US 20150248917A1

(19) **United States**

(12) **Patent Application Publication**

Chang et al.

(10) **Pub. No.: US 2015/0248917 A1**

(43) **Pub. Date: Sep. 3, 2015**

(54) **SYSTEM AND METHOD FOR PERFORMING SPATIO-TEMPORAL ANALYSIS OF SPORTING EVENTS**

(71) Applicant: **Second Spectrum, Inc.**, Pasadena, CA (US)

(72) Inventors: **Yu-Han Chang**, Pasadena, CA (US); **Rajiv Maheswaran**, Marina Del Rey, CA (US); **Jeff Su**, Los Angeles, CA (US)

(21) Appl. No.: **14/634,070**

(22) Filed: **Feb. 27, 2015**

Related U.S. Application Data

(60) Provisional application No. 61/945,899, filed on Feb. 28, 2014, provisional application No. 62/072,308, filed on Oct. 29, 2014.

Publication Classification

(51) **Int. Cl.**

G11B 27/034	(2006.01)
H04N 13/02	(2006.01)
H04N 21/2187	(2006.01)
H04N 13/00	(2006.01)
H04N 21/8549	(2006.01)

(52) **U.S. Cl.**

CPC ***G11B 27/034*** (2013.01); ***H04N 13/0011*** (2013.01); ***H04N 13/0055*** (2013.01); ***H04N 21/8549*** (2013.01); ***H04N 21/2187*** (2013.01); ***H04N 13/0203*** (2013.01); *G06T 2207/20081* (2013.01); *G06T 2207/30221* (2013.01)

(57) **ABSTRACT**

Methods and systems are provided to enable the exploration of event data captured from video feeds, such as from sporting event venues, the discovery of relevant events (such as within a video feed of a sporting event), and the presentation of novel insights, analytic results, and visual displays that enhance decision-making, provide improved entertainment and provide other benefits.

100 EAGLE STACK	CUSTOMIZATION	102	TEAM-SPECIFIC ANALYTICS, VISUALIZATIONS AND TOOLS ANALYTICS EXPORTS, CROWD-SOURCED FEEDBACK
	INTERACTION	104	REAL-TIME INTERACTIVE TOOLS TO SLICE AND DICE DATA SHOT MATRIX, SCREEN BREAKDOWN, POSSESSION DETECTIVE
	VISUALIZATIONS	108	DYNAMIC VISUALIZATIONS OF PATTERNS AND ANALYTICS SCATTER RANK, SHOT COMPARISON, CLIP VIEW
	ANALYTICS	110	NEW METRICS BASED ON AI, MACHINE LEARNING SITUATIONAL EFG, VORONOI-BASED RB%
	PATTERNS	112	INFRASTRUCTURE FOR RAPID DISCOVERY OF NEW PATTERNS SCREENS, DEFENSE, FACE-UP, POST-UP, TRANSITION,.....
	EVENTS	114	CREATE NEW EVENTS AND CORRECT CURRENT ERRORS ANALYZE ACCURACY OF MARKINGS [FGA,DRB,...]
	DATA	118	CATALOGUE, CORRECT AND ADAPT RAW DATA [X,Y,Z,T] MANAGE ONLINE WAREHOUSING IN THE CLOUD

100 EAGLE STACK	CUSTOMIZATION	102	TEAM-SPECIFIC ANALYTICS, VISUALIZATIONS AND TOOLS ANALYTICS EXPORTS, CROWD-SOURCED FEEDBACK
	INTERACTION	104	REAL-TIME INTERACTIVE TOOLS TO SLICE AND DICE DATA SHOT MATRIX, SCREEN BREAKDOWN, POSSESSION DETECTIVE
	VