more additional shots, without the need to reinvent metadata for each of those additional shots with the same subject such as a person, place or thing. In one or more embodiments, each one of the plurality of reference masks may be a dedicated template of the subject.

In one or more embodiments, the plurality of reference masks may be obtained from a second project associated with a second motion picture that differs from the first motion picture. This allows the system to create different sequels and/or motions pictures with similar subjects, such as people, places or things, from the first motion picture, with similar metadata to enable time efficiency, consistency and accuracy.

By way of one or more embodiments, the grouping tool interface may display a plurality of search results including the lists of shots associated with a plurality of respective selected metadata. In addition, in at least one embodiment of the invention, the grouping tool interface may present a timeline of the plurality of images associated with the list of shots.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a plurality of feature film or television film frames representing a scene or cut in which there is a single instance or perceptive of a background.

FIG. **2** shows an isolated background processed scene from the plurality of frames shown in FIG. **1** in which all motion elements are removed using various subtraction and differencing techniques. The single background image is then used to create a background mask overlay representing designer selected color lookup tables in which dynamic pixel colors automatically compensate or adjust for moving shadows and other changes in luminance.

FIG. **3** shows a representative sample of each motion object (M-Object) in the scene receives a mask overlay that represents designer selected color lookup tables in which dynamic pixel colors automatically compensate or adjust for moving shadows and other changes in luminance as the M-Object moves within the scene.

FIG. **4** shows all mask elements of the scene are then rendered to create a fully colored frame in which M-Object masks are applied to each appropriate frame in the scene followed by the background mask, which is applied only where there is no pre-existing mask in a Boolean manner.

FIGS. **5**A and **5**B show a series of sequential frames loaded into display memory in which one frame is fully masked with the background (key frame) and ready for mask propagation to the subsequent frames via automatic mask fitting methods.

FIGS. **6**A and **6**B show the child window displaying an enlarged and scalable single image of the series of sequential images in display memory. The Child window enables the

operator to manipulate masks interactively on a single frame or in multiple frames during real time or slowed motion.

FIGS. **7**A and **7**B shows a single mask (flesh) is propagated automatically to all frames in the display memory.

FIG. **8** shows all masks associated with the motion object are propagated to all sequential frames in display memory.

FIG. **9**A shows a picture of a face.

FIG. **9**B shows a close up of the face in FIG. **9**A wherein the "small dark" pixels shown in FIG. **9**B are used to calculate a weighed index using bilinear interpolation.

FIGS. **10**A-D show searching for a Best Fit on the Error Surface: An error surface calculation in the Gradient Descent Search method involves calculating mean squared differences of pixels in the square fit box centered on reference image pixel (x0, y0), between the reference image frame and the corresponding (offset) location (x, y) on the search image frame.

FIGS. **11**A-C show a second search box derived from a descent down the error surface gradient (evaluated separately), for which the evaluated error function is reduced, possibly minimized, with respect to the original reference box (evident from visual comparison of the boxes with the reference box in FIGS. **10**A, B, C and D.).

FIG. **12** depicts the gradient component evaluation. The error surface gradient is calculated as per definition of the gradient. Vertical and horizontal error deviations are evaluated at four positions near the search box center position, and combined to provide an estimate of the error gradient for that position **12**.

FIG. **13** shows a propagated mask in the first sequential instance where there is little discrepancy between the underlying image data and the mask data. The dress mask and hand mask can be clearly seen to be off relative to the image data.

FIG. **14** shows that by using the automatic mask fitting routine, the mask data adjusts to the image data by referencing the underlying image data in the preceding image.

FIG. **15** shows the mask data in later images within the sequence show marked discrepancy relative to the underlying image data. Eye makeup, lipstick, blush, hair, face, dress and hand image data are all displaced relative to the mask data.

FIG. **16** shows that the mask data is adjusted automatically based on the underlying image data from the previous mask and underlying image data.

FIG. **17** shows the mask data from FIG. **16** is shown with appropriate color transforms after whole frame automatic mask fitting. The mask data is adjusted to fit the underlying luminance pattern based on data from the previous frame or from the initial key frame.

FIG. **18** shows polygons that are used to outline a region of interest for masking in frame one. The square polygon points snap to the edges of the object of interest. Using a Bezier curve the Bezier points snap to the object of interest and the control points/curves shape to the edges.

FIG. **19** shows the entire polygon or Bezier curve is carried to a selected last frame in the display memory where the operator adjusts the polygon points or Bezier points and curves using the snap function which automatically snaps the points and curves to the edges of the object of interest.

FIG. **20** shows that if there is a marked discrepancy between the points and curves in frames between the two frames where there was an operator interactive adjustment, the operator will further adjust a frame in the middle of the plurality of frames where there is maximum error of fit.

FIG. **21** shows that when it is determined that the polygons or Bezier curves are correctly animating between the two adjusted frames, the appropriate masks are applied to all frames.

FIG. **22** shows the resulting masks from a polygon or Bezier animation with automatic point and curve snap to edges. The brown masks are the color transforms and the green masks are the arbitrary color masks.

FIG. **23** shows an example of two pass blending: The objective in two-pass blending is to eliminate moving objects from the final blended mosaic. This can be done by first blending the frames so the moving object is completely removed from the left side of the background mosaic. As shown in FIG. **23**, the character can is removed from the scene, but can still be seen in the right side of the background mosaic.

FIG. **24** shows the second pass blend. A second background mosaic is then generated, where the blend position and width is used so that the moving object is removed from the right side of the final background mosaic. As shown in FIG. **24**, the character can is removed from the scene, but can still be seen the left side of the background mosaic. In the second pass blend as shown in FIG. **24**, the moving character is shown on the left.

FIG. **25** shows the final background corresponding to FIGS. **23**-**24**. The two-passes are blended together to generate the final blended background mosaic with the moving object removed from the scene. As shown in FIG. **25**, the final blended background with moving character is removed.

FIG. **26** shows an edit frame pair window.

FIG. **27** shows sequential frames representing a camera pan that are loaded into memory. The motion object (butler moving left to the door) has been masked with a series of color transform information leaving the background black and white with no masks or color transform information applied.

FIG. **28** shows six representative sequential frames of the pan above are displayed for clarity.

FIG. **29** shows the composite or montage image of the entire camera pan that was built using phase correlation techniques. The motion object (butler) included as a transparency for reference by keeping the first and last frame and averaging the phase correlation in two directions. The single montage representation of the pan is color designed using the same color transform masking techniques as used for the foreground object.

FIG. **30** shows that the sequence of frames in the camera pan after the background mask color transforms the montage has been applied to each frame used to create the montage. The mask is applied where there is no pre-existing mask thus retaining the motion object mask and color transform information while applying the background information with appropriate offsets.

FIG. **31** shows a selected sequence of frames in the pan for clarity after the color background masks have been automatically applied to the frames where there is no pre-existing masks.

FIG. **32** shows a sequence of frames in which all moving objects (actors) are masked with separate color transforms.

FIG. **33** shows a sequence of selected frames for clarity prior to background mask information. All motion elements have been fully masked using the automatic mask-fitting algorithm.

FIG. **34** shows the stationary background and foreground information minus the previously masked moving objects. In this case, the single representation of the complete background has been masked with color transforms in a manner similar to the motion objects. Note that outlines of removed foreground objects appear truncated and unrecognizable due to their motion across the input frame sequence interval, i.e., the black objects in the frame represent areas in which the motion objects (actors) never expose the background and

foreground. The black objects are ignored during the masking operation in colorization-only projects because the resulting background mask is later applied to all frames used to create the single representation of the background only where there is no pre-existing mask. In depth conversion projects the missing data area may be displayed so that image data may be obtained/generated for the missing data area so as to provide visually believable image data when translating foreground objects horizontally to generate a second viewpoint.

FIG. **35** shows the sequential frames in the static camera scene cut after the background mask information has been applied to each frame with appropriate offsets and where there is no pre-existing mask information.

FIG. **36** shows a representative sample of frames from the static camera scene cut after the background information has been applied with appropriate offsets and where there is no pre-existing mask information.

FIGS. **37**A-C show embodiments of the Mask Fitting functions, including calculate fit grid and interpolate mask on fit grid.

FIGS. **38**A-B show embodiments of the extract background functions.

FIGS. **39**A-C show embodiments of the snap point functions.

FIGS. **40**A-C show embodiments of the bimodal threshold masking functions, wherein FIG. **40**C corresponds to step **2.1** in FIG. **40**A, namely "Create Image of Light/Dark Cursor Shape" and FIG. **40**B corresponds to step **2.2** in FIG. **40**A, namely "Apply Light/Dark shape to mask".

FIGS. **41**A-B show embodiments of the calculate fit value functions.

FIG. **42** shows two image frames that are separated in time by several frames, of a person levitating a crystal ball wherein the various objects in the image frames are to be converted from two-dimensional objects to three-dimensional objects.

FIG. **43** shows the masking of the first object in the first image frame that is to be converted from a two-dimensional image to a three-dimensional image.

FIG. **44** shows the masking of the second object in the first image frame.

FIG. **45** shows the two masks in color in the first image frame allowing for the portions associated with the masks to be viewed.

FIG. **46** shows the masking of the third object in the first image frame.

FIG. **47** shows the three masks in color in the first image frame allowing for the portions associated with the masks to be viewed.

FIG. **48** shows the masking of the fourth object in the first image frame.

FIG. **49** shows the masking of the fifth object in the first image frame.

FIG. **50** shows a control panel for the creation of three-dimensional images, including the association of layers and three-dimensional objects to masks within an image frame, specifically showing the creation of a Plane layer for the sleeve of the person in the image.

FIG. **51** shows a three-dimensional view of the various masks shown in FIGS. **43-49**, wherein the mask associated with the sleeve of the person is shown as a Plane layer that is rotated toward the left and right viewpoints on the right of the page.

FIG. **52** shows a slightly rotated view of FIG. **51**.

FIG. **53** shows a slightly rotated view of FIG. **51**.

FIG. **54** shows a control panel specifically showing the creation of a sphere object for the crystal ball in front of the person in the image.

FIG. **55** shows the application of the sphere object to the flat mask of the crystal ball, that is shown within the sphere and as projected to the front and back of the sphere to show the depth assigned to the crystal ball.

FIG. **56** shows a top view of the three-dimensional representation of the first image frame showing the Z-dimension assigned to the crystal ball shows that the crystal ball is in front of the person in the scene.

FIG. **57** shows that the sleeve plane rotating in the X-axis to make the sleeve appear to be coming out of the image more.

FIG. **58** shows a control panel specifically showing the creation of a Head object for application to the person's face in the image, i.e., to give the person's face realistic depth without requiring a wire model for example.

FIG. **59** shows the Head object in the three-dimensional view, too large and not aligned with the actual person's head.

FIG. **60** shows the Head object in the three-dimensional view, resized to fit the person's face and aligned, e.g., translated to the position of the actual person's head.

FIG. **61** shows the Head object in the three-dimensional view, with the Y-axis rotation shown by the circle and Y-axis originating from the person's head thus allowing for the correct rotation of the Head object to correspond to the orientation of the person's face.

FIG. **62** shows the Head object also rotated slightly clockwise, about the Z-axis to correspond to the person's slightly tilted head.

FIG. **63** shows the propagation of the masks into the second and final image frame.

FIG. **64** shows the original position of the mask corresponding to the person's hand.

FIG. **65** shows the reshaping of the mask, that can be performed automatically and/or manually, wherein any intermediate frames get the tweened depth information between the first image frame masks and the second image frame masks.

FIG. **66** shows the missing information for the left viewpoint as highlighted in color on the left side of the masked objects in the lower image when the foreground object, here a crystal ball is translated to the right.

FIG. **67** shows the missing information for the right viewpoint as highlighted in color on the right side of the masked objects in the lower image when the foreground object, here a crystal ball is translated to the left.

FIG. **68** shows an anaglyph of the final depth enhanced first image frame viewable with Red/Blue 3-D glasses.

FIG. **69** shows an anaglyph of the final depth enhanced second and last image frame viewable with Red/Blue 3-D glasses, note rotation of person's head, movement of person's hand and movement of crystal ball.

FIG. **70** shows the right side of the crystal ball with fill mode "smear", wherein the pixels with missing information for the left viewpoint, i.e., on the right side of the crystal ball are taken from the right edge of the missing image pixels and "smeared" horizontally to cover the missing information.

FIG. **71** shows a mask or alpha plane, for an actor's upper torso and head (and transparent wings). The mask may include opaque areas shown as black and transparent areas that are shown as grey areas.

FIG. **72** shows an occluded area, that corresponds to the actor of FIG. **71**, and that shows an area of the background that is never exposed in any frame in a scene. This may be a composite background for example.

FIG. **73** shows the occluded area artistically rendered to generate a complete and realistic background for use in two-dimensional to three-dimensional conversion, so as to enable an artifact-free conversion.

FIG. **73**A shows the occluded area partially drawn or otherwise rendered to generate just enough of a realistic looking background for use in minimizing artifacts two-dimensional to three-dimensional conversion.

FIG. **74** shows a light area of the shoulder portion on the right side of FIG. **71** that represents a gap where stretching (as is also shown in FIG. **70**) would be used when shifting the foreground object to the left to create a right viewpoint. The dark portion of the figure is taken from the background where data is available in at least one frame of a scene.

FIG. **75** shows an example of the stretching of pixels, i.e., smearing, corresponding to the light area in FIG. **74** without the use of a generated background, i.e., if no background data is available for an area that is occluded in all frames of a scene.

FIG. **76** shows a result of a right viewpoint without artifacts on the edge of the shoulder of the person wherein the dark area includes pixels available in one or more frames of a scene, and generated data for always-occluded areas of a scene.

FIG. **77** shows an example of a computer-generated element, here a robot, which is modeled in three-dimensional space and projected as a two-dimensional image. If metadata such as alpha, mask, depth or any combination thereof exists, the metadata can be utilized to speed the conversion process from two-dimensional image to a pair of two-dimensional images for left and right eye for three-dimensional viewing.

FIG. **78** shows an original image separated into a background and foreground elements, (mountain and sky in the background and soldiers in the bottom left also see FIG. **79**) along with the imported color and depth of the computer-generated element, i.e., the robot with depth automatically set via the imported depth metadata. As shown in the background, any area that is covered for the scene can be artistically rendered for example to provide believable missing data, as is shown in FIG. **73** based on the missing data of FIG. **73**A, which results in artifact free edges as shown in FIG. **76** for example.

FIG. **79** shows masks associated with the photograph of soldiers in the foreground to apply depth to the various portions of the soldiers that lie in depth in front of the computer-generated element, i.e., the robot. The dashed lines horizontally extending from the mask areas show horizontal translation of the foreground objects takes place and where imported metadata can be utilized to accurately auto-correct over-painting of depth or color on the masked objects when metadata exists for the other elements of a movie. For example, when an alpha exists for the objects that occur in front of the computer-generated elements. One type of file that can be utilized to obtain mask edge data is a file with alpha file and/or mask data such as an RGBA file.

FIG. **80** shows an imported alpha layer which can also be utilized as a mask layer to limit the operator defined, and potentially less accurate masks used for applying depth to the edges of the three soldiers A, B and C. In addition, a computer-generated element for dust can be inserted into the scene along the line annotated as "DUST", to augment the reality of the scene.

FIG. **81** shows the result of using the operator-defined masks without adjustment when overlaying a motion element such as the soldier on the computer-generated element such as the robot. Through use of the alpha metadata of FIG. **80** applied to the operated-defined mask edges of FIG. **79**, artifact free edges on the overlapping areas is thus enabled.

FIG. **82** shows a source image to be depth enhanced and provided along with left and right translation files and alpha masks so that downstream workgroups may perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove and/or or adjust depths without iterative workflow paths back to the original workgroups.

FIG. **83** shows masks generated by the mask workgroup for the application of depth by the depth augmentation group, wherein the masks are associated with objects, such as for example human recognizable objects in the source image of FIG. **82**.

FIG. **84** shows areas where depth is applied generally as darker for nearer objects and lighter for objects that are further away.

FIG. **85**A shows a left UV map containing translations or offsets in the horizontal direction for each source pixel.

FIG. **85**B shows a right UV map containing translations or offsets in the horizontal direction for each source pixel.

FIG. **85**C shows a black value shifted portion of the left UV map of FIG. **85**A to show the subtle contents therein.

FIG. **85**D shows a black value shifted portion of the right UV map of FIG. **85**B to show the subtle contents therein.

FIG. **86**A shows a left U map containing translations or offsets in the horizontal direction for each source pixel.

FIG. **86**B shows a right U map containing translations or offsets in the horizontal direction for each source pixel.

FIG. **86**C shows a black value shifted portion of the left U map of FIG. **86**A to show the subtle contents therein.

FIG. **86**D shows a black value shifted portion of the right U map of FIG. **86**B to show the subtle contents therein.

FIG. **87** shows known uses for UV maps, wherein a three-dimensional model is unfolded so that an image in UV space can be painted onto the 3D model using the UV map.

FIG. **88** shows a disparity map showing the areas where the difference between the left and right translation maps is the largest.

FIG. **89** shows a left eye rendering of the source image of FIG. **82**.

FIG. **90** shows a right eye rendering of the source image of FIG. **82**.

FIG. **91** shows an anaglyph of the images of FIG. **89** and FIG. **90** for use with Red/Blue glasses.

FIG. **92** shows an image that has been masked and is in the process of depth enhancement for the various layers.

FIG. **93** shows a UV map overlaid onto an alpha mask associated with the actress shown in FIG. **92** which sets the translation offsets in the resulting left and right UV maps based on the depth settings of the various pixels in the alpha mask.

FIG. **94** shows a workspace generated for a second depth enhancement program, or compositing program such as NUKE®, i.e., generated for the various layers shown in FIG. **92**, i.e., left and right UV translation maps for each of the alphas wherein the workspace allows for quality assurance personnel (or other work groups) to perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts or otherwise adjust masks and hence alter the 3D image pair (or anaglyph) without iteratively sending fixes to any other workgroup.

FIG. **95** shows a workflow for iterative corrective workflow.

FIG. **96** shows an embodiment of the workflow enabled by one or more embodiments of the system in that each workgroup can perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts and otherwise correct work product from another workgroup without iterative delays associated with re-rendering/ray-tracing or sending work product back through the workflow for corrections.

FIG. **97** illustrates and architectural view of an embodiment of the invention.

FIG. **98** illustrates an annotated view of a session manager window utilized to define a plurality of images to work on or assign work to.

FIG. **99** illustrates a view of the production display showing a project, shots and tasks related to the selected shot, along with status for each task context associated with the shot.

FIG. **100** illustrates a view of the actuals associated with a particular shot within a project for each task context associated with the shot wherein "under bid" task actuals are shown in a first manner and tasks that are within a predefined percentage within the bid amount are shown in a second manner while tasks that are over bid are shown in a third manner.

FIG. **101** illustrates the amount of disk space that may be saved by deleting files which can be reconstructed from other files, for example after completion of a project to save disk drive expenditures.

FIG. **102** illustrates a view of an artist display showing the task context, project, shot, status, tools, start time button, check-in input, rendering input, internal shot review input, meal, start/time/stop, review and submit inputs.

FIG. **103** illustrates an annotated view of the menu bar of the artist display.

FIG. **104** illustrates an annotated view of the task row of the artist display.

FIG. **105** illustrates an annotated view of the main portion of the user interface of the artist display.

FIG. **106** illustrates a build timeline display for the artist display for creating a timeline to work on.

FIG. **107** illustrates a browse snapshots display for the artist display that enables an artist to view the snapshot of a shot or otherwise cache important information related to a shot so that the database does not have to field requests for often utilized data.

FIG. **108** illustrates an artist actual window showing the actuals of time spent on tasks, for example versus time allocated for the tasks, with dropdown menus for specific timesheets.

FIG. **109** illustrates a notes display for the artist display that enables artists to enter a note related to the shot.

FIG. **110** illustrates a check in display for the artist display that enables work to be checked in after work on the shot is complete.

FIG. **111** illustrates a view of an editorial display showing the project, filter inputs, timeline inputs and search results showing the shots in the main portion of the window along with the context of work and assigned worker.

FIG. **112** illustrates a view of the session manager display of the editorial display for selecting shots to review.

FIG. **113** illustrates a view of the advanced search display of the editorial display.

FIG. **114** illustrates a view of the simple search display of the editorial display.

FIG. **115** illustrates a view of the reviewing pane for a shot also showing integrated notes and/or snapshot information in the same frame.

FIG. **116** illustrates a view of the timelines to select for review and/or checkin after modification.

FIG. **117** illustrates the annotations added to a frame for feedback using the tool of FIG. **117**.

FIG. **118** illustrates an overview of the grouping tool interface according to one or more embodiments.

FIG. **119** illustrates a view of the grouping tool interface with metadata tags and shot tables according to one or more embodiments.

FIG. **120** illustrates a view of the grouping tool interface with shots that share metadata characteristics according to one or more embodiments.

FIG. **121** illustrates a view of the grouping tool interface with a key select shot and color code, depth complexity and color code and clean plate complexity and color code according to one or more embodiments.

FIG. **122** illustrates another view of the grouping tool interface according to one or more embodiments.

FIG. **123** illustrates a close-up view of the grouping tool interface according to one or more embodiments.

FIG. **124** illustrates a close-up view of a shot of the grouping tool interface according to one or more embodiments.

FIG. **125** illustrates a close-view of a plurality of shot to add metadata to according to one or more embodiments.

FIG. **126** illustrates an overall view of the grouping tool interface with a shot from a plurality of images with metadata.

DETAILED DESCRIPTION OF THE INVENTION

FIGS. **118-126** illustrate embodiments of the invention directed at non-linear workflow. The interface is described at the bottom portion of the detailed description after the project management system portion and overall production workflow is described.

FIG. **97** illustrates and architectural view of an embodiment of the project management portion of the invention and system elements that enable non-linear workflow on motion picture projects. One or more embodiments of the system include computer **9702** and database **9701** coupled with computer **9702**. Any computer architecture having any number of computers, for example coupled via a computer communication network is in keeping with the spirit of the invention. Database **9701** coupled with computer **9702** includes at least a project table, shot table, task table and timesheet table. The project table generally includes project identifier and description of a project related to a motion picture. The shot table generally includes a shot identifier and references a plurality of images with a starting frame value and an ending frame value wherein the plurality of images are associated with the motion picture that is associated with the project. The shot table generally includes at least one shot having status related to progress of work performed on the shot. The task table generally references the project using a project identifier in also located in the project table. The task table generally includes at least one task which generally includes a task identifier and an assigned worker, e.g., artist, and which may also include a context setting associated with a type of task related to motion picture work selected from region design, setup, motion, composite, and review for example (or any other set of motion picture related types of tasks). The context setting may also imply or have a default workflow so that region design flows into depth which flows into composite. This enables the system to assign the next type of task, or context that the shot is to have work performed on. The flow may be linear or may iterate back for rework for example. The at least one task generally includes a time allocated to complete the at least one task. The timesheet item table generally references the project identifier in the project table and the task identifier in the task table. The task table generally includes at least one timesheet item that includes a start time and an end time. The completion of tasks may in one or more embodiments set the context of the task to the next task in sequence in a workflow and the system may automatically notify the next worker in the workflow based on the next context of work to be performed and the workers may work under different contexts as previously described. In one or

more embodiments contexts may have sub-contexts, i.e., region design may be broken into masking and outsource masking while depth may be broken into key frame and motion contexts, depending on the desired workflow for the specific type of operations to be performed on a motion picture project.

Embodiments of the database may also include a snapshot table which includes a snapshot identifier and search type and which includes a snapshot of the at least one shot, for example that includes a subset of the at least one shot wherein the snapshot is cached on the computer to reduce access to the shot table. Other embodiments of the snapshot table keep track of the resources on the network, stores information about the resources and track versioning of the resource. Embodiments may also include other context settings for other types of task categories, for example source and cleanup related tasks. Any other context settings or values that are related to motion picture work may also be included in keeping with the spirit of the invention. Embodiments of the database may also include an asset request table that includes an asset request identifier and shot identifier that may be utilized to request work on assets or assets themselves to be worked on or created by other workers for example. Embodiments of the database may also include a request table that includes an mask request identifier and shot identifier and that may be utilized to request any type of action by another worker for example. Embodiments of the database may also include a note table which includes a note identifier and that references the project identifier and that includes at least one note related to at least one of the plurality of images from the motion picture. Embodiments of the database may also include a delivery table that includes a delivery identifier that references the project identifier and which includes information related to delivery of the motion picture.

One or more embodiments of the database may utilize a schema as follows or any other schema that is capable of supporting the functionality of the invention as specifically programmed in computer **9702** and as described in any combination or sub-combination as follows so long as motion picture project management may be performed as detailed herein or to in any other way better manage motion picture projects using the exemplary specifications herein:

Project Table

unique project identifier, project code (text name), title of motion picture, type of project (test or for hire), last database update date and time, status (retired or active), last version update of the database, project type (colorization, effects, 2D→3D conversion, feature film, catalogue), lead worker, review drive (where the review shots are stored).

Task Table

unique task identifier, assigned worker, description (what to do), status (ingested, waiting, complete, returned, approved), bid start date, bid end date, bid duration, actual start date, actual end date, priority, context (stacking, asset, motion, motion visual effects, outsource, cleanup, alpha generation, composite, masking, clean plate, setup, keyframe, quality control), project code in project table, supervisor or production or editorial worker, time spent per process.

Snapshot Table

unique snapshot identifier, search type (which project the search is for), description (notes related to the shot), login (worker associated with the snapshot), timestamp, context (setup, cleanup, motion, composite . . .), version of the snapshot on the shot, snapshot type (directory, file, information, review), project code, review sequence data (where the data is stored on the network), asset name (alpha, mask, . . .), snapshots used (codes of other snapshots used to make this snapshot), checkin path (path to where the data was checked in from), tool version, review date, archived, rebuildable (true or false), source delete, source delete date, source delete login.

Note Table

unique note identifier, project code, search type, search id, login (worker id), context (composite, review, motion, editorial . . .), timestamp, note (text description of note associated with set of images defined by search).

Delivery Table

unique deliver identifier, login (of worker), timestamp, status (retired or not), delivery method (how it was delivered), description (what type of media used to deliver project), returned (true or false), drive (serial number of the drive), case (serial number on the case), delivery date, project identifier, client (text name of client), producer (name of producer).

Delivery_Item Table

unique delivery item identifier, timestamp, delivery code, project identifier, file path (where delivery item is stored).

Timesheet Table

time sheet unique identifier, login (worker), timestamp, total time, timesheet approver, start time, end time, meal 1 (half hour break start time), meal 2 (half hour break start time), status (pending or approved).

Timesheet_Item Table

timesheet item unique identifier, login (worker), timestamp, context (region design, composite, rendering, motion, management, mask cleanup, training, cleanup, admin), project identifier, timesheet identifier, start time, end time, status (pending or approved), approved by (worker), task identifier.

Sequence Table

sequence unique identifier, login (worker that defined sequence), timestamp, shot order (that makes up the sequence).

Shot Table

shot unique identifier, login (worker that defined shot), timestamp, shot status (in progress, final, final client approval), client status (composite in progress, depth client review, composite client review, final), description (text description of shot, e.g., 2 planes flying by each other), first frame number, last frame number, number of frames, assigned worker, region design, depth completion date, depth worker assigned, composite supervisor, composite lead worker, composite completion date.

Asset_Request Table

asset unique identifier, timestamp, asset worker assigned, status (pending or resolved), shot identifier, problem description in text, production worker, lead worker assigned, priority, due date.

Mask_Request Table

mask request unique identifier, login (worker making mask request), timestamp, depth artist, depth lead, depth coordinator or production worker, masking issues, masks (versions with issue related to mask request), source used, due date, rework notes.

In one or more embodiments of the invention, the computer is generally configured to present a session manager to select a series of images within a shot to work on and/or assign tasks to or to review. The computer is generally configured to present a first display configured to be viewed by production that includes a search display having a context, project, shot, status and artist and wherein the second display further includes a list of a plurality of artists and respective status and actuals based on time spent in the at least one timesheet item versus the time allocated per the at least one task associated with the at least one shot.

The database may also include tables and fields for the support of non-linear workflow, including any type of meta-data related to a scene or regions within a image or several images that comprise a scene as is discussed at the end of this section.

FIG. **98** illustrates an annotated view of a session manager window utilized to define a plurality of images to work on or assign work to or to review for example. Computer **9702** accepts inputs for the project, sequence (the motion picture or a trailer for example that utilizes shots in a particular sequence), along with the shot, mask version and various frame offsets and optionally downloads the images to the local computer for local processing for example. Each field on the figure is annotated with further details.

FIG. **99** illustrates a view of the production display showing a project, shots and tasks related to the selected shot, along with status for each task context associated with the shot. As shown on the left, the shots that make up a project may be selected, which redirects the main portion of the window to display information related to the selected shot(s) including tabs for "shot info", "assets" used in the shot, frames to select to view, notes, task information, actuals, check-in and data integrity. As shown in the main portion of the window, several task contexts are shown with their associated status, assigned workers, etc. Production utilizes this display and the computer accepts inputs from the production worker, e.g., user via this display to set tasks for artists along with allotted times. A small view of the shot is shown in the lower left of the display to give production workers a view of the motion picture shot related to the tasks and context settings. One potential goal of the production role is to assign tasks and review status and actuals, while one potential goal of the artist is to use a set of tools to manipulate images, while one potential goal of the reviewer is to have a higher resolution image for review with integrated meta data related to the shot and status thereof. In other words, the displays in the system are tailored for roles, yet integrated to prioritize the relevant information mainly of importance to that role in the motion picture creation/conversion process.

FIG. **100** illustrates a view of the actuals associated with a particular shot within a project for each task context associated with the shot wherein "under bid" task actuals are shown in a first manner and tasks that are within a predefined percentage within the bid amount are shown in a second manner while tasks that are over bid are shown in a third manner.

FIG. **101** illustrates the amount of disk space that may be saved by deleting files which can be reconstructed from other files, for example after completion of a project to save disk drive expenditures. As shown rebuildable amounts for a project only partially complete can be in the terabyte range easily. By being able to securely delete rebuildable assets and compress other assets after a project is over, huge amounts of disk drive costs may be saved. In one or more embodiments the computer accesses the database and determines which resources depend on other resources and whether they can be compressed and with what general compression ratio that may be calculated in advance and/or based on other projects for example. The computer then calculates the total storage and the amount of storage that may be freed up by compression and/or regeneration of resources and displays the information on a computer display for example.

The computer is also generally configured to present a second display configured to be viewed by an artist that includes at least one daily assignment having a context, project, shot and a status input that is configured to update the status in the task table and a timer input that is configured to update the start time and the end time in the timesheet item table.

FIG. **102** illustrates a view of an artist display showing the task context, project, shot, status, tools, start time button, check-in input, rendering input, internal shot review input, meal, start/time/stop, review and submit inputs. This enables motion picture related tasks to be viewed and updated as work on a project progresses, which gives production a specific motion picture project management related view into the status of a conversion and/or special effects movie project.

FIG. **103** illustrates an annotated view of the menu bar of the artist display. The menu bar is shown at the top left portion of the display in FIG. **102**.

FIG. **104** illustrates an annotated view of the task row of the artist display. This annotated view shows only one row, however multiple rows may be displayed as per FIG. **102**.

FIG. **105** illustrates an annotated view of the main portion of the user interface of the artist display of FIG. **102**.

FIG. **106** illustrates a build timeline display for the artist display for creating a timeline to work on.

FIG. **107** illustrates a browse snapshots display for the artist display that enables an artist to view the snapshot of a shot or otherwise cache important information related to a shot so that the database does not have to field requests for often utilized data. The snapshot keeps track of the locations of various files associated with a shot, and keeps track of other information related to a work product related to the shot, i.e., source, masks, resolution, file type. In addition, the snapshot keeps track of versioning of the various files and file types and optionally the versions of the tools utilized to work on the various files.

FIG. **108** illustrates an artist actual window showing the actuals of time spent on tasks, for example versus time allocated for the tasks, with dropdown menus for specific timesheets.

FIG. **109** illustrates a notes display for the artist display that enables artists to enter a note related to the shot.

FIG. **110** illustrates a check in display for the artist display that enables work to be checked in after work on the shot is complete.

The computer is generally also configured to present a third display configured to be viewed by an editor, i.e., an editorial worker, that includes an annotation frame configured to accept commentary or drawing or both commentary and drawing on the at least one of said plurality of images associated with the at least one shot. One or more embodiments of the computer may be configured to provide the third display configured to be viewed by an editor that includes an annotation overlaid on at least one of the plurality of images. This capability provides information on one display that has generally required three workers to integrate in known systems, and which is novel in and of itself.

FIG. **111** illustrates a view of an editorial display showing the project, filter inputs, timeline inputs and search results showing the shots in the main portion of the window along with the context of work and assigned worker.

FIG. **112** illustrates a view of the session manager display of the editorial display for selecting shots to review.

FIG. **113** illustrates a view of the advanced search display of the editorial display.

FIG. **114** illustrates a view of the simple search display of the editorial display.

FIG. **115** illustrates a view of the reviewing pane for a shot also showing integrated notes and/or snapshot information in the same frame. This view has in the past generally required three workers to create and saves great amounts of time and

speeds the review process greatly. Any type of information may be overlaid onto an image to enable consolidated disparate views of images and related data on one display.

FIG. **116** illustrates a view of the timelines to select for review and/or checkin after modification.

FIG. **117** illustrates the annotations added to a frame for feedback using the tool of FIG. **117**.

One or more embodiments of the computer are configured to accept a rating input from production or editorial based on work performed by the artist, optionally in a blind manner in which the reviewer does not know the identity of the artist in order to prevent favoritism for example. One or more embodiments of the computer are configured to accept a difficulty of the at least one shot and calculate a rating based on work performed by the artist and based on the difficulty of the shot and time spent on the shot. One or more embodiments of the computer are configured to accept a rating input from production or editorial based on work performed by the artist, or, accept a difficulty of the at least one shot and calculate a rating based on work performed by the artist and based on the difficulty of the shot and time spent on the shot, and, signify an incentive with respect to the artist based on the rating accepted by the computer or calculated by the computer. One or more embodiments of the computer are configured to estimate remaining cost based on the actuals that are based on total time spent for all of the at least one tasks associated with all of the at least one shot in the project versus time allocated for all of the at least one tasks associated with all of the at least one shot in the project. One or more embodiments of the computer are configured to compare the actuals associated with a first project with actuals associated with a second project and signify at least one worker to be assigned from the first project to the second project based on at least one rating of the first worker that is assigned to the first project. One or more embodiments of the computer are configured to analyze a prospective project having a number of shots and estimated difficulty per shot and based on actuals associated with the project, calculate a predicted cost for the prospective project. One or more embodiments of the computer are configured to analyze a prospective project having a number of shots and estimated difficulty per shot and based on the actuals associated with a first previously performed project and a second previously performed project that completed after the first previously performed project, calculate a derivate of the actuals, calculate a predicted cost for the prospective project based on the derivative of the actuals. For example, as the process improves, tools improve and workers improve, the efficiency of work improves and the budgeting and bid processes can take this into account by calculating how efficiency is changing versus time and use this rate of change to predict costs for a prospective project. One or more embodiments of the computer are configured to analyze the actuals associated with said project and divide completed shots by total shots associated with said project and present a time of completion of the project. One or more embodiments of the computer are configured to analyze the actuals associated with the project and divide completed shots by total shots associated with the project, present a time of completion of the project, accept an input of at least one additional artist having a rating, accept a number of shots in which to use the additional artist, calculate a time savings based on the at least one additional artist and the number of shots, subtract the time savings from the time of completion of the project and present an updated time of completion of the project. One or more embodiments of the computer are configured to calculate amount of disk space that may be utilized to archive the project and signify at least one asset that may be rebuilt from other assets to avoid archival of the at least one asset. One or more embodiments of the computer are configured to display an error message if the artist is working with a frame number that is not current in the at least one shot. This may occur when fades, dissolves or other effects lengthen a particular shot for example wherein the shot contains frames not in the original source assets.

Overview of Various Motion Picture Workflows

Feature Film and TV series Data Preparation for Colorization/Depth enhancement: Feature films are tele-cined or transferred from 35 mm or 16 mm film using a high resolution scanner such as a 10-bit SPIRIT DATACINE® or similar device to HDTV (1920 by 1080 24P) or data-cined on a laser film scanner such as that manufactured by IMAGICA® Corp. of America at a larger format 2000 lines to 4000 lines and up to 16 bits of grayscale. The high resolution frame files are then converted to standard digital files such as uncompressed TIP files or uncompressed TGA files typically in 16 bit three-channel linear format or 8 bit three channel linear format. If the source data is HDTV, the 10-bit HDTV frame files are converted to similar TIF or TGA uncompressed files at either 16-bits or 8-bit per channel. Each frame pixel is then averaged such that the three channels are merged to create a single 16 bit channel or 8 bit channel respectively. Any other scanning technologies capable of scanning an existing film to digital format may be utilized. Currently, many movies are generated entirely in digital format, and thus may be utilized without scanning the movie. For digital movies that have associated metadata, for example for movies that make use of computer-generated characters, backgrounds or any other element, the metadata can be imported for example to obtain an alpha and/or mask and/or depth for the computer-generated element on a pixel-by-pixel or sub-pixel-by-sub-pixel basis. One format of a file that contains alpha/mask and depth data is the RGBAZ file format, of which one implementation is the EXR file format.

Digitization Telecine and Format Independence Monochrome elements of either 35 or 16 mm negative or positive film are digitized at various resolutions and bit depth within a high resolution film scanner such as that performed with a SPIRIT DATACINE® by PHILIPS® and EASTMAN KODAK® which transfers either 525 or 625 formats, HDTV, (HDTV) 1280×720/60 Hz progressive, 2K, DTV (ATSC) formats like 1920×1080/24 Hz/25 Hz progressive and 1920×1080/48 Hz/50 Hz segmented frame or 1920×1080 501 as examples. The invention provides improved methods for editing film into motion pictures. Visual images are transferred from developed motion picture film to a high definition video storage medium, which is a storage medium adapted to store images and to display images in conjunction with display equipment having a scan density substantially greater than that of an NTSC compatible video storage medium and associated display equipment. The visual images are also transferred, either from the motion picture film or the high definition video storage medium to a digital data storage format adapted for use with digital nonlinear motion picture editing equipment. After the visual images have been transferred to the high definition video storage medium, the digital nonlinear motion picture editing equipment is used to generate an edit decision list, to which the motion picture film is then conformed. The high definition video storage medium is generally adapted to store and display visual images having a scan density of at least 1080 horizontal lines. Electronic or optical transformation may be utilized to allow use of visual aspect ratios that make full use of the storage formats used in the method. This digitized film data as well as data already transferred from film to one of a multiplicity of formats such as HDTV are entered into a conversion system such as the

HDTV STILL STORE® manufactured by AVICA® Technology Corporation. Such large scale digital buffers and data converters are capable of converting digital image to all standard formats such as 1080i HDTV formats such as 720p, and 1080p/24. An Asset Management System server provides powerful local and server back ups and archiving to standard SCSI devices, C2-level security, streamlined menu selection and multiple criteria database searches.

During the process of digitizing images from motion picture film the mechanical positioning of the film frame in the telecine machine suffers from an imprecision known as "film weave", which cannot be fully eliminated. However various film registration and ironing or flattening gate assemblies are available such as that embodied in U.S. Pat. No. 5,328,073, Film Registration and Ironing Gate Assembly, which involves the use of a gate with a positioning location or aperture for focal positioning of an image frame of a strip film with edge perforations. Undersized first and second pins enter a pair of transversely aligned perforations of the film to register the image frame with the aperture. An undersized third pin enters a third perforation spaced along the film from the second pin and then pulls the film obliquely to a reference line extending between the first and second pins to nest against the first and second pins the perforations thereat and register the image frame precisely at the positioning location or aperture. A pair of flexible bands extending along the film edges adjacent the positioning location moves progressively into incrementally increasing contact with the film to iron it and clamp its perforations against the gate. The pins register the image frame precisely with the positioning location, and the bands maintain the image frame in precise focal position. Positioning can be further enhanced following the precision mechanical capture of images by methods such as that embodied in U.S. Pat. No. 4,903,131, Method For The Automatic Correction Of Errors In Image Registration During Film Scanning.

To remove or reduce the random structure known as grain within exposed feature film that is superimposed on the image as well as scratches or particles of dust or other debris which obscure the transmitted light various algorithms will be used such as that embodied in U.S. Pat. No. 6,067,125 Structure And Method For Film Grain Noise Reduction and U.S. Pat. No. 5,784,176, Method Of Image Noise Reduction Processing.

Reverse Editing of the Film Element Preliminary to Visual Database Creation:

The digital movie is broken down into scenes and cuts. The entire movie is then processed sequentially for the automatic detection of scene changes including dissolves, wipe-a-ways and cuts. These transitions are further broken down into camera pans, camera zooms and static scenes representing little or no movement. All database references to the above are entered into an edit decision list (EDT) within the database based on standard SMPTE time code or other suitable sequential naming convention. There exists, a great deal of technologies for detecting dramatic as well as subtle transitions in film content such as:

U.S. Pat. No. 5,959,697 Sep. 28, 1999 Method And System For Detecting Dissolve Transitions In A Video Signal

U.S. Pat. No. 5,920,360 Jul. 6, 1999 Method And System For Detecting Fade Transitions In A Video Signal

U.S. Pat. No. 5,841,512 Nov. 24, 1998 Methods Of Previewing And Editing Motion Pictures

U.S. Pat. No. 5,835,163 Nov. 10, 1998 Apparatus For Detecting A Cut In A Video

U.S. Pat. No. 5,767,923 16 Jun. 1998 Method And System For Detecting Cuts In A Video Signal

U.S. Pat. No. 5,778,108 Jul. 6, 1996 Method And System For Detecting Transitional Markers Such As Uniform Fields In A Video Signal

U.S. Pat. No. 5,920,360 Jun. 7, 1999 Method And System For Detecting Fade Transitions In A Video Signal

All cuts that represent the same content such as in a dialog between two or more people where the camera appears to volley between the two talking heads are combined into one file entry for later batch processing.

An operator checks all database entries visually to ensure that:

1. Scenes are broken down into camera moves

2. Cuts are consolidated into single batch elements where appropriate

3. Motion is broken down into simple and complex depending on occlusion elements, number of moving objects and quality of the optics (e.g., softness of the elements, etc).

Pre-Production—scene analysis and scene breakdown for reference frame ID and data base creation:

Files are numbered using sequential SMPTE time code or other sequential naming convention. The image files are edited together at 24-frame/sec speed (without field related 3/2 pull down which is used in standard NTSC 30 frame/sec video) onto a DVD using ADOBE® AFTER EFFECTS® or similar programs to create a running video with audio of the feature film or TV series. This is used to assist with scene analysis and scene breakdown.

Scene and Cut Breakdown:

1. A database permits the entering of scene, cut, design, key frame and other critical data in time code format as well as descriptive information for each scene and cut.

2. Each scene cut is identified relative to camera technique. Time codes for pans, zooms, static backgrounds, static backgrounds with unsteady or drifting camera and unusual camera cuts that require special attention.

3. Designers and assistant designers study the feature film for color clues and color references or for the case of depth projects, the film is studied for depth clues, generally for non-standard sized objects. Research is provided for color/depth accuracy where applicable. The Internet for example may be utilized to determine the color of a particular item or the size of a particular item. For depth projects, knowing the size of an object allows for the calculation of the depth of an item in a scene for example. For depth projects related to converting two-dimensional movies to three-dimensional movies where depth metadata is available for computer-generated elements within the movies, the depth metadata can be scaled, or translated or otherwise normalized to the coordinate system or units used for the background and motion elements for example.

4. Single frames from each scene are selected to serve as design frames. These frames are color designed or metadata is imported for depth and/or mask and/or alpha for computer-generated elements, or depth assignments (see FIGS. **42-70**) are made to background elements or motion elements in the frames to represent the overall look and feel of the feature film. Approximately 80 to 100 design frames are typical for a feature film.

5. In addition, single frames called key frames from each cut of the feature film are selected that contain all the elements within each cut that require color/depth consideration. There may be as many as 1,000 key frames. These frames will contain all the color/depth transform information necessary to apply color/depth to all sequential frames in each cut without additional color choices.

Color/Depth Selection:

Historical reference, studio archives and film analysis provides the designer with color references. Using an input device such as a mouse, the designer masks features in a selected single frame containing a plurality of pixels and assigns color to them using an HSL color space model based on creative considerations and the grayscale and luminance distribution underlying each mask. One or more base colors are selected for image data under each mask and applied to the particular luminance pattern attributes of the selected image feature. Each color selected is applied to an entire masked object or to the designated features within the luminance pattern of the object based on the unique gray-scale values of the feature under the mask.

A lookup table or color transform for the unique luminance pattern of the object or feature is thus created which represent the color to luminance values applied to the object. Since the color applied to the feature extends the entire range of potential grayscale values from dark to light the designer can insure that as the distribution of the gray-scale values representing the pattern change homogeneously into dark or light regions within subsequent frames of the movie such as with the introduction of shadows or bright light, the color for each feature also remains consistently homogeneous and correctly lighten or darken with the pattern upon which it is applied.

Depth can imported for computer-generated objects where metadata exists and/or can be assigned to objects and adjusted using embodiments of the invention using an input device such as a mouse to assign objects particular depths including contour depths, e.g., geometric shapes such as an ellipsoid to a face for example. This allows objects to appear natural when converted to three-dimensional stereoscopic images. For computer-generated elements, the imported depth and/or alpha and/or mask shape can be adjusted if desired. Assigning a fixed distance to foreground objects tends to make the objects appear as cut-outs, i.e., flat. See also FIGS. **42-70**.

Propagation of Mask Color Transform/Depth Information from One Frame to a Series of Subsequent Frames:

The masks representing designed selected color transforms/depth contours in the single design frame are then copied to all subsequent frames in the series of movie frames by one or more methods such as auto-fitting bezier curves to edges, automatic mask fitting based on Fast Fourier Transforms and Gradient Descent Calculation tied to luminance patterns in a subsequent frame relative to the design frame or a successive preceding frames, mask paint to a plurality of successive frames by painting the object within only one frame, auto-fitting vector points to edges and copying and pasting individual masks or a plurality of masks to selected subsequent frames. In addition, depth information may be "tweened" to account for forward/backward motion or zooming with respect to the camera capture location. For computer-generated elements, the alpha and/or mask data is generally correct and may be skipped for reshaping processes since the metadata associated with computer-generated elements is obtained digitally from the original model of an object and hence does not require adjustment in general. (See FIG. **37**C, step **3710** for setting mask fit location to border of CG element to potentially skip large amounts of processing in fitting masks in subsequent frames to reshape the edges to align a photographic element). Optionally, computer-generated elements may be morphed or reshaped to provide special effects not originally in a movie scene.

Single Frame Set Design and Colorization:

In embodiments of the invention, camera moves are consolidated and separated from motion elements in each scene by the creation of a montage or composite image of the background from a series of successive frames into a single frame containing all background elements for each scene and cut. The resulting single frame becomes a representation of the entire common background of a multiplicity of frames in a movie, creating a visual database of all elements and camera offset information within those frames.

In this manner most set backgrounds can be designed and colorized/depth enhanced in one pass using a single frame montage. Each montage is masked without regard to the foreground moving objects, which are masked separately. The background masks of the montage are then automatically extracted from the single background montage image and applied to the subsequent frames that were used to create the single montage using all the offsets stored in the image data for correctly aligning the masks to each subsequent frame.

There is a basic formula in filmmaking that varies little within and between feature films (except for those films employing extensive hand-held or stabilized camera shots.) Scenes are composed of cuts, which are blocked for standard camera moves, i.e., pans, zooms and static or locked camera angles as well as combinations of these moves. Cuts are either single occurrences or a combination of cut-a-ways where there is a return to a particular camera shot such as in a dialog between two individuals. Such cut-a-ways can be considered a single scene sequence or single cut and can be consolidate in one image-processing pass.

Pans can be consolidated within a single frame visual database using special panorama stitching techniques but without lens compensation. Each frame in a pan involves:

1. The loss of some information on one side, top and/or bottom of the frame
2. Common information in the majority of the frame relative to the immediately preceding and subsequent frames and
3. New information on the other side, top and/or bottom of the frame.

By stitching these frames together based on common elements within successive frames and thereby creating a panorama of the background elements a visual database is created with all pixel offsets available for referencing in the application of a single mask overlay to the complete set of sequential frames.

Creation of a Visual Database:

Since each pixel within a single frame visual database of a background corresponds to an appropriate address within the respective "raw" (unconsolidated) frame from which it was created, any designer determined masking operation and corresponding masking lookup table designation applied to the visual database will be correctly applied to each pixel's appropriate address within the raw film frames that were used to create the single frame composite.

In this manner, sets for each scene and cut are each represented by a single frame (the visual database) in which pixels have either single or multiple representations within the series of raw frames from which they were derived. All masking within a single visual database frame will create a one-bit mask per region representation of an appropriate lookup table that corresponds to either common or unique pixel addresses within the sequential frames that created the single composite frame. These address-defined masking pixels are applied to the full resolution frames where total masking is automatically checked and adjusted where necessary using feature, edge detection and pattern recognition routines. Where adjustments are required, i.e., where applied masked region edges do not correspond to the majority of feature edges within the gray scale image, a "red flag" exception comment signals the operator that frame-by-frame adjustments may be necessary.

Single Frame Representation of Motion within Multiple Frames:

The differencing algorithm used for detecting motion objects will generally be able to differentiate dramatic pixel region changes that represent moving objects from frame to frame. In cases where cast shadows on a background from a moving object may be confused with the moving object the resulting masks will be assigned to a default alpha layer that renders that part of the moving object mask transparent. In some cases an operator using one or more vector or paint tools will designate the demarcation between the moving object and cast shadow. In most cases however, the cast shadows will be detected as an extraneous feature relative to the two key motion objects. In this invention cast shadows are handled by the background lookup table that automatically adjusts color along a luminance scale determined by the spectrum of light and dark gray scale values in the image.

Action within each frame is isolated via differencing or frame-to-frame subtraction techniques that include vector (both directional and speed) differencing (i.e., where action occurs within a pan) as well as machine vision techniques, which model objects and their behaviors. Difference pixels are then composited as a single frame (or isolated in a tiling mode) representing a multiplicity of frames thus permitting the operator to window regions of interest and otherwise direct image processing operations for computer controlled subsequent frame masking.

As with the set or background montage discussed above, action taking place in multiple frames within a scene can be represented by a single frame visual database in which each unique pixel location undergoes appropriate one bit masking from which corresponding lookup tables are applied. However, unlike the set or background montage in which all color/depth is applied and designated within the single frame pass, the purpose of creating an action composite visual data base is to window or otherwise designate each feature or region of interest that will receive a particular mask and apply region of interest vectors from one key frame element to subsequent key frame elements thus provide operator assistance to the computer processing that will track each region of interest.

During the design phase, masks are applied to designer designated regions of interest for a single instance of a motion object appearing within the background (i.e., a single frame of action appears within the background or stitched composited background in the proper x, y coordinates within the background corresponding to the single frame of action from which it was derived). Using an input device such as a mouse the operator uses the following tools in creating the regions of interest for masking. Alternatively, projects having associated computer-generated element metadata may import and if necessary, scale the metadata to the units utilized for depth in the project. Since these masks are digitally created, they can be assumed to be accurate throughout the scene and thus the outlines and depths of the computer-generated areas may be ignored for reshaping operations. Elements that border these objects, may thus be more accurately reshaped since the outlines of the computer-generated elements are taken as correct. Hence, even for computer-generated elements having the same underlying gray scale of a contiguous motion or background element, the shape of the mask at the junction can be taken to be accurate even though there is no visual difference at the junction. Again, see FIG. **37**C, step **3710** for setting mask fit location to border of CG element to potentially skip large amounts of processing in fitting masks in subsequent frames to reshape the edges to align a photographic element

1. A combination of edge detection algorithms such as standard Laplacian filters and pattern recognition routines
2. Automatic or assisted closing of a regions
3. Automatic seed fill of selected regions
4. Bimodal luminance detection for light or dark regions
5. An operator-assisted sliding scale and other tools create a "best fit" distribution index corresponding to the dynamic range of the underlying pixels as well as the underlying luminance values, pattern and weighted variables
6. Subsequent analysis of underlying gray scale, luminance, area, pattern and multiple weighting characteristics relative to immediately surrounding areas creating a unique determination/discrimination set called a Detector File.

In the pre-production key frame phase—The composited single, design motion database described above is presented along with all subsequent motion inclusive of selected key frame motion objects. All motion composites can be toggled on and off within the background or viewed in motion within the background by turning each successive motion composite on and off sequentially.

Key Frame Motion Object Creation: The operator windows all masked regions of interest on the design frame in succession and directs the computer by various pointing instruments and routines to the corresponding location (regions of interest) on selected key frame motion objects within the visual database thereby reducing the area on which the computer must operate (i.e., the operator creates a vector from the design frame moving object to each subsequent key frame moving object following a close approximation to the center of the region of interest represented within the visual database of the key frame moving object. This operator-assisted method restricts the required detection operations that must be performed by the computer in applying masks to the corresponding regions of interest in the raw frames).

In the production phase—The composited key frame motion object database described above is presented along with all subsequent motion inclusive of fully masked selected key frame motion objects. As above, all motion composites can be toggled on and off within the background or sequentially turned on and off in succession within the background to simulate actual motion. In addition, all masked regions (regions of interest) can be presented in the absence of their corresponding motion objects. In such cases the one-bit color masks are displayed as either translucent or opaque arbitrary colors.

During the production process and under operator visual control, each region of interest within subsequent motion object frames, between two key motion object frames undergoes a computer masking operation. The masking operation involves a comparison of the masks in a preceding motion object frame with the new or subsequent Detector File operation and underlying parameters (i.e., mask dimensions, gray scale values and multiple weighting factors that lie within the vector of parameters in the subsequent key frame motion object) in the successive frame. This process is aided by the windowing or pointing (using various pointing instruments) and vector application within the visual database. If the values within an operator assisted detected region of the subsequent motion object falls within the range of the corresponding region of the preceding motion object, relative to the surrounding values and if those values fall along a trajectory of values (vectors) anticipated by a comparison of the first key frame and the second key frame then the computer will determine a match and will attempt a best fit.

The uncompressed, high resolution images all reside at the server level, all subsequent masking operations on the regions of interest are displayed on the compressed composited frame

in display memory or on a tiled, compressed frame in display memory so that the operator can determine correct tracking and matching of regions. A zoomed region of interest window showing the uncompressed region is displayed on the screen to determine visually the region of interest best fit. This high-resolution window is also capable of full motion viewing so that the operator can determine whether the masking operation is accurate in motion.

In a first embodiment as shown in FIG. **1**, a plurality of feature film or television film frames **14** *a-n* representing a scene or cut in which there is a single instance or perceptive of a background **16** (FIG. **3**). In the scene **10** shown, several actors or motion elements **18'**, **18"** and **18'"** are moving within an outdoor stage and the camera is performing a pan left. FIG. **1** shows selected samples of the 120 total frames **14** making up the 5-second pan.

In FIG. **2**, an isolated background **16** processed scene from the plurality of frames **14***a-n* represented in FIG. **1** in which all motion elements **18** are removed using various subtraction and differencing techniques. The separate frames that created the pan are combined into a visual database in which unique and common pixels from each of the 120 frames **14** composing the original pan are represented in the single composite background image **12** shown in FIG. **3**. The single background image **12** is then used to create a background mask overlay **20** representing designer selected color lookup tables in which dynamic pixel colors automatically compensate or adjust for moving shadows and other changes in luminance. For depth projects, any object in the background may be assigned any depth. A variety of tools may be utilized to perform the assignment of depth information to any portion of the background including paint tools, geometric icon based tools that allow setting a contour depth to an object, or text field inputs to allow for numeric inputs. The composite background shown in FIG. **2** for example may also have a ramp function assigned to allow for a nearer depth to be assigned to the left portion of the scene and a linear increase in depth to the right of the image to be automatically assigned. See also FIGS. **42-70**.

In one illustrative embodiment of this invention, operator assisted and automated operations are used to detect obvious anchor points represented by clear edge detected intersects and other contiguous edges n each frame **14** making up the single composite image **12** and over laid mask **20**. These anchor points are also represented within the composite image **12** and are used to aide in the correct assignment of the mark to each frame **14** represented by the single composite image **12**.

Anchor points and objects and/or areas that are clearly defined by closed or nearly closed edges are designed as a single mask area and given a single lookup table. Within those clearly delineated regions polygons are created of which anchor points are dominant points. Where there is no clear edge detected to create a perfectly closed region, polygons are generated using the edge of the applied mask.

The resulting polygon mesh includes the interior of anchor point dominant regions plus all exterior areas between those regions.

Pattern parameters created by the distribution of luminance within each polygon are registered in a database for reference when corresponding polygonal addresses of the overlying masks are applied to the appropriate addresses of the frames which were used to create the composite single image **12**.

In FIG. **3**, a representative sample of each motion object (M-Object) **18** in the scene **10** receives a mask overlay that represents designer selected color lookup tables/depth assignments in which dynamic pixel colors automatically compensate or adjust for moving shadows and other changes in luminance as the M-Object **18** moves within the scene **10**. The representative sample are each considered Key M-Objects **18** that are used to define the underlying patterns, edges, grouped luminance characteristics, etc., within the masked M-Object **18**. These characteristics are used to translate the design masks from one Key M-Object **18***a* to subsequent M-Objects **18***b* along a defined vector of parameters leading to Key M-Object **18***c*, each Subsequent M-Object becoming the new Key M-Object in succession as masks are applied. As shown, Key M-Object **18***a* may be assigned a depth of 32 feet from the camera capture point while Key M-Object **18***c* may be assigned a depth of 28 feet from the camera capture point. The various depths of the object may be "tweened" between the various depth points to allow for realistic three-dimensional motion to occur within the cut without for example requiring wire frame models of all of the objects in the objects in a frame.

As with the background operations above, operator assisted and automated operations are used to detect obvious anchor points represented by clear edge detected intersects and other contiguous edges in each motion object used to create a keyframe.

Anchor points and specific regions of interest within each motion object that are clearly defined by closed or nearly closed edges are designated as a single mask area and given a single lookup table. Within those clearly delineated regions, polygons are created of which anchor points are dominant points. Where there is no clear edge detected to create a perfectly closed region, polygons are generated using the edge of the applied mask.

The resulting polygon mesh includes the interior of the anchor point dominant regions plus all exterior areas between those regions.

Pattern parameters created by the distribution of luminance values within each polygon are registered in a database for reference when corresponding polygonal addresses of the overlying masks are applied to the appropriate addresses of the frames that were used to create the composite single frame **12**.

The greater the polygon sampling the more detailed the assessment of the underlying luminance values and the more precise the fit of the overlying mask.

Subsequent or in-between motion key frame objects **18** are processed sequentially. The group of masks comprising the motion key frame object remains in its correct address location in the subsequent frame **14** or in the subsequent instance of the next motion object **18**. The mask is shown as an opaque or transparent color. An operator indicates each mask in succession with a mouse or other pointing device and along with its corresponding location in the subsequent frame and/or instance of the motion object. The computer then uses the prior anchor point and corresponding polygons representing both underlying luminance texture and mask edges to create a best fit to the subsequent instance of the motion object.

The next instance of the motion object **18** is operated upon in the same manner until all motion objects **18** in a cut **10** and/or scene are completed between key motion objects.

In FIG. **4**, all mask elements of the scene **10** are then rendered to create a fully colored and/or depth enhanced frame in which M-Object **18** masks are applied to each appropriate frame in the scene followed by the background mask **20**, which is applied only where there is no pre-existing mask in a Boolean manner. Foreground elements are then applied to each frame **14** according to a pre-programmed priority set. Aiding the accurate application of background masks **20** are vector points which are applied by the designer to the visual

database at the time of masking where there are well defined points of reference such as edges and/or distinct luminance points. These vectors create a matrix of reference points assuring accuracy of rendering masks to the separate frames that compose each scene. The applied depths of the various objects determine the amount of horizontal translation applied when generating left and right viewpoints as utilized in three-dimensional viewing as one skilled in the art will appreciate. In one or more embodiments of the invention, the desired objects may be dynamically displayed while shifting by an operator set and observe a realistic depth. In other embodiments of the invention, the depth value of an object determines the horizontal shift applied as one skilled in the art will recognize and which is taught in at least U.S. Pat. No. 6,031,564, to Ma et al., the specification of which is hereby incorporated herein by reference.

The operator employs several tools to apply masks to successive movie frames.

Display: A key frame that includes all motion objects for that frame is fully masked and loaded into the display buffer along with a plurality of subsequent frames in thumbnail format; typically 2 seconds or 48 frames.

FIGS. **5**A and **5**B show a series of sequential frames **14***a-n* loaded into display memory in which one frame **14** is fully masked with the background (key frame) and ready for mask propagation to the subsequent frames **14** via automatic mask fitting methods.

All frames **14** along with associated masks and/or applied color transforms/depth enhancements can also be displayed sequentially in real-time (24 frames/sec) using a second (child) window to determine if the automatic masking operations are working correctly. In the case of depth projects, stereoscopic glasses or red/blue anaglyph glasses may be utilized to view both viewpoints corresponding to each eye. Any type of depth viewing technology may be utilized to view depth enhanced images including video displays that require no stereoscopic glasses yet which utilizes more than two image pairs which may be created utilizing embodiments of the invention.

FIGS. **6**A and **6**B show the child window displaying an enlarged and scalable single image of the series of sequential images in display memory. The Child window enables the operator to manipulate masks interactively on a single frame or in multiple frames during real time or slowed motion.

Mask Modification: Masks can be copied to all or selected frames and automatically modified in thumbnail view or in the preview window. In the preview window mask modification takes place on either individual frames in the display or on multiple frames during real-time motion.

Propagation of Masks to Multiple Sequential Frames in Display Memory: Key Frame masks of foreground motion objects are applied to all frames in the display buffer using various copy functions:

Copy all masks in one frame to all frames;

Copy all masks in one frame to selected frames;

Copy selected mask or masks in one frame to all frames;

Copy selected mask or masks in one frame to selected frames; and

Create masks generated in one frame with immediate copy at the same addresses in all other frames.

Refining now to FIGS. **7**A and **7**B, a single mask (flesh) is propagated automatically to all frames **14** in the display memory. The operator could designate selective frames to apply the selected mask or indicate that it is applied to all frames **14**. The mask is a duplication of the initial mask in the first fully masked frame. Modifications of that mask occur only after they have been propagated.

As shown in FIG. **8**, all masks associated with the motion object are propagated to all sequential frames in display memory. The images show the displacement of the underlying image data relative to the mask information.

None of the propagation methods listed above actively fit the masks to objects in the frames **14**. They only apply the same mask shape and associated color transform information from one frame, typically the key frame to all other frames or selected frames.

Masks are adjusted to compensate for object motion in subsequent frames using various tools based on luminance, pattern and edge characteristics of the image.

Automatic Mask Fitting: Successive frames of a feature film or TV episode exhibit movement of actors and other objects. These objects are designed in a single representative frame within the current embodiment such that operator selected features or regions have unique color transformations identified by unique masks, which encompass the entire feature. The purpose of the mask-fitting tool is to provide an automated means for correct placement and reshaping of a each mask region of interest (ROI) in successive frames such that the mask accurately conforms to the correct spatial location and two dimensional geometry of the ROI as it displaces from the original position in the single representative frame. This method is intended to permit propagation of a mask region from an original reference or design frame to successive frames, and automatically enabling it to adjust shape and location to each image displacement of the associated underlying image feature. For computer-generated elements, the associated masks are digitally created and can be assumed to be accurate throughout the scene and thus the outlines and depths of the computer-generated areas may be ignored for automatic mask fitting or reshaping operations. Elements that border these objects, may thus be more accurately reshaped since the outlines of the computer-generated elements are taken as correct. Hence, even for computer-generated elements having the same underlying gray scale of a contiguous motion or background element, the shape of the mask at the junction can be taken to be accurate even though there is no visual difference at the junction. Hence, whenever automatic mask fitting of mask takes shape with a border of a computer-generated element mask, the computer-generated element mask can be utilized to define the border of the operator-defined mask as per step **3710** of FIG. **37**C. This saves processing time since automatic mask fitting in a scene with numerous computer-generated element masks can be minimized.

The method for automatically modifying both the location and correctly fitting all masks in an image to compensate for movement of the corresponding image data between frames involves the following:

Set Reference Frame Mask and Corresponding Image Data:

1. A reference frame (frame **1**) is masked by an operator using a variety of means such as paint and polygon tools so that all regions of interest (i.e., features) are tightly covered.

2. The minimum and maximum x,y coordinate values of each masked region are calculated to create rectangular bounding boxes around each masked region encompassing all underlying image pixels of each masked region.

3. A subset of pixels are identified for each region of interest within its bounding rectangle (i.e., every 10th pixel)

Copy Reference Frame Mask and Corresponding Image Data To All Subsequent Frames: The masks, bounding boxes and corresponding subset of pixel locations from the reference frame are copied over to all subsequent frames by the operator.

Approximate Offset of Regions Between Reference Frame and the Next Subsequent Frame:

1. Fast Fourier Transform (FFT) are calculated to approximate image data displacements between frame **1** and frame **2**

2. Each mask in frame **2** with the accompanying bounding boxes are moved to compensate for the displacement of corresponding image data from frame **1** using the FFT calculation.

3. The bounding box is augmented by an additional margin around the region to accommodate other motion and shape morphing effects.

Fit Masks to the New Location:

1. Using the vector of offset determined by the FFT, a gradient decent of minimum errors is calculated in the image data underlying each mask by:

2. Creating a fit box around each pixel within the subset of the bounding box

3. Calculating a weighed index of all pixels within the fit box using a bilinear interpolation method.

4. Determining offset and best fit to each subsequent frame use Gradient Decent calculations to fit the mask to the desired region.

Mask fit initialization: An operator selects image features in a single selected frame of a scene (the reference frame) and creates masks with contain all color transforms (color lookup tables) for the underlying image data for each feature. The selected image features that are identified by the operator have well-defined geometric extents which are identified by scanning the features underlying each mask for minimum and maximum x, y coordinate values, thereby defining a rectangular bounding box around each mask.

The Fit Grid used for Fit Grid Interpolation: For optimization purposes, only a sparse subset of the relevant mask-extent region pixels within each bounding box are fit with the method; this subset of pixels defines a regular grid in the image, as labeled by the light pixels of FIG. **9**A.

The "small dark" pixels shown in FIG. **9**B are used to calculate a weighed index using bilinear interpolation. The grid spacing is currently set at 10 pixels, so that essentially no more than 1 in 50 pixels are presently fit with a gradient descent search. This grid spacing could be a user controllable parameter.

Fast Fourier Transform (FFT) to Estimate Displacement Values: Masks with corresponding rectangular bounding boxes and fit grids are copied to subsequent frames. Forward and inverse FFTs are calculated between the reference frame the next subsequent frame to determine the x,y displacement values of image features corresponding to each mask and bounding box. This method generates a correlation surface, the largest value of which provides a "best fit" position for the corresponding feature's location in the search image. Each mask and bounding box is then adjusted within the second frame to the proper x,y locations.

Fit Value Calculation (Gradient Descent Search): The FFT provides a displacement vector, which directs the search for ideal mask fitting using the Gradient Descent Search method. Gradient descent search requires that the translation or offset be less than the radius of the basin surrounding the minimum of the matching error surface. A successful FFT correlation for each mask region and bounding box will create the minimum requirements.

Searching for a Best Fit on the Error Surface: An error surface calculation in the Gradient Descent Search method involves calculating mean squared differences of pixels in the square fit box centered on reference image pixel (x0, y0), between the reference image frame and the corresponding (offset) location (x, y) on the search image frame, as shown in FIGS. **10**A, B, C and D.

Corresponding pixel values in two (reference and search) fit boxes are subtracted, squared, summed/accumulated, and the square-root of the resultant sum finally divided by the number of pixels in the box (#pixels=height×width=height2) to generate the root mean square fit difference ("Error") value at the selected fit search location

$$\text{Error}(x0,y0;x,y)=\{\Sigma i\square\Sigma j\square(\text{reference box}(x0,y0)\text{pixel}[i,j]-\text{search box}(x,y)\text{pixel}[i,j])2\}/(\text{height2})$$

Fit Value Gradient: The displacement vector data derived from the FFT creates a search fit location, and the error surface calculation begins at that offset position, proceeding down (against) the gradient of the error surface to a local minimum of the surface, which is assumed to be the best fit This method finds best fit for each next frame pixel or groups of pixels based on the previous frame, using normalized squared differences, for instance in a 10×10 box and finding a minimum down the mean squared difference gradients. This technique is similar to a cross correlation but with a restricted sampling box for the calculation. In this way the corresponding fit pixel in the previous frame can be checked for its mask index, and the resulting assignment is complete.

FIGS. **11**A, B and C show a second search box derived from a descent down the error surface gradient (evaluated separately), for which the evaluated error function is reduced, possibly minimized, with respect to the original reference box (evident from visual comparison of the boxes with the reference box in FIGS. **10**A, B, C and D.).

The error surface gradient is calculated as per definition of the gradient. Vertical and horizontal error deviations are evaluated at four positions near the search box center position, and combined to provide an estimate of the error gradient for that position. The gradient component evaluation is explained with the help of FIG. **12**.

The gradient of a surface S at coordinate (x, y) is given by the directional derivatives of the surface:

$$\text{gradient}(x,y)=[dS(x,y)/dx, dS(x,y)/dy],$$

which for the discrete case of the digital image is provided by:

$$\text{gradient}(x,y)=[(\text{Error}(x+dx,y)-\text{Error}(x-dx,y))/(2*dx),\ (\text{Error}(x,y+dy)-\text{Error}(x,y-dy))/(2*dy)]$$

where dx, dy are one-half the box-width or box-height, also defined as the fit-box "box-radius": box-width=box-height=2×box-radius+1

Note that with increasing box-radius, the fit-box dimensions increase and consequently the size and detail of an image feature contained therein increase as well; the calculated fit accuracy is therefore improved with a larger box and more data to work with, but the computation time per fit (error) calculation increases as the square of the radius increase. For any computer-generated element mask area pixel that is found at a particular pixel x, y location, then that location is taken to be the edge of the overlying operated-defined mask and mask fitting continues at other pixel locations until all pixels of the mask are checked

Previous vs. Propagated Reference Images: The reference image utilized for mask fitting is usually an adjacent frame in a film image-frame sequence. However, it is sometimes preferable to use an exquisitely fit mask as a reference image (e.g. a key frame mask, or the source frame from which mask regions were propagated/copied). The present embodiment provides a switch to disable "adjacent" reference frames,

using the propagated masks of the reference image if that frame is defined by a recent propagation event.

The process of mask fitting: In the present embodiment the operator loads n frames into the display buffer. One frame includes the masks that are to be propagated and fitted to all other frames. All or some of the mask(s) are then propagated to all frames in the display buffer. Since the mask-fitting algorithm references the preceding frame or the first frame in the series for fitting masks to the subsequent frame, the first frame masks and/or preceding masks must be tightly applied to the objects and/or regions of interest. If this is not done, mask errors will accumulate and mask fitting will break down. The operator displays the subsequent frame, adjusts the sampling radius of the fit and executes a command to calculate mask fitting for the entire frame. The execution command can be a keystroke or mouse-hotkey command.

As shown in FIG. **13**, a propagated mask in the first sequential instance where there is little discrepancy between the underlying image data and the mask data. The dress mask and hand mask can be clearly seen to be off relative to the image data.

FIG. **14** shows that by using the automatic mask fitting routine, the mask data adjusts to the image data by referencing the underlying image data in the preceding image.

In FIG. **15**, the mask data in later images within the sequence show marked discrepancy relative to the underlying image data. Eye makeup, lipstick, blush, hair, face, dress and hand image data are all displaced relative to the mask data.

As shown in FIG. **16**, the mask data is adjusted automatically based on the underlying image data from the previous mask and underlying image data. In this FIG. **13**, the mask data is shown with random colors to show the regions that were adjusted automatically based on underlying pattern and luminance data. The blush and eye makeup did not have edge data to reference and were auto-adjusted on the basis of luminance and grayscale pattern.

In FIG. **17**, mask data from FIG. **16** is shown with appropriate color transforms after whole frame automatic mask fitting. The mask data is adjusted to fit the underlying luminance pattern based on data from the previous frame or from the initial key frame.

Mask Propagation With Bezier and Polygon Animation Using Edge Snap: Masks for motion objects can be animated using either Bezier curves or polygons that enclose a region of interest. A plurality of frames are loaded into display memory and either Bezier points and curves or polygon points are applied close to the region of interest where the points automatically snap to edges detected within the image data. Once the object in frame one has been enclosed by the polygon or Bezier curves the operator adjusts the polygon or Bezier in the last frame of the frames loaded in display memory. The operator then executes a fitting routine, which snaps the polygons or Bezier points plus control curves to all intermediate frames, animating the mask over all frames in display memory. The polygon and Bezier algorithms include control points for rotation, scaling and move-all to handle camera zooms, pans and complex camera moves.

In FIG. **18**, polygons are used to outline a region of interest for masking in frame one. The square polygon points snap to the edges of the object of interest. Using a Bezier curve the Bezier points snap to the object of interest and the control points/curves shape to the edges.

As disclosed in FIG. **19**, the entire polygon or Bezier curve is carried to a selected last frame in the display memory where the operator adjusts the polygon points or Bezier points and curves using the snap function, which automatically snaps the points and curves to the edges of the object of interest.

As shown in FIG. **20**, if there is a marked discrepancy between the points and curves in frames between the two frames where there was an operator interactive adjustment, the operator will further adjust a frame in the middle of the plurality of frames where there is maximum error of fit.

As shown in FIG. **21**, when it is determined that the polygons or Bezier curves are correctly animating between the two adjusted frames, the appropriate masks are applied to all frames. In these figures, the arbitrary mask color is seen filling the polygon or Bezier curves.

FIG. **22** shows the resulting masks from a polygon or Bezier animation with automatic point and curve snap to edges. The brown masks are the color transforms and the green masks are the arbitrary color masks. For depth projects, areas that have been depth assigned may be of one color while those areas that have yet to be depth assigned may be of another color for example.

Colorization/Depth Enhancement of Backgrounds in feature films and television episode: The process of applying mask information to sequential frames in a feature film or television episode is known, but is laborious for a number of reasons. In all cases, these processes involve the correction of mask information from frame to frame to compensate for the movement of underlying image data. The correction of mask information not only includes the re-masking of actors and other moving objects within a scene or cut but also correction of the background and foreground information that the moving objects occlude or expose during their movement. This has been particularly difficult in camera pans where the camera follows the action to the left, right, up or down in the scene cut. In such cases the operator must not only correct for movement of the motion object, the operator must also correct for occlusion and exposure of the background information plus correct for the exposure of new background information as the camera moves to new parts of the background and foreground. Typically these instances greatly increase the time and difficulty factor of colorizing a scene cut due to the extreme amount of manual labor involved. Embodiments of the invention include a method and process for automatically colorizing/depth enhancing a plurality of frames in scenes cuts that include complex camera movements as well as scene cuts where there is camera weave or drifting cameras movement that follows erratic action of the motion objects.

Camera Pans: For a pan camera sequence, the background associated with non-moving objects in a scene form a large part of the sequence. In order to colorize/depth enhance a large amount of background objects for a pan sequence, a mosaic that includes the background objects for an entire pan sequence with moving objects removed is created. This task is accomplished with a pan background stitcher tool. Once a background mosaic of the pan sequence is generated, it can be colorized/depth enhanced once and applied to the individual frames automatically, without having to manually colorize/depth assign the background objects in each frame of the sequence.

The pan background stitcher tool generates a background image of a pan sequence using two general operations. First, the movement of the camera is estimated by calculating the transformation needed to align each frame in the sequence with the previous frame. Since moving objects form a large portion of cinematic sequences, techniques are used that minimize the effects of moving objects on the frame registration. Second, the frames are blended into a final background mosaic by interactively selecting two pass blending regions that effectively remove moving objects from the final mosaic.

Background composite output data includes a greyscale/(or possibly color for depth projects) image file of standard

digital format such as TIFF image file (bkg.*.tif) comprised of a background image of the entire pan shot, with the desired moving objects removed, ready for color design/depth assignments using the masking operations already described, and an associated background text data file needed for background mask extraction after associated background mask/colorization/depth data components (bkg.*.msk, bkg.*lut, . . .) have been established. The background text data file provides filename, frame position within the mosaic, and other frame-dimensioning information for each constituent (input) frame associated with the background, with the following per line (per frame) content: Frame-filename, frame-x-position, frame-y-position, frame-width, frame-height, frame-left-margin-x-max, frame-right-margin-x-min. Each of the data fields are integers except for the first (frame-filename), which is a string.

Generating Transforms: In order to generate a background image for a pan camera sequence, the motion of the camera first is calculated. The motion of the camera is determined by examining the transformation needed to bring one frame into alignment with the previous frame. By calculating the movement for each pair of consecutive frames in the sequence, a map of transformations giving each frame's relative position in the sequence can be generated.

Translation Between Image Pairs: Most image registration techniques use some form of intensity correlation. Unfortunately, methods based on pixel intensities will be biased by any moving objects in the scene, making it difficult to estimate the movement due to camera motion. Feature based methods have also been used for image registration. These methods are limited by the fact that most features occur on the boundaries of moving objects, also giving inaccurate results for pure camera movement. Manually selecting feature points for a large number of frames is also too costly.

The registration method used in the pan stitcher uses properties of the Fourier transform in order to avoid bias towards moving objects in the scene. Automatic registration of frame pairs is calculated and used for the final background image assembly.

Fourier Transform of an Image Pair: The first step in the image registration process consists of taking the Fourier transform of each image. The camera motion can be estimated as a translation. The second image is translated by a certain amount given by:

$$I_2(x,y)=I_1(x-x_0,y-y_0). \tag{1}$$

Taking the Fourier transform of each image in the pair yields the following relationship:

$$F_2(\alpha,\beta)=e^{-j\cdot 2\pi\cdot(\alpha x_0-\beta y_0)}\cdot F_1(\alpha,\beta). \tag{2}$$

Phase Shift Calculation: The next step involves calculating the phase shift between the images. Doing this results in an expression for the phase shift in terms of the Fourier transform of the first and second image:

$$e^{-j\cdot 2\pi\cdot(\alpha x_0-\beta y_0)}=\frac{F_1^*\cdot F_2}{|F_1^*\cdot F_2|}. \tag{3}$$

Inverse Fourier Transform

By taking the inverse Fourier transform of the phase shift calculation given in (3) results in delta function whose peak is located at the translation of the second image.

$$\delta(x-x_0,\ y-y_0)=F^{-1}[e^{-j\cdot 2\pi\cdot(\alpha x_0-\beta y_0)}]=F^{-1}\left[\frac{F_1^*\cdot F_2}{|F_1^*\cdot F_2|}\right] \tag{4}$$

Peak Location: The two-dimensional surface that results from (4) will have a maximum peak at the translation point from the first image to the second image. By searching for the largest value in the surface, it is simple to find the transform that represents the camera movement in the scene. Although there will be spikes present due to moving objects, the dominant motion of the camera should represent the largest peak value. This calculation is performed for every consecutive pair of frames in the entire pan sequence.

Dealing with Image Noise: Unfortunately, spurious results can occur due to image noise which can drastically change the results of the transform calculation. The pan background stitcher deals with these outliers using two methods that detect and correct erroneous cases: closest peak matching and interpolated positions. If these corrections fail for a particular image pair, the stitching application has an option to manually correct the position of any pair of frames in the sequence.

Closest Matching Peak: After the transform is calculated for an image pair, the percent difference between this transform and the previous transform is determined. If the difference is higher than a predetermined threshold than a search for neighboring peaks is done. If a peak is found that is a closer match and below the difference threshold, then this value is used instead of the highest peak value.

This assumes that for a pan camera shot, the motion with be relatively steady, and the differences between motions for each frame pair will be small. This corrects for the case where image noise may cause a peak that is slightly higher that the true peak corresponding to the camera transformation.

Interpolating Positions: If the closest matching peak calculation fails to yield a reasonable result given by the percent difference threshold, then the position is estimated based on the result from the previous image pair. Again, this gives generally good results for a steady pan sequence since the difference between consecutive camera movements should be roughly the same. The peak correlation values and interpolated results are shown in the stitching application, so manual correction can be done if needed.

Generating the Background: Once the relative camera movement for each consecutive frame pair has been calculated, the frames can be composited into a mosaic which represents the entire background for the sequence. Since the moving objects in the scene need to be removed, different image blending options are used to effectively remove the dominant moving objects in the sequence.

Assembling the Background Mosaic: First a background image buffer is generated which is large enough to span the entire sequence. The background can be blended together in a single pass, or if moving objects need to be removed, a two-pass blend is used, which is detailed below. The position and width of the blend can be edited in the stitching application and can be set globally set or individually set for each frame pair. Each blend is accumulated into the final mosaic and then written out as a single image file.

Two Pass Blending: The objective in two-pass blending is to eliminate moving objects from the final blended mosaic. This can be done by first blending the frames so the moving object is completely removed from the left side of the background mosaic. An example is shown in FIG. **23**, where the character can is removed from the scene, but can still be seen

in the right side of the background mosaic. FIG. **23**. In the first pass blend shown in FIG. **23**, the moving character is shown on the stairs to the right

A second background mosaic is then generated, where the blend position and width is used so that the moving object is removed from the right side of the final background mosaic. An example of this is shown in FIG. **24**, where the character can is removed from the scene, but can still be seen the left side of the background mosaic. In the second pass blend as shown in FIG. **24**, the moving character is shown on the left.

Finally, the two-passes are blended together to generate the final blended background mosaic with the moving object removed from the scene. The final background corresponding to FIGS. **23** and **24** is shown in FIG. **25**. As shown in FIG. **25**, the final blended background with moving character is removed.

In order to facilitate effective removal of moving objects, which can occupy different areas of the frame during a pan sequence, the stitcher application has on option to interactively set the blending width and position for each pass and each frame individually or globally. An example screen shot from the blend editing tool, showing the first and second pass blend positions and widths, can be seen in FIG. **26**, which is a screen shot of the blend-editing tool.

Background Text Data Save: An output text data file containing parameter values relevant for background mask extraction as generated from the initialization phase described above. As mentioned above, each text data record includes: Frame-filename frame-x-position frame-y-position frame-width frame-height frame-left-margin-x-max frame-right-margin-x-min.

The output text data filename is composed from the first composite input frame rootname by prepending the "bkg." prefix and appending the ".txt" extension.

Example: Representative lines output text data file called "bkgA.00233.txt" that may include data from 300 or more frames making up the blended image:

4.00233.tif 0 0 1436 1080 0 1435
4.00234.tif 7 0 1436 1080 0 1435
4.00235.tif 20 0 1436 1080 0 1435
4.00236.tif 37 0 1436 1080 0 1435
4.00237.tif 58 0 1436 1080 0 1435

Image offset information used to create the composite representation of the series of frames is contained within a text file associated with the composite image and used to apply the single composite mask to all the frames used to create the composite image.

In FIG. **27**, sequential frames representing a camera pan are loaded into memory. The motion object (butler moving left to the door) has been masked with a series of color transform information leaving the background black and white with no masks or color transform information applied. Alternatively for depth projects, the motion object may be assigned a depth and/or depth shape. See FIGS. **42-70**.

In FIG. **28**, six representative sequential frames of the pan above are displayed for clarity.

FIG. **29** show the composite or montage image of the entire camera pan that was built using phase correlation techniques. The motion object (butler) included as a transparency for reference by keeping the first and last frame and averaging the phase correlation in two directions. The single montage representation of the pan is color designed using the same color transform masking techniques as used for the foreground object.

FIG. **30** shows that the sequence of frames in the camera pan after the background mask color transforms the montage has been applied to each frame used to create the montage. The mask is applied where there is no pre-existing mask thus retaining the motion object mask and color transform information while applying the background information with appropriate offsets. Alternatively for depth projects, the left and right eye views of each frame may be shown as pairs, or in a separate window for each eye for example. Furthermore, the images may be displayed on a three-dimensional viewing display as well.

In FIG. **31**, a selected sequence of frames in the pan for clarity after the color background/depth enhanced background masks have been automatically applied to the frames where there is no pre-existing masks.

Static and drifting camera shots: Objects which are not moving and changing in a film scene cut can be considered "background" objects, as opposed to moving "foreground" objects. If a camera is not moving throughout a sequence of frames, associated background objects appear to be static for the sequence duration, and can be masked and colorized only once for all associated frames. This is the "static camera" (or "static background") case, as opposed to the moving (e.g. panning) camera case, which requires stitching tool described above to generate a background composite.

Cuts or frame sequences involving little or no camera motion provide the simplest case for generating frame-image background "composites" useful for cut background colorization. However, since even a "static" camera experiences slight vibrations for a variety of reasons, the static background composition tool cannot assume perfect pixel alignment from frame-to-frame, requiring an assessment of inter-frame shifts, accurate to 1 pixel, in order to optimally associated pixels between frames prior to adding their data contribution into the composite (an averaged value). The Static Background Composite tool provides this capability, generating all the data necessary to later colorize and extract background colorization information for each of the associated frames.

Moving foreground objects such as actors, etc., are masked leaving the background and stationary foreground objects unmasked. Wherever the masked moving object exposes the background or foreground the instance of background and foreground previously occluded is copied into the single image with priority and proper offsets to compensate for movement. The offset information is included in a text file associated with the single representation of the background so that the resulting mask information can be applied to each frame in the scene cut with proper mask offsets.

Background composite output data uses a greyscale TIFF image file (bkg.*.tif) that includes averaged input background pixel values lending itself to colorization/depth enhancement, and an associated background text data file required for background mask extraction after associated background mask/colorization data/depth enhancement components (bkg.*.msk, bkg.*.lut, . . .) have been established. Background text data provides filename, mask-offset, and other frame-dimensioning information for each constituent (input) frame associated with the composite, with the following per line (per frame) format: Frame-filename frame-x-offset frame-y-offset frame-width frame-height frame-left-margin-x-max frame-right-margin-x-min. Each of these data fields are integers except for the first (frame-filename), which is a string.

Initialization: Initialization of the static background composition process involves initializing and acquiring the data necessary to create the composited background image-buffer and -data. This requires a loop over all constituent input image frames. Before any composite data initialization can occur, the composite input frames must be identified, loaded,

and have all foreground objects identified/colorized (i.e. tagged with mask labels, for exclusion from composite). These steps are not part of the static background composition procedure, but occur prior to invoking the composite tool after browsing a database or directory tree, selecting and loading relevant input frames, painting/depth assigning the foreground objects.

Get Frame Shift: Adjacent frames' image background data in a static camera cut may exhibit small mutual vertical and horizontal offsets. Taking the first frame in the sequence as a baseline, all successive frames' background images are compared to the first frames', fitting line-wise and column-wise, to generate two histograms of "measured" horizontal and vertical offsets, from all measurable image-lines and -columns. The modes of these histograms provide the most frequent (and likely) assessed frame offsets, identified and stored in arrays DVx[iframe], DVy[iframe] per frame [iframe]. These offset arrays are generated in a loop over all input frames.

Get Maximum Frame Shift: While looping over input frames during initialization to generate the DVx[ ], DVy[ ] offset array data, the absolute maximum DVxMax, DVyMax values are found from the DVx[ ], DVy[ ] values. These are required when appropriately dimensioning the resultant background composite image to accommodate all composited frames' pixels without clipping.

Get Frame Margin: While looping over input frames during initialization, an additional procedure is invoked to find the right edge of the left image margin as well as the left edge of the right image margin. As pixels in the margins have zero or near-zero values, the column indexes to these edges are found by evaluating average image-column pixel values and their variations. The edge column-indexes are stored in arrays lMarg[iframe] and rMarg[iframe] per frame [iframe], respectively.

Extend Frame Shifts with Maximum: The Frame Shifts evaluated in the GetFrameShift( ) procedure described are relative to the "baseline" first frame of a composited frame sequence, whereas the sought frame shift values are shifts/offsets relative to the resultant background composite frame. The background composite frame's dimensions equal the first composite frame's dimensions extended by vertical and horizontal margins on all sides with widths DVxMax, DVyMax pixels, respectively. Frame offsets must therefore include margin widths relative to the resultant background frame, and therefore need to be added, per iframe, to the calculated offset from the first frame:

$$DVx[\text{iframe}]=DVx[\text{iframe}]+DVx\text{Max}$$

$$DVy[\text{iframe}]=DVy[\text{iframe}]+DVy\text{Max}$$

Initialize Composite Image: An image-buffer class object instance is created for the resultant background composite. The resultant background composite has the dimensions of the first input frame increased by 2*DVxMax (horizontally) and 2*DVyMax (vertically) pixels, respectively. The first input frame background image pixels (mask-less, non-foreground pixels) are copied into the background image buffer with the appropriate frame offset. Associated pixel composite count buffer values are initialized to one (1) for pixels receiving an initialization, zero (0) otherwise. See FIG. **38**A for the flow of the processing for extracting a background, which occurs by generating a frame mask for all frames of a scene for example. FIG. **38**B illustrations the determination of the amount of Frame shift and margin that is induced for example by a camera pan. The composite image is saved after determining and overlaying the shifted images from each of the desired frames for example.

FIG. **39**A shows the edgeDetection and determination of points to snap to (1.1 and 1.2 respectively), which are detailed in FIGS. **39**B and **39**C respectively and which enable one skilled in the art to implement a image edge detection routine via Average Filter, Gradient Filter, Fill Gradient Image and a comparison with a Threshold. In addition, the GetSnapPoint routine of FIG. **39**C shows the determination of a NewPoint based on the BestSnapPoint as determined by the RangeImage less than MinDistance as shown.

FIGS. **40**A-C shows how a bimodal threshold tool is implemented in one or more embodiments of the invention. Creation of an image of light and dark cursor shape is implemented with the MakeLightShape routine wherein the light/dark values for the shape are applied with the respective routine as shown at the end of FIG. **40**A. These routines are shown in FIGS. **40**C and **40**B respectively. FIGS. **41**A-B show the calculation of FitValues and gradients for use in one or more of the above routines.

Composite Frame Loop: Input frames are composited (added) sequentially into the resultant background via a loop over the frames. Input frame background pixels are added into the background image buffer with the relevant offset (DVx [iframe], DVy[iframe]) for each frame, and associated pixel composite count values are incremented by one (1) for pixels receiving a composite addition (a separate composite count array/buffer is provided for this). Only background pixels, those without an associated input mask index, are composited (added) into the resultant background; pixels with nonzero (labeled) mask values are treated as foreground pixels and are therefore not subject to composition into the background; thus they are ignored. A status bar in the Gill is incremented per pass through the input frame loop.

Composite Finish: The final step in generating the output composite image buffer requires evaluating pixel averages which constitute the composite image. Upon completion of the composite frame loop, a background image pixel value represents the sum of all contributing aligned input frame pixels. Since resultant output pixels must be an average of these, division by a count of contributing input pixels is required. The count per pixel is provided by the associated pixel composite count buffer, as mentioned. All pixels with nonzero composite counts are averaged; other pixels remain zero.

Composite Image Save: A TIFF format output gray-scale image with 16 bits per pixel is generated from composite-averaged background image buffer. The output filename is composed from the first composite input frame filename by pre-pending the "bkg." prefix (and appending the usual ".tif" image extension if required), and writing to the associated background folder at path " . . . /Bckgrnd Frm", if available, otherwise to the default path (same as input frames').

Background Text Data Save: An output text data file containing parameter values relevant for background mask extraction as generated from the initialization phase described in (**40**A-C). As mentioned in the introduction (see FIG. **39**A), each text data record consists of: Frame-filename frame-x-offset frame-y-offset frame-width frame-height frame-left-margin-x-max frame-right-margin-x-min.

The output text data filename is composed from the first composite input frame rootname by prepending the "bkg." prefix and appending the ".txt" extension, and writing to the associated background folder at path " . . . /Bckgrnd Frm", if available, otherwise to the default path (same as input frames').

Example: A complete output text data file called "bkg.02.00.06.02.txt":

C:\NewYolder\Static_Backgrounding_Test\02.00.06.02.tif 1 4 1920 1080 0 1919

C:\New_Folder\Static_Backgrounding_Test\02.00.06.03.tif 1 4 1920 1080 0 1919

C:\New_Folder\Static_Backgrounding_Test\02.00.06.04.tif 1 3 1920 1080 0 1919

C:\New_Folder\Static_Backgrounding_Test\02.00.06.05.tif 2 3 1920 1080 0 1919

C:\New_Folder\Static_Backgrounding_Test\02.00.06.06.tif 1 3 1920 1080 0 1919

Data Cleanup: Releases memory allocated to data objects used by the static background composite procedure. These include the background composite GUI dialog object and its member arrays DVx[ ], DVy[ ], lMarg[ ], rMarg[ ], and the background composite image buffer object, whose contents have previously been saved to disk and are no longer needed.

Colorization/Depth Assignment of the Composite Background

Once the background is extracted as described above the single frame can be masked by an operator with.

The offset data for the background composite is transferred to the mask data overlaying the background such that the mask for each successive frame used to create the composite is placed appropriately.

The background mask data is applied to each successive frame wherever there are no pre-existing masks (e.g. the foreground actors).

FIG. **32** shows a sequence of frames in which all moving objects (actors) are masked with separate color transforms/depth enhancements.

FIG. **33** shows a sequence of selected frames for clarity prior to background mask information. All motion elements have been fully masked using the automatic mask-fitting algorithm.

FIG. **34** shows the stationary background and foreground information minus the previously masked moving objects. In this case, the single representation of the complete background has been masked with color transforms in a manner similar to the motion objects. Note that outlines of removed foreground objects appear truncated and unrecognizable due to their motion across the input frame sequence interval, i.e., the black objects in the frame represent areas in which the motion objects (actors in this case) never expose the background and foreground, i.e., missing background image data **3401**. The black objects are ignored for colorization-only projects during the masking operation because the resulting background mask is later applied to all frames used to create the single representation of the background only where there is no pre-existing mask. For depth related projects, the black objects where missing background image data **3401** exists, may artistically or realistically rendered, for example to fill in information to be utilized in the conversion of two-dimensional images into three-dimensional images. Since these areas are areas where pixels may not be borrowed from other frames since they are never exposed in a scene, drawing them or otherwise creating believable images there, allows for all background information to be present and used for artifact free two-dimensional to three-dimensional conversion. For example, in order to create artifact-free three-dimensional image pairs from a two-dimensional image having areas that are never exposed in a scene, backgrounds having all or enough required information for the background areas that are always occluded may be generated. The missing background image data **3401** may be painted, drawn, created, computer-generated or otherwise obtained from a studio for example, so that there is enough information in a background, including the black areas to translate foreground objects horizontally and borrow generated background data for the translated edges for occluded areas. This enables the generation of artifact free three-dimensional image pairs since translation of foreground objects horizontally, which may expose areas that are always occluded in a scene, results in the use of the newly created background data instead of stretching objects or morphing pixels which creates artifacts that are human detectable errors. Hence, obtaining backgrounds with occluded areas filled in, either partially with enough horizontal realistic image data or fully with all occluded areas rendered into a realistic enough looking area, i.e., drawn and colorized and/or depth assigned, thus results in artifact free edges for depth enhanced frames. See also FIGS. **70** and **71-76** and the associated description respectively. Generation of missing background data may also be utilized to create artifact free edges along computer-generated elements as well.

FIG. **35** shows the sequential frames in the static camera scene cut after the background mask information has been applied to each frame with appropriate offsets and where there is no pre-existing mask information.

FIG. **36** shows a representative sample of frames from the static camera scene cut after the background information has been applied with appropriate offsets and where there is no pre-existing mask information.

Colorization Rendering: After color processing is completed for each scene, subsequent or sequential color motion masks and related lookup tables are combined within 24-bit or 48-bit RGB color space and rendered as TIF or TGA files. These uncompressed, high-resolution images are then rendered to various media such as HDTV, 35 mm negative film (via digital film scanner), or a variety of other standard and non standard video and film formats for viewing and exhibit.

Process Flow:

Digitization, Stabilization and Noise Reduction:

1. 35 mm film is digitized to 1920×1080×10 in any one of several digital formats.

2. Each frame undergoes standard stabilization techniques to minimize natural weaving motion inherent in film as it traverses camera sprockets as well as any appropriate digital telecine technology employed. Frame-differencing techniques are also employed to further stabilize image flow.

3. Each frame then undergoes noise reduction to minimize random film grain and electronic noise that may have entered into the capture process.

Pre-Production Movie Dissection into Camera Elements and Visual Database Creation:

1. Each scene of the movie is broken down into background and foreground elements as well as movement objects using various subtraction, phase correlation and focal length estimation algorithms. Background and foreground elements may include computer-generated elements or elements that exist in the original movie footage for example.

2. Backgrounds and foreground elements m pans are combined into a single frame using uncompensated (lens) stitching routines.

3. Foregrounds are defined as any object and/or region that move in the same direction as the background but may represent a faster vector because of its proximity to the camera lens. In this method pans are reduced to a single representative image, which contains all of the background and foreground information taken from a plurality of frames.

4. Zooms are sometimes handled as a tiled database in which a matrix is applied to key frames where vector points of reference correspond to feature points in the image and cor-

respond to feature points on the applied mask on the composited mask encompassing any distortion.

5. A database is created from the frames making up the single representative or composited frame (i.e., each common and novel pixel during a pan is assigned to the plurality of frames from which they were derived or which they have in common).

6. In this manner, a mask overlay representing an underlying lookup table will be correctly assigned to the respective novel and common pixel representations of backgrounds and foregrounds in corresponding frames.

Pre-Production Design Background Design:

1. Each entire background is colorized/depth assigned as a single frame in which all motion objects are removed. Background masking is accomplished using a routine that employs standard paint, fill, digital airbrushing, transparency, texture mapping, and similar tools. Color selection is accomplished using a 24-bit color lookup table automatically adjusted to match the density of the underlying gray scale and luminance. Depth assignment is accomplished via assigning depths, assigning geometric shapes, entry of numeric values with respect to objects, or in any other manner in the single composite frame. In this way creatively selected colors/depths are applied that are appropriate for mapping to the range of gray scale/depth underlying each mask. The standard color wheel used to select color ranges detects the underlying grayscale dynamic range and determines the corresponding color range from which the designer may choose (i.e., only from those color saturations that will match the grayscale luminance underlying the mask.)

2. Each lookup table allows for a multiplicity of colors applied to the range of gray scale values underlying the mask. The assigned colors will automatically adjust according to luminance and/or according to pre-selected color vectors compensating for changes in the underlying gray scale density and luminance.

Pre-Production Design Motion Element Design:

1. Design motion object frames are created which include the entire scene background as well as a single representative moment of movement within the scene in which all characters and elements within the scene are present. These moving non-background elements are called Design Frame Objects (DFO).

2. Each DFO is broken down into design regions of interest (regions of interest) with special attention focused on contrasting elements within the DFOs that can be readily be isolated using various gray scale and luminance analyses such as pattern recognition and or edge detection routines. As existing color movies may be utilized for depth enhancement, regions of interest may be picked with color taken into account.

3. The underlying gray scale- and luminance distribution of each masked region is displayed graphically as well as other gray scale analyses including pattern analysis together with a graphical representation of the region's shape with area, perimeter and various weighting parameters.

4. Color selection is determined for each region of interest comprising each object based on appropriate research into the film genre, period, creative intention, etc. and using a 24 bit color lookup table automatically adjusted to match the density of the underlying gray scale and luminance suitable and creatively selected colors are applied. The standard color wheel detects the underlying grayscale range and restricts the designer to choose only from those color saturations that will match the grayscale luminance underlying the mask. Depth assignments may be made or adjusted for depth projects until realistic depth is obtained for example.

5. This process continues until a reference design mask is created for all objects that move in the scene.

Pre-Production Design Key Frame Objects Assistant Designer:

1. Once all color selection/depth assignment is generally completed for a particular scene the design motion object frame is then used as a reference to create the larger number of key frame objects within the scene.

2. Key Frame Objects (all moving elements within the scene such as people, cars, etc that do not include background elements) are selected for masking.

3. The determining factor for each successive key frame object is the amount of new information between one key frame and the next key frame object.

Method of Colorizing/Depth Enhancing Motion Elements in Successive Frames:

1. The Production Colorist (operator) loads a plurality of frames into the display buffer.

2. One of the frames in the display buffer will include a key frame from which the operator obtains all masking information. The operator makes no creative or color/depth decisions since all color transform information is encoded within the key frame masks.

3. The operator can toggle from the colorized or applied lookup tables to translucent masks differentiated by arbitrary but highly contrasting colors.

4. The operator can view the motion of all frames in the display buffer observing the motion that occurs in successive frames or they can step through the motion from one key frame to the next.

5. The operator propagates (copies) the key frame mask information to all frames in the display buffer.

6. The operator then executes the mask fitting routine on each frame successively. FIG. **37**A shows the mask fitting generally processing flow chart that is broken into subsequent detailed flow charts **37**B and **37**C. The program makes a best fit based on the grayscale/luminance, edge parameters and pattern recognition based on the gray scale and luminance pattern of the key frame or the previous frame in the display. For computer-generated elements, the mask fitting routines are skipped since the masks or alphas define digitally created (and hence non-operator-defined) edges that accurately define the computer-generated element boundaries. Mask fitting operations take into account the computer-generated element masks or alphas and stop when hitting the edge of a computer-generated element mask since these boundaries are accepted as accurate irrespective of grey-scale as per step **3710** of FIG. **37**C. This enhances the accuracy of mask edges and reshapes when colors of a computer-generated element and operator-defined mask are of the same base luminance for example. As shown in FIG. **37**A, the Mask Fit initializes the region and fit grid parameters, then calls the Calculate fit grid routine and then the Interpolate mask on fit grid routine, which execute on any computer as described herein, wherein the routines are specifically configured to calculate fit grids as specified in FIGS. **37**B and **37**C. The flow of processing of FIG. **37**B from the Initialize region routine, to the initialization of image line and image column and reference image flows into the CalculateFitValue routine which calls the fit gradient routine which in turn calculates xx, and yy as the difference between the xfit, yfit and gradients for x and y. If the FitValue is greater than the fit, for x, y and xx and yy, then the xfit and yfit values are stored in the FitGrid. Otherwise, processing continues back at the fit gradient routine with new values for xfit and yfit. When the processing for the size of the Grid is complete for x and y, then the mask is interpolated as per FIG. **37**C. After initialization, the indices i and j for the

FitGridCell are determined and a bilinear interpolation is performed at the fitGridA-D locations wherein the Mask is fit up to any border found for any CG element at **3710** (i.e., for a known alpha border or border with depth values for example that define a digitally rendered element that is taken as a certified correct mask border). The mask fitting interpolation is continued up to the size of the mask defined by xend and yend.

7. In the event that movement creates large deviations in regions from one frame to the next the operator can select individual regions to mask-fit. The displaced region is moved to the approximate location of the region of interest where the program attempts to create a best fit. This routine continues for each region of interest in succession until all masked regions have been applied to motion objects in all sequential frames in the display memory.

a. The operator clicks on a single mask in each successive frame on the corresponding area where it belongs in frame **2**. The computer makes a best fit based on the grayscale/luminance, edge parameters, gray scale pattern and other analysis.

b. This routine continues for each region in succession until all regions of interest have been repositioned in frame two.

c. The operator then indicates completion with a mouse click and masks in frame two are compared with gray scale parameters in frame three.

d. This operation continues until all motion in all frames between two or more key frames is completely masked.

8. Where there is an occlusion, a modified best-fit parameter is used. Once the occlusion is passed, the operator uses the pre-occlusion frame as a reference for the post occlusion frames.

9. After all motion is completed, the background/set mask is applied to each frame in succession. Application is: apply mask where no mask exists.

10. Masks for motion objects can also be animated using either Bezier curves or polygons that enclose a region of interest.

a. A plurality of frames are loaded into display memory and either Bezier points and curves of polygon points are applied close to the region of interest where the points automatically snap to edges detected within the image data.

b. Once the object in frame one has been enclosed by the polygon or Bezier curves the operator adjusts the polygon or Bezier in the last frame of the frames loaded in display memory.

c. The operator then executes a fitting routine, which snaps the polygons or Bezier points plus control curves to all intem1ediate frames, animating the mask over all frames in display memory.

d. The polygon and Bezier algorithms include control points for rotation, scaling and move-all to handle zooms, pans and complex camera moves where necessary.

FIG. **42** shows two image frames that are separated in time by several frames, of a person levitating a crystal ball wherein the various objects in the image frames are to be converted from two-dimensional objects to three-dimensional objects. As shown the crystal ball moves with respect to the first frame (shown on top) by the time that the second frame (shown on the bottom) occurs. As the frames are associated with one another, although separated in time, much of the masking information can be utilized for both frames, as reshaped using embodiments of the invention previously described above. For example, using the mask reshaping techniques described above for colorization, i.e., using the underlying grey-scale for tracking and reshaping masks, much of the labor involved with converting a two-dimensional movie to a three-dimensional movie is eliminated. This is due to the fact that once key frames have color or depth information applied to them, the mask information can be propagated automatically throughout a sequence of frames which eliminates the need to adjust wire frame models for example. Although there are only two images shown for brevity, these images are separated by several other images in time as the crystal ball slowly moves to the right in the sequence of images.

FIG. **43** shows the masking of the first object in the first image frame that is to be converted from a two-dimensional image to a three-dimensional image. In this figure, the first object masked is the crystal ball. There is no requirement to mask objects in any order. In this case a simple free form drawing tool is utilized to apply a somewhat round mask to the crystal ball. Alternatively, a circle mask may be dropped on the image and resized and translated to the correct position to correspond to the round crystal ball. However, since most objects masked are not simple geometric shapes, the alternative approach is shown herein. The grey-scale values of the masked object are thus utilized to reshape the mask in subsequent frames.

FIG. **44** shows the masking of the second object in the first image frame. In this figure, the hair and face of the person behind the crystal ball are masked as the second object using a free form drawing tool. Edge detection or grey-scale thresholds can be utilized to accurately set the edges of the masks as has been previously described above with respect to colorization. There is no requirement that an object be a single object, i.e., the hair and face of a person can be masked as a single item, or not and depth can thus be assigned to both or individually as desired.

FIG. **45** shows the two masks in color in the first image frame allowing for the portions associated with the masks to be viewed. This figure shows the masks as colored transparent masks so that the masks can be adjusted if desired.

FIG. **46** shows the masking of the third object in the first image frame. In this figure the hand is chosen as the third object. A free form tool is utilized to define the shape of the mask.

FIG. **47** shows the three masks in color in the first image frame allowing for the portions associated with the masks to be viewed. Again, the masks can be adjusted if desired based on the transparent masks.

FIG. **48** shows the masking of the fourth object in the first image frame. As shown the person's jacket form the fourth object.

FIG. **49** shows the masking of the fifth object in the first image frame. As shown the person's sleeve forms the fifth object.

FIG. **50** shows a control panel for the creation of three-dimensional images, including the association of layers and three-dimensional objects to masks within an image frame, specifically showing the creation of a Plane layer for the sleeve of the person in the image. On the right side of the screendump, the "Rotate" button is enabled, shown a "Translate Z" rotation quantity showing that the sleeve is rotated forward as is shown in the next figure.

FIG. **51** shows a three-dimensional view of the various masks shown in FIGS. **43-49**, wherein the mask associated with the sleeve of the person is shown as a Plane layer that is rotated toward the left and right viewpoints on the right of the page. Also, as is shown the masks associated with the jacket and person's face have been assigned a Z-dimension or depth that is in front of the background.

FIG. **52** shows a slightly rotated view of FIG. **51**. This figure shows the Plane layer with the rotated sleeve tilted toward the viewpoints. The crystal ball is shown as a flat

object, still in two-dimensions as it has not yet been assigned a three-dimensional object type.

FIG. **53** shows a slightly rotated view of FIGS. **51** (and **52**), wherein the sleeve is shown tilting forward, again without ever defining a wire frame model for the sleeve. Alternatively, a three-dimensional object type of column can be applied to the sleeve to make an even more realistically three-dimensional shaped object. The Plane type is shown here for brevity.

FIG. **54** shows a control panel specifically showing the creation of a sphere object for the crystal ball in front of the person in the image. In this figure, the Sphere three-dimensional object is created and dropped into the three-dimensional image by clicking the "create selected" button in the middle of the frame, which is then shown (after translation and resizing onto the crystal ball in the next figure).

FIG. **55** shows the application of the sphere object to the flat mask of the crystal ball, that is shown within the sphere and as projected to the front and back of the sphere to show the depth assigned to the crystal ball. The Sphere object can be translated, i.e., moved in three axis, and resized to fit the object that it is associated with. The projection of the crystal ball onto the sphere shows that the Sphere object is slightly larger than the crystal ball, however this ensures that the full crystal ball pixels are assigned depths. The Sphere object can be resized to the actual size of the sphere as well for more refined work projects as desired.

FIG. **56** shows a top view of the three-dimensional representation of the first image frame showing the Z-dimension assigned to the crystal ball shows that the crystal ball is in front of the person in the scene.

FIG. **57** shows that the sleeve plane rotating in the X-axis to make the sleeve appear to be coming out of the image more. The circle with a line (X axis line) projecting through it defines the plane of rotation of the three-dimensional object, here a plane associated with the sleeve mask.

FIG. **58** shows a control panel specifically showing the creation of a Head object for application to the person's face in the image, i.e., to give the person's face realistic depth without requiring a wire model for example. The Head object is created using the "Created Selected" button in the middle of the screen and is shown in the next figure.

FIG. **59** shows the Head object in the three-dimensional view, too large and not aligned with the actual person's head. After creating the Head object as per FIG. **58**, the Head object shows up in the three-dimensional view as a generic depth primitive that is applicable to heads in general. This is due to the fact that depth information is not exactly required for the human eye. Hence, in depth assignments, generic depth primitives may be utilized in order to eliminate the need for three-dimensional wire frames. The Head object is translated, rotated and resized in subsequent figures as detailed below.

FIG. **60** shows the Head object in the three-dimensional view, resized to fit the person's face and aligned, e.g., translated to the position of the actual person's head.

FIG. **61** shows the Head object in the three-dimensional view, with the Y-axis rotation shown by the circle and Y-axis originating from the person's head thus allowing for the correct rotation of the Head object to correspond to the orientation of the person's face.

FIG. **62** shows the Head object also rotated slightly clockwise, about the Z-axis to correspond to the person's slightly tilted head. The mask shows that the face does not have to be exactly lined up for the result three-dimensional image to be believable to the human eye. More exacting rotation and resizing can be utilized where desired.

FIG. **63** shows the propagation of the masks into the second and final image frame. All of the methods previously disclosed above for moving masks and reshaping them are applied not only to colorization but to depth enhancement as well. Once the masks are propagated into another frame, all frames between the two frames may thus be tweened. By tweening the frames, the depth information (and color information if not a color movie) are thus applied to non-key frames.

FIG. **64** shows the original position of the mask corresponding to the person's hand.

FIG. **65** shows the reshaping of the mask, that is performed automatically and with can be adjusted in key frames manually if desired, wherein any intermediate frames get the tweened depth information between the first image frame masks and the second image frame masks. The automatic tracking of masks and reshaping of the masks allows for great savings in labor. Allowing manual refinement of the masks allows for precision work where desired.

FIG. **66** shows the missing information for the left viewpoint as highlighted in color on the left side of the masked objects in the lower image when the foreground object, here a crystal ball is translated to the right. In generating the left viewpoint of the three-dimensional image, the highlighted data must be generated to fill the missing information from that viewpoint.

FIG. **67** shows the missing information for the right viewpoint as highlighted in color on the right side of the masked objects in the lower image when the foreground object, here a crystal ball is translated to the left. In generating the right viewpoint of the three-dimensional image, the highlighted data must be generated to fill the missing information from that viewpoint. Alternatively, a single camera viewpoint may be offset from the viewpoint of the original camera, however the missing data is large for the new viewpoint. This may be utilized if there are a large number of frames and some of the missing information is found in adjacent frames for example.

FIG. **68** shows an anaglyph of the final depth enhanced first image frame viewable with Red/Blue 3-D glasses. The original two-dimensional image is now shown in three-dimensions.

FIG. **69** shows an anaglyph of the final depth enhanced second and last image frame viewable with Red/Blue 3-D glasses, note rotation of person's head, movement of person's hand and movement of crystal ball. The original two-dimensional image is now shown in three-dimensions as the masks have been moved/reshaped using the mask tracking/reshaping as described above and applying depth information to the masks in this subsequent frame from an image sequence. As described above, the operations for applying the depth parameter to a subsequent frame is performed using a general purpose computer having a central processing unit (CPU), memory, bus situated between the CPU and memory for example specifically programmed to do so wherein figures herein which show computer screen displays are meant to represent such a computer.

FIG. **70** shows the right side of the crystal ball with fill mode "smear", wherein the pixels with missing information for the left viewpoint, i.e., on the right side of the crystal ball are taken from the right edge of the missing image pixels and "smeared" horizontally to cover the missing information. Any other method for introducing data into hidden areas is in keeping with the spirit of the invention. Stretching or smearing pixels where missing information is creates artifacts that are recognizable to human observers as errors. By obtaining or otherwise creating realistic data for the missing information is, i.e., for example via a generated background with missing information filled in, methods of filling missing data can be avoided and artifacts are thus eliminated. For example,

providing a composite background or frame with all missing information designated in a way that an artist can use to create a plausible drawing or painting of a missing area is one method of obtaining missing information for use in two-dimensional to three-dimensional conversion projects.

FIG. **71** shows a mask or alpha plane for a given frame of a scene, for an actor's upper torso and head **7101**, and transparent wings **7102**. The mask may include opaque areas shown as black and transparent areas that are shown as grey areas. The alpha plane may be generated for example as an 8 bit grey-scale "OR" of all foreground masks. Any other method of generating a foreground mask having motion objects or foreground object related masks defined is in keeping with the spirit of the invention.

FIG. **72** shows an occluded area, i.e., missing background image data **7201** as a colored sub-area of the actor of FIG. **71** that never uncovers the underlying background, i.e., where missing information in the background for a scene or frame occurs. This area is the area of the background that is never exposed in any frame in a scene and hence cannot be borrowed from another frame. When for example generating a composite background, any background pixel not covered by a motion object mask or foreground mask can have a simple Boolean TRUE value, all other pixels are thus the occluded pixels as is also shown in FIG. **34**.

FIG. **73** shows the occluded area of FIG. **72** with generated data **7201***a* for missing background image data that is artistically drawn or otherwise rendered to generate a complete and realistic background for use in artifact free two-dimensional to three-dimensional conversion. See also FIG. **34** and the description thereof. As shown, FIG. **73** also has masks drawn on background objects, which are shown in colors that differ from the source image. This allows for colorization or colorization modifications for example as desired.

FIG. **73**A shows the occluded area with missing background image data **7201***b* partially drawn or otherwise rendered to generate just enough of a realistic looking background for use in artifact free two-dimensional to three-dimensional conversion. An artist in this example may draw narrower versions of the occluded areas, so that offsets to foreground objects would have enough realistic background to work with when projecting a second view, i.e., translating a foreground object horizontally which exposes occluded areas. In other words, the edges of the missing background image data area may be drawn horizontally inward by enough to allow for some of the generated data to be used, or all of the generated data to be used in generating a second viewpoint for a three-dimensional image set.

In one or more embodiments of the invention, a number of scenes from a movie may be generated for example by computer drawing by artists or sent to artists for completion of backgrounds. In one or more embodiments, a website may be created for artists to bid on background completion projects wherein the website is hosted on a computer system connected for example to the Internet. Any other method for obtaining backgrounds with enough information to render a two-dimensional frame into a three-dimensional pair of viewpoints is in keeping with the spirit of the invention, including rendering a full background with realistic data for all of the occluded area of FIG. **72** (which is shown in FIG. **73**) or only a portion of the edges of the occluded area of FIG. **72**, (which is shown as FIG. **73**A). By estimating a background depth and a depth to a foreground object and knowing the offset distance desired for two viewpoints, it is thus possible to obtain less than the whole occluded area for use in artifact free two-dimensional to three-dimensional conversion. In one or more embodiments, a fixed offset, e.g., 100 pixels on each edge of each occluded area, or a percentage of the size of the foreground object, i.e., 5% for example, may flagged to be created and if more data is needed, then the frame is flagged for updating, or smearing or pixel stretching may be utilized to minimize the artifacts of missing data.

FIG. **74** shows a light area of the shoulder portion on the right side of FIG. **71**, where missing background image data **7201** exists when generating a right viewpoint for a right image of a three-dimensional image pair. Missing background image data **7201** represents a gap where stretching (as is also shown in FIG. **70**) or other artifact producing techniques would be used when shifting the foreground object to the left to create a right viewpoint. The dark portion of the figure is taken from the background where data is available in at least one frame of a scene.

FIG. **75** shows an example of the stretching of pixels, or "smeared pixels" **7201***c*, corresponding to the light area in FIG. **74**, i.e., missing background image data **7201**, wherein the pixels are created without the use of a generated background, i.e., if no background data is available for an area that is occluded in all frames of a scene.

FIG. **76** shows a result of a right viewpoint without artifacts on the edge of the shoulder of the person through use of generated data **7201***a* (or **7201***b*) for missing background image data **7201** shown as for always-occluded areas of a scene.

FIG. **77** shows an example of a computer-generated element, here robot **7701**, which is modeled in three-dimensional space and projected as a two-dimensional image. The background is grey to signify invisible areas. As is shown in the following figures, metadata such as alpha, mask, depth or any combination thereof is utilized to speed the conversion process from two-dimensional image to a pair of two-dimensional images for left and right eye for three-dimensional viewing. Masking this character by hand, or even in a computer-aided manner by an operator is extremely time consuming since there are literally hundreds if not thousands of sub-masks required to render depth (and/or color) correctly to this complex object.

FIG. **78** shows an original image separated into background **7801** and foreground elements **7802** and **7803**, (mountain and sky in the background and soldiers in the bottom left also see FIG. **79**) along with the imported color and depth of the computer-generated element, i.e., robot **7803** with depth automatically set via the imported depth metadata. Although the soldiers exist in the original image, their depths are set by an operator, and generally shapes or masks with varying depths are applied at these depths with respect to the original objects to obtain a pair of stereo images for left and right eye viewing. (See FIG. **79**). As shown in the background, any area that is covered for the scene such as outline **7804** (of a soldier's head projected onto the background) can be artistically rendered for example to provide believable missing data, as is shown in FIG. **73** based on the missing data of FIG. **73**A, which results in artifact free edges as shown in FIG. **76** for example. Importing data for computer generated elements may include reading a file that has depth information on a pixel-by-pixel basis for computer-generated element **7701** and displaying that information in a perspective view on a computer display as an imported element, e.g., robot **7803**. This import process saves enormous amounts of operator time and makes conversion of a two-dimensional movie into a three-dimensional movie economically viable. One or more embodiments of the invention store the masks and imported data in computer memory and/or computer disk drives for use by one or more computers in the conversion process.

FIG. **79** shows mask **7901** (forming a portion of the helmet of the rightmost soldier) associated with the photograph of soldiers **7802** in the foreground. Mask **7901** along with all other operated-defined masks shown in multiple artificial colors on the soldiers, to apply depth to the various portions of the soldiers occurring in the original image that lie in depth in front of the computer-generated element, i.e., robot **7803**. The dashed lines horizontally extending from the mask areas **7902** and **7903** show horizontal translation of the foreground objects takes place and where imported metadata can be utilized to accurately auto-correct over-painting of depth or color on the masked objects when metadata exists for the other elements of a movie. For example, when an alpha exists for the objects that occur in front of the computer-generated elements, the edges can be accurately determined. One type of file that can be utilized to obtain mask edge data is a file with alpha file and/or mask data such as an RGBA file. (See FIG. **80**). In addition, use of generated data for missing areas of the background at these horizontally translated mask areas **7902** and **7903** enables artifact free two-dimensional to three-dimensional conversion.

FIG. **80** shows an imported alpha layer **8001** shown as a dark blue overlay, which can also be utilized as a mask layer to limit the operator defined, and potentially less accurate masks used for applying depth to the edges of the three soldiers **7802** and designated as soldiers A, B and C. In addition, an optional computer-generated element, such as dust can be inserted into the scene along the line annotated as "DUST", to augment the reality of the scene if desired. Any of the background, foreground or computer-generated elements can be utilized to fill portions of the final left and right image pairs as is required.

FIG. **81** shows the result of using the operator-defined masks without adjustment when overlaying a motion element such as the soldier on the computer-generated element such as the robot. Without the use of metadata associated with the original image objects, such as matte or alpha **8001**, artifacts occur where operator-defined masks do not exactly align with the edges of the masked objects. In the topmost picture, the soldier's lips show a light colored edge **8101** while the lower picture shows an artifact free edge since the alpha of FIG. **80** is used to limit the edges of any operator-defined masks. Through use of the alpha metadata of FIG. **80** applied to the operated-defined mask edges of FIG. **79**, artifact free edges on the overlapping areas is thus enabled. As one skilled in the art will appreciate, application of successively nearer elements combined with their alphas is used to layer all of the objects at their various depths from back to front to create a final image pair for left eye and right eye viewing.

Embodiments of the invention enable real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts and to minimize or eliminate iterative workflow paths back through different workgroups by generating translation files that can be utilized as portable pixel-wise editing files. For example, a mask group takes source images and creates masks for items, areas or human recognizable objects in each frame of a sequence of images that make up a movie. The depth augmentation group applies depths, and for example shapes, to the masks created by the mask group. When rendering an image pair, left and right viewpoint images and left and right translation files may be generated by one or more embodiments of the invention. The left and right viewpoint images allow 3D viewing of the original 2D image. The translation files specify the pixel offsets for each source pixel in the original 2D image, for example in the form of UV or U maps. These files are generally related to an alpha mask for each layer, for example a layer for an actress, a layer for a door, a layer for a background, etc. These translation files, or maps are passed from the depth augmentation group that renders 3D images, to the quality assurance workgroup. This allows the quality assurance workgroup (or other workgroup such as the depth augmentation group) to perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts such as masking errors without delays associated with processing time/re-rendering and/or iterative workflow that requires such re-rendering or sending the masks back to the mask group for rework, wherein the mask group may be in a third world country with unskilled labor on the other side of the globe. In addition, when rendering the left and right images, i.e., 3D images, the Z depth of regions within the image, such as actors for example, may also be passed along with the alpha mask to the quality assurance group, who may then adjust depth as well without re-rendering with the original rendering software. This may be performed for example with generated missing background data from any layer so as to allow "downstream" real-time editing without re-rendering or ray-tracing for example. Quality assurance may give feedback to the masking group or depth augmentation group for individuals so that these individuals may be instructed to produce work product as desired for the given project, without waiting for, or requiring the upstream groups to rework anything for the current project. This allows for feedback yet eliminates iterative delays involved with sending work product back for rework and the associated delay for waiting for the reworked work product. Elimination of iterations such as this provide a huge savings in wall-time, or end-to-end time that a conversion project takes, thereby increasing profits and minimizing the workforce needed to implement the workflow.

FIG. **82** shows a source image to be depth enhanced and provided along with left and right translation files (see FIGS. **85**A-D and **86**A-D for embodiments of translation files) and alpha masks (such as shown in FIG. **79**) to enable real-time editing of 3D images without re-rendering or ray-tracing the entire image sequence in a scene (e.g., by downstream workgroups) for example to alter layers/colors/masks and/or remove and/or or adjust depths or otherwise change the 3D images without iterative workflow paths back to the original workgroups (as per FIG. **96** versus FIG. **95**).

FIG. **83** shows masks generated by the mask workgroup for the application of depth by the depth augmentation group, wherein the masks are associated with objects, such as for example human recognizable objects in the source image of FIG. **82**. Generally, unskilled labor is utilized to mask human recognizable objects in key frames within a scene or sequence of images. The unskilled labor is cheap and generally located offshore. Hundreds of workers may be hired at low prices to perform this tedious work associated with masking Any existing colorization masks may be utilized as a starting point for 3D masks, which may be combined to form a 3D mask outline that is broken into sub-masks that define differing depths within a human recognizable object. Any other method of obtaining masks for areas of an image are in keeping with the spirit of the invention.

FIG. **84** shows areas where depth is applied generally as darker for nearer objects and lighter for objects that are further away. This view gives a quick overview of the relative depths of objects in a frame.

FIG. **85**A shows a left UV map containing translations or offsets in the horizontal direction for each source pixel. When rendering a scene with depths applied, translation maps that map the offsets of horizontal movement of individual pixels in a graphical manner may be utilized. FIG. **85**B shows a right

UV map containing translations or offsets in the horizontal direction for each source pixel. Since each of these images looks the same, it is easier to observe that there are subtle differences in the two files by shifting the black value of the color, so as to highlight the differences in a particular area of FIGS. **85**A and **85**B. FIG. **85**C shows a black value shifted portion of the left UV map of FIG. **85**A to show the subtle contents therein. This area corresponds to the tree branches shown in the upper right corner of FIGS. **82**, **83** and **84** just above the cement mixer truck and to the left of the light pole. FIG. **85**D shows a black value shifted portion of the right UV map of FIG. **85**B to show the subtle contents therein. The branches shown in the slight variances of color signify that those pixels would be shifted to the corresponding location in a pure UV map that maps Red from darkest to lightest in the horizontal direction and maps Green from darkest to lightest in the vertical direction. In other words, the translation map in the UV embodiment is a graphical depiction of the shifting that occurs when generating a left and right viewpoint with respect to the original source image. UV maps may be utilized, however, any other file type that contains horizontal offsets from a source image on a pixel-by-pixel basis (or finer grained) may be utilized, including compressed formats that are not readily viewable as images. Some software packages for editing come with pre-built UV widgets, and hence, UV translation files or maps can therefore be utilized if desired. For example, certain compositing programs have pre-built objects that enable UV maps to be readily utilized and otherwise manipulated graphically and hence for these implementations, graphically viewable files may be utilized, but are not required.

Since creation of a left and right viewpoint from a 2D image uses horizontal shifts, it is possible to use a single color for the translation file. For example, since each row of the translation file is already indexed in a vertical direction based on the location in memory, it is possible to simply use one increasing color, for example Red in the horizontal direction to signify an original location of a pixel. Hence, any shift of pixels in the translation map are shown as shifts of a given pixel value from one horizontal offset to another, which makes for subtle color changes when the shifts are small, for example in the background. FIG. **86**A shows a left U map containing translations or offsets in the horizontal direction for each source pixel. FIG. **86**B shows a right U map containing translations or offsets in the horizontal direction for each source pixel. FIG. **86**C shows a black value shifted portion of the left U map of FIG. **86**A to show the subtle contents therein. FIG. **86**D shows a black value shifted portion of the right U map of FIG. **86**B to show the subtle contents therein. Again there is no requirement that a humanly viewable file format be utilized, and any format that stores horizontal offsets on a pixel-by-pixel basis relative to a source image may be utilized. Since memory and storage is so cheap, any format whether compressed or not may be utilized without any significant increase in cost however. Generally, creation of a right eye image makes foreground portions of the U map (or UV map) appear darker since they are shifting left and visa versa. This is easy to observe by looking at something in the foreground with only the right eye open and then moving slightly to the right (to observe that the foreground object has indeed been shifted to the left). Since the U map (or UV map) in the unaltered state is a simple ramp of color from dark to light, it then follows that shifting something to the left, i.e., for the right viewpoint, maps it to a darker area of the U map (or UV map). Hence the same tree branches in the same area of each U map (or UV map) are darker for the right eye and brighter for the left eye with respect to un-shifted pixels. Again, use of a viewable map is not required, but shows the concept of shifting that occurs for a given viewpoint.

FIG. **87** shows known uses for UV maps, wherein a three-dimensional model is unfolded so that an image in UV space can be painted onto the 3D model using the UV map. This figure shows how UV maps have traditionally been utilized to apply a texture map to a 3D shape. For example, the texture, here a painting or flat set of captured images of the Earth is mapped to a U and V coordinate system, that is translated to an X, Y and Z coordinate on the 3D model. Traditional animation has been performed in this manner in that wire frame models are unraveled and flattened, which defines the U and V coordinate system in which to apply a texture map.

Embodiments of the invention described herein utilize UV and U maps in a new manner in that a pair of maps are utilized to define the horizontal offsets for two images (left and right) that each source pixel is translated to as opposed to a single map that is utilized to define a coordinate onto which a texture map is placed on a 3D model or wire frame. I.e., embodiments of the invention utilize UV and U maps (or any other horizontal translation file format) to allow for adjustments to the offset objects without re-rendering the entire scene. Again, as opposed to the known use of a UV map, for example that maps two orthogonal coordinates to a three-dimensional object, embodiments of the invention enabled herein utilize two maps, i.e., one for a left and one for a right eye, that map horizontal translations for the left and right viewpoints. In other words, since pixels translate only in the horizontal direction (for left and right eyes), embodiments of the invention map within one-dimension on a horizontal line-by-line basis. I.e., the known art maps 2 dimensions to 3 dimensions, while embodiments of the invention utilize 2 maps of translations within 1 dimension (hence visible embodiments of the translation map can utilize one color). For example, if one line of a translation file contains 0, 1, 2, 3 . . . 1918, 1919, and the 2nd and 3rd pixels are translated right by 4 pixels, then the line of the file would read 0, 4, 5, 3 . . . 1918, 1919. Other formats showing relative offsets are not viewable as ramped color areas, but may provide great compression levels, for example a line of the file using relative offsets may read, 0, 0, 0, 0 . . . 0, 0, while a right shift of 4 pixels in the 2nd and 3rd pixels would make the file read 0, 4, 4, 0, . . . 0, 0. This type of file can be compressed to a great extent if there are large portions of background that have zero horizontal offsets in both the right and left viewpoints. However, this file could be viewed as a standard U file is it was ramped, i.e., made absolute as opposed to relative to view as a color-coded translation file. Any other format capable of storing offsets for horizontal shifts for left and right viewpoints may be utilized in embodiments of the invention. UV files similarly have a ramp function in the Y or vertical axis as well, the values in such a file would be (0,0), (0,1), (0,2) . . . (0, 1918), (0,1919) corresponding to each pixel, for example for the bottom row of the image and (1,0), (1,1), etc., for the second horizontal line, or row for example. This type of offset file allows for movement of pixels in non-horizontal rows, however embodiments of the invention simply shift data horizontally for left and right viewpoints, and so do not need the to keep track of which vertical row a source pixel moves to since horizontal movement is by definition within the same row.

FIG. **88** shows a disparity map showing the areas where the difference between the left and right translation maps is the largest. This shows that objects closest to the viewer have pixels that are shifted the most between the two UV (or U) maps shown in FIG. **85**A-B (or **86**A-B).

FIG. **89** shows a left eye rendering of the source image of FIG. **82**. FIG. **90** shows a right eye rendering of the source

image of FIG. **82**. FIG. **91** shows an anaglyph of the images of FIG. **89** and FIG. **90** for use with Red/Blue glasses.

FIG. **92** shows an image that has been masked and is in the process of depth enhancement for the various layers, including the actress layer, door layer, background layer (showing missing background information that may be filled in through generation of missing information—see FIGS. **34**, **73** and **76** for example). I.e., the empty portion of the background behind the actress in FIG. **92** can be filled with generated image data, (see the outline of the actress's head on the background wall). Through utilization of generated image data for each layer, a compositing program for example may be utilized as opposed to re-rendering or ray-tracing all images in a scene for real-time editing. For example, if the hair mask of the actress in FIG. **92** is altered to more correctly cover the hair, then any pixels uncovered by the new mask that are obtained from the background and are nearly instantaneous available to view (as opposed to standard re-rendering or ray-tracing that can take hours of processing power to re-render all of the images in a scene when anything in a scene is edited). This may include obtaining generated data for any layer including the background for use in artifact free 3D image generation.

FIG. **93** shows a UV map overlaid onto an alpha mask associated with the actress shown in FIG. **92** which sets the translation offsets in the resulting left and right UV maps based on the depth settings of the various pixels in the alpha mask. This UV layer may be utilized with other UV layers to provide a quality assurance workgroup (or other workgroup) with the ability to real-time edit the 3D images, for example to correct artifacts, or correct masking errors without re-rendering an entire image. Iterative workflows however may require sending the frame back to a third-world country for rework of the masks, which are then sent back to a different workgroup for example in the United States to re-render the image, which then viewed again by the quality assurance workgroup. This type of iterative workflow is eliminated or minor artifacts altogether since the quality assurance workgroup can simply reshape an alpha mask and regenerate the pixel offsets from the original source image to edit the 3D images in real-time and avoid involving other workgroups for example. Setting the depth of the actress as per FIGS. **42-70** for example or in any other method determines the amount of shift that the unaltered UV map undergoes to generate to UV maps, one for left-eye and one for right-eye image manipulation as per FIGS. **85**A-D, (or U maps in FIGS. **86**A-D). The maps may be supplied for each layer along with an alpha mask for example to any compositing program, wherein changes to a mask for example allows the compositing program to simply obtain pixels from other layers to "add up" an image in real-time. This may include using generated image data for any layer (or gap fill data if no generated data exists for a deeper layer). One skilled in the art will appreciate that a set of layers with masks are combined in a compositing program to form an output image by arbitrating or otherwise determining which layers and corresponding images to lay on top of one another to form an output image. Any method of combining a source image pixel to form an output pixel using a pair of horizontal translation maps without re-rendering or ray-tracing again after adding depth is in keeping with the spirit of the invention.

FIG. **94** shows a workspace generated for a second depth enhancement program, based on the various layers shown in FIG. **92**, i.e., left and right UV translation maps for each of the alphas wherein the workspace allows for quality assurance personnel (or other workgroups) to adjust masks and hence alter the 3D image pair (or anaglyph) in real-time without re-rendering or ray-tracking and/or without iteratively sending fixes to any other workgroup. One or more embodiments of the invention may loop through a source file for the number of layers and create script that generates the workspace as shown in FIG. **94**. For example, once the mask workgroup has created the masks for the various layers and generated mask files, the rendering group may read in the mask files programmatically and generate script code that includes generation of a source icon, alpha copy icons for each layer, left and right UV maps for each layer based on the rendering groups rendered output, and other icons to combine the various layers into left and right viewpoint images. This allows the quality assurance workgroup to utilize tools that they are familiar with and which may be faster and less complex than the rendering tools utilized by the rendering workgroup. Any method for generation of a graphical user interface for a worker to enable real-time editing of 3D images including a method to create a source icon for each frame, that connects to an alpha mask icon for each layer and generates translation maps for left and right viewpoints that connect to one another and loops for each layer until combining with an output viewpoint for 3D viewing is in keeping with the spirit of the invention. Alternatively, any other method that enables real-time editing of images without re-rendering through use of a pair of translation maps is in keeping with the spirit of the invention even if the translation maps are not viewable or not shown to the user.

FIG. **95** shows a workflow for iterative corrective workflow. A mask workgroup generates masks for objects, such as for example, human recognizable objects or any other shapes in an image sequence at **9501**. This may include generation of groups of sub-masks and the generation of layers that define different depth regions. This step is generally performed by unskilled and/or low wage labor, generally in a country with very low labor costs. The masked objects are viewed by higher skilled employees, generally artists, who apply depth and/or color to the masked regions in a scene at **9502**. The artists are generally located in an industrialized country with higher labor costs. Another workgroup, generally a quality assurance group then views the resulting images at **9503** and determines if there are any artifacts or errors that need fixing based on the requirements of the particular project. If so, the masks with errors or locations in the image where errors are found are sent back to the masking workgroup for rework, i.e., from **9504** to **9501**. Once there are no more errors, the process completes at **9505**. Even in smaller workgroups, errors may be corrected by re-reworking masks and re-rendering or otherwise ray-tracing all of the images in a scene which can take hours of processing time to make a simple change for example. Errors in depth judgment generally occur less often as the higher skilled laborers apply depths based on a higher skill level, and hence kickbacks to the rendering group occur less often in general, hence this loop is not shown in the figure for brevity although this iterative path may occur. Masking "kickback" may take a great amount of time to work back through the system since the work product must be re-masked and then re-rendered by other workgroups.

FIG. **96** shows an embodiment of the workflow enabled by one or more embodiments of the system in that each workgroup can perform real-time editing of 3D images without re-rendering for example to alter layers/colors/masks and/or remove artifacts and otherwise correct work product from another workgroup without iterative delays associated with re-rendering/ray-tracing or sending work product back through the workflow for corrections. The generation of masks occurs as in FIG. **95** at **9501**, depth is applied as occurs in FIG. **95** at **9502**. In addition, the rendering group generates

translation maps that accompany the rendered images to the quality assurance group at **9601**. The quality assurance group views the work product at **9503** as in FIG. **95** and also checks for artifacts as in FIG. **95** at **9504**. However, since the quality assurance workgroup (or other workgroup) has translation maps, and the accompanying layers and alpha masks, they can edit 3D images in real-time or otherwise locally correct images without re-rendering at **9602**, for example using commercially available compositing programs such as NUKE® as one skilled in the art will appreciate. For instance as is shown in FIG. **94**, the quality assurance workgroup can open a graphics program that they are familiar with (as opposed to a complex rendering program used by the artists), and adjust an alpha mask for example wherein the offsets in each left right translation map are reshaped as desired by the quality assurance workgroup and the output images are formed layer by layer (using any generated missing background information as per FIGS. **34**, **73** and **76** and any computer generated element layers as per FIG. **79**). As one skilled in the art will recognize, generating two output images from furthest back layer to foreground layer can be done without ray-tracing, by only overlaying pixels from each layer onto the final output images nearly instantaneously. This effectively allows for local pixel-by-pixel image manipulation by the quality assurance workgroup instead of 3D modeling and ray-tracing, etc., as utilized by the rendering workgroup. This can save multiple hours of processing time and/or delays associated with waiting for other workers to re-render a sequence of images that make up a scene.

FIGS. **118-126** illustrate exemplary embodiments of an embodiment of the invention that includes the grouping tool interface **11801**. In one or more embodiments of the invention, the multi-stage production pipeline system for the motion picture projects includes the motion picture project management system described above. According to at least one embodiment, the multi-stage production pipeline system includes the computer **9702** and the database **9701** coupled with the computer **9702** as previously described in FIG. **97**. In at least one embodiment, the plurality of images are associated with a motion picture and wherein the database further includes metadata associated with at least one shot or associated with regions within the plurality of images in the at least one shot, or both. In at least one embodiment of the invention, the multi-stage production pipeline system includes a project table, such that the project table includes a project identifier and description of a project related to the motion picture as previously described.

FIG. **118** shows the grouping tool interface **11801** that may include metadata categories **11802** associated with shot characteristics or elements in the images or scenes, text field **11803** to add additional metadata to the one or more metadata categories **11802**, sort options **11804** for one or more search results and a refresh search tabs **11805**. According to one or more embodiments of the invention, the metadata categories **11802** for shot characteristics may include a range of elements such as locale, subject, framing, depth complexity and clean plate complexity. In one or more embodiments, the metadata associated with the at least one shot is associated with the metadata category **11802**, wherein the metadata category **11802** includes one or more of the depth complexity associated with the shot and the clean plate complexity associated with the shot. The grouping tool interface **11801** may present user interface elements, accept input of metadata and accept selected shots associated with the metadata via the user interface elements. By way of one or more embodiments, the metadata associated with the at least one shot is associated with at least one metadata category **11802**. In at least one embodiment of the invention, the grouping tool interface may accept at least one additional metadata category via text field **11803** and additional metadata values associated the metadata category **11802**.

For example, if a certain character is to be tagged, then a metadata value is added to the "subject" metadata category wherein the subject's value may be the actor's name. Alternatively, if a chair exists in multiple scenes, then the artist or manager may enter a value of "chair" and tag particular masks associated with the chair and associate the metadata value with the scenes where the chair appears. This enables an artist to work on the chair in non-linear time sequence, which produces extremely consistent results, shortens the time for applying color and/or depth to the object and lowers cost for the project.

In one or more embodiments, as explained above, the metadata category **11802** comprises one or more of a locale or location at which the shot was obtained, a subject that appears in the shot wherein the subject is a person, place or thing, a shot framing associated with the shot wherein the shot framing includes one or more of a close up or CU, a mid shot or MS, wide shot or WS, and an extreme wide shot or XWS. In at least one embodiment, the grouping tool interface and the metadata categories may transition shot framing including any combination of CU, MS, SW and XWS, such that a CU may transition to a MS, WS or XWS, and/or a MS may transition to a CU, WS or XWS, and/or a WS may transition to a CU, MS or XWS, and/or a XWS may transition to a CU, MS or WS.

FIG. **119** illustrates a view of the grouping tool interface with metadata tags and shot tables according to one or more embodiments of the invention. According to FIG. **119**, the grouping tool interface **11801** presents a shot table image **11901**, at least one tab **11902** that adds metadata to preselected scenes or images within the scenes, at least one tab **11903** that replaces pre-existing metadata associated with a scene or images within the scene, metadata tags **11904** and at least one shot list **11905**, for example as stored in the shot table in database **9701**. In at least one embodiment of the invention, the computer **9702** may one or more of store the metadata **11904** associated with the selected shots in the shot list **11905**, accept selected metadata to search the at least one shot, query the shot list **11905** with the selected metadata associated with the at least one shot or regions within a plurality of images in the at least one shot or both, and display a list of shots having the selected metadata. By way of one or more embodiments, the database **9701** includes the shot table **11905** with a shot identifier associated with a plurality of images that are ordered in time and that make up a shot, such that the shot table **11905** has a starting frame value and an ending frame value associated with each shot.

Once a user has searched for the shots to add/change metadata to, or to find scenes to work on with a particular subject for example, the user may select one or more of the metadata tags **11904** from the metadata category columns **11802**. In one or more embodiments, the list of shots may include at least one shot that is non-sequential in time in the motion picture with respect to another shot in the list of shots, which enables non-linear workflow to occur on the motion picture project. The computer **9702** may assign work tasks based on the list of shots wherein the list of shots includes the at least one shot that is non-sequential in time in the motion picture with respect to another shot in the list of shots. Once the selected shots have the appropriate metadata tagged, the user may search for shots that share the same metadata tags in order to group or "bucket" the respective shots into groups.

FIG. **120** illustrates a view of the grouping tool interface **11801** with shots that share metadata characteristics according to one or more embodiments of the invention. For example, as shown in FIG. **120**, a user may search for all the sequential shots that contain the subject: "MAX", locale: "SHIP ERIC", and framing: "CU", as presented in the respective metadata category columns **11802**. FIG. **120** illustrates the grouping tool interface with group codes **12001**, shots that share characteristics **12002**, for example shown as highlighted or otherwise selected, and a tab **12003** that adds selected shots to a group to be assigned together.

FIG. **121** illustrates a view of the grouping tool interface with a key select shot and associated color code, the depth complexity and associated color code and the clean plate complexity and associated color code according to one or more embodiments. FIG. **121** presents key select element **12101**, depth master or complexity element **12102**, and clean plate complexity element **12103**, all using color codes. Element **12104** represents a key select and depth complexity master element with a respective color code. In at least one embodiment of the invention, the grouping tool interface **11801** includes a reference mask library tool with a plurality of reference masks, wherein each of the list of shots share selected metadata characteristics as described with reference to FIG. **120** above.

Once the selected shots have been grouped together, the reference masks may be assigned and checked out to all shots in a group that were assigned together as discussed with reference to FIG. **120** above. For example, selecting "Add Mask Lib Ref" opens an additional panel in the grouping tool interface **11802** to allow a user to search for masks that fit the assigned group a user is working on. For example, a user may select a reference that shares similar metadata to the selected shots, and right clicking on the metadata characteristics needed to use in the selected shots to add metadata to. In one or more embodiments, the reference mask library tool may present an interface to accept a selection of one or more of the plurality of reference masks to be utilize in shots in the list of shots that do not already utilize the reference mask associated with the subject, for example. Integration of the Mark Library Reference Tool within the grouping tool interface **11802** allows a user to search for masks across multiple projects. This may enable a worker to locate similar subjects in one or more additional shots and use reference masks associated with the subject located to add metadata to the one or more additional shots, without the need to reinvent metadata for each of those additional shots with the same subject such as a person, place or thing. In one or more embodiments, each one of the plurality of reference masks may be a dedicated template of the subject.

FIG. **121** also presents a drop down menu **12105** with different search options to enable a worker to build, search and sort timelines, for example to show the non-linear scenes in a movie viewing window or otherwise display images to work on, and a tab that refines search criteria **12106** to allow a user to search across multiple projects of a single motion picture, or several motion pictures. For example, the plurality of reference masks may be obtained from a second project associated with a second motion picture that differs from the first motion picture. This allows the system to create different sequels and/or motions pictures with similar subjects, such as people, places or things, from the first motion picture, with similar metadata to enable time efficiency, consistency and accuracy.

By way of one or more embodiments, the grouping tool interface may display a plurality of search results including the lists of shots associated with a plurality of respective selected metadata. In addition, in at least one embodiment of the invention, the grouping tool interface may present a timeline of the plurality of images associated with the list of shots. For example, two different search results may be presented to enable a user to see differences in shots or lists of shots associated with different sets of metadata, and add any necessary metadata if missing metadata is found.

In one or more embodiments, the may computer may one or more of present a first display to be viewed by a production worker that includes a search display with one or more of a context, a project, a shot, the list of shots, a status and an artist, present a second display to be viewed by an artist that includes at least one daily assignment having a context, project and shot or the list of shots or both, and present a third display to be viewed by an editorial worker that includes an annotation frame to accept commentary or drawing or both commentary and drawing on at least one of the plurality of images associated with the at least one shot or the list of shots or both. In one or more embodiments of the invention, the at least one shot or the list of shots, or both, include status related to progress of work performed.

FIG. **122** illustrates another view of the grouping tool interface according to one or more embodiments on the invention. As shown in FIG. **122**, by way one or more embodiments, the grouping tool interface **11802** may accept an input to designate the at least one shot as a master shot associated with depth, key selects or clean plate or any combination thereof. This enables the at least one shot to be utilized as a benchmark for quality or volume or to improve efficiency or both. For example, right clicking on shots in the search results window allows a user to set shots to be Key Selects, Depth Masters, Key Selects/Depth Masters and Clean Plate. For example, Key Selects, Depth Masters and Key Selects/Depth Masters may be benchmarks for quality and volume for a shot, and Clean Plate may set one shot as the reference, improving efficiency when other shots are clean plated within the assigned group.

FIG. **123** illustrates a close-up view of the grouping tool interface according to one or more embodiments, with metadata category columns **11802**, text field **11803** to add additional metadata take to the one or more metadata categories **11802** and metadata tags **11904**, such as "NORA" under the subject category, and "WS" under the framing category.

FIG. **124** illustrates a close-up view of a shot of the grouping tool interface according to one or more embodiments, showing a search results columns. Selected items in the search column may also be loaded into a Frame Cycler timeline by clicking a "Build Timeline" drop down menu (not shown) also part of the grouping tool interface **11801**. Built timelines may be referenced as base timelines within a Review Tool of the grouping tool interface **11801**, and may be populated with the latest metadata to maintain quality, style, technique and efficiency when working within a sequential list of shots, non-sequential list of shots, over an entire motion picture, or several motion pictures if similar characteristics are shared.

As also discussed with reference to FIG. **99**, a user may select an Open MSI (Master Shot Index) Option that will open an MSI tool for any shot or list of shots the user has selected from the search results column. In doing so, a user is able to enter notes, look at metadata tagged shots, look at original shots and check data integrity associated with each shot within the grouping tool interface **11801**. This allows for an integrated view between data describing one or more shots and the entire shot metadata tagging process taking place.

FIG. **125** illustrates a close-view of a plurality of shot to add metadata to according to one or more embodiments and in

addition shows a popup of other metadata associated with a scene, including for example the number of frames in the shot, the reel on which the shot occurs, the locale of the shot, here "LIVINGROOM", the subject(s) that appear in the scene, here "FREDERICK" and the type of framing, here "MS" or mid-shot.

FIG. **126** illustrates an overall view of the grouping interface with a shot from a plurality of images with metadata.

Embodiments enable a large studio workforce to work non-linearly on a film while maintaining a unified vision driven by key creative figures. Thus, work product is more consistent, higher quality, faster, less expensive and enables reuse of project files, masks and other production element across projects since work is no longer constrained by shot order when using embodiments of the invention.

While the invention herein disclosed has been described by means of specific embodiments and applications thereof, numerous modifications and variations could be made thereto by those skilled in the art without departing from the scope of the invention set forth in the claims.

What is claimed is:

1. A multi-stage production pipeline system for motion picture projects comprising:

a computer;

a database coupled with said computer, wherein said database comprises

a shot table that comprises a shot identifier associated with a plurality of images that are ordered in time and that each make up at least one shot wherein said shot table comprises a starting frame value and an ending frame value associated with each of said shots;

an asset table that comprises information on one or more assets used in production of said motion picture;

wherein

said plurality of images are associated with a motion picture, and

said database further includes metadata associated with said at least one shot or associated with regions within said plurality of images in said at least one shot or both;

wherein said computer is configured to

calculate an amount of disk space that may be utilized to archive said one or more assets and

signify at least one asset of said one or more assets that may be rebuilt from other assets to avoid archival of said at least one asset.

2. The system of claim **1** wherein said computer is further configured to calculate an amount of disk space that may be saved by said avoid archival of said at least one asset.

3. The system of claim **1** wherein said information on one or more assets comprises an indicator of whether an asset can be rebuilt from said other assets.

4. The system of claim **1** wherein said information on one or more assets comprises a dependency graph that indicates which assets depend on which other assets.

5. The system of claim **1** wherein said information on one or more assets comprises a compression value that indicates the extent to which an asset can be compressed.

6. The system of claim **1** wherein said computer is further configured to calculate or estimate an amount of disk space that may be saved by compression of one or more of said one or more assets.

7. The system of claim **1** wherein

said computer is further configured to present a grouping tool interface coupled with said computer and said database;

said grouping tool interface is configured to

present user interface elements, accept input of said metadata and accept selected shots associated with said metadata via said user interface elements;

store said metadata associated with said selected shots in said shot table;

accept selected metadata to search said at least one shot;

query said shot table with said selected metadata associated with said at least one shot or said regions within said plurality of images in said at least one shot or said both; and,

display a list of shots having said selected metadata, wherein said list of shots includes at least one shot that is non-sequential in time in said motion picture with respect to another shot in said list of shots.

8. The system of claim **7**, wherein said metadata associated with said at least one shot is associated with a metadata category comprising a locale or location at which said shot was obtained.

9. The system of claim **7**, wherein said metadata associated with said at least one shot is associated with a metadata category comprising a subject that appears in said shot wherein said subject is a person, place or thing.

10. The system of claim **7**, wherein said metadata associated with said at least one shot is associated with a metadata category comprising a shot framing associated with said at least one shot.

11. The system of claim **7**, wherein said metadata associated with said at least one shot is associated with a metadata category comprising a depth complexity or a clean plate complexity associated with said shot.

12. The system of claim **7**, wherein said grouping tool interface is configured to accept at least one additional metadata category and additional metadata values associated with said metadata category.

13. The system of claim **7**, wherein said grouping tool interface is configured to accept an input to designate said at least one shot as a master shot associated with depth, key selects or clean plate or any combination thereof that enables said at least one shot to be utilized as a benchmark for quality or volume, or to improve efficiency, or both.

14. The system of claim **9**, wherein said grouping tool interface comprises a reference mask library tool with a plurality of reference masks, wherein each of said list of shots share said selected metadata, and wherein said reference mask library tool is configured to present an interface to accept a selection of one or more of said plurality of reference masks to be utilized in shots in said list of shots that do not already utilize said reference mask associated with said subject.

15. The system of claim **14**, wherein each one of said plurality of reference masks is configured as a dedicated template of said subject.

16. The system of claim **14**, wherein at least one of said plurality of reference masks is configured to be obtained from a second motion picture that differs from said motion picture.

17. The system of claim **7**, wherein said grouping tool interface is further configured to present a timeline of said plurality of images associated with said list of shots.

18. The system of claim **7**, wherein said computer is further configured to assign work tasks based on said list of shots.

19. The system of claim **1**, wherein said shot table in said database further comprises status related to progress of work performed on said at least one shot.

20. The system of claim **1**, wherein said database further comprises

a task table which includes at least one task which comprises a task identifier and an assigned worker and which

further comprises a context setting associated with a type of task related to motion picture work wherein said task includes at least definition of a region within said plurality of images, work on said region and composite work on said region and wherein said at least one task comprises a time allocated to complete said at least one task.

21. The system of claim **20**, wherein said database further comprises

a project table, wherein said project table comprises a project identifier and description of a project related to said motion picture.

22. The system of claim **21**, wherein said database further comprises

a timesheet item table which references said project identifier in said project table and said task identifier in said task table and which includes at least one timesheet item which comprises a start time and an end time.

23. The system of claim **1**, wherein said database further comprises an asset request table which comprises an asset request identifier and shot identifier.

24. The system of claim **1**, wherein said database further comprises a mask request table which comprises a mask request identifier and shot identifier.

25. The system of claim **21**, wherein said database further comprises

a note table which comprises a note identifier and which references said project identifier and which comprises at least one note related to at least one of said plurality of images from said motion picture.

26. The system of claim **1**, wherein said database further comprises

a snapshot table which comprises a snapshot identifier and search type and which includes a snapshot of said at least one shot that includes at least one location of a resource associated with said at least one shot.

27. The system of claim **1**, wherein said computer is further configured to

present a first display configured to be viewed by a production worker that includes a search display comprising a context, project, shot, status and artist;

present a second display configured to be viewed by an artist that includes at least one daily assignment having a context, project and shot; and,

present a third display configured to be viewed by an editorial worker that includes an annotation frame configured to accept commentary or drawing or both commentary and drawing on at least one of said plurality of images associated with said at least one shot.

28. The system of claim **27**, wherein said computer is further configured to

present an annotation overlaid on at least one of said plurality of images on said third display configured to be viewed by said editorial worker.

29. The system of claim **22**, wherein said computer is further configured to

calculate actuals based on total time spent for all of said at least one tasks associated with all of said at least one shot in said project.

30. The system of claim **29**, wherein said computer is further configured to

compare said actuals to time allocated for all of said at least one tasks associated with all of said at least one shot in said project;

based on said compare said actuals to time allocated for all of said at least one tasks, estimate one or more of

remaining cost for said project;

time of completion of said project.

* * * * *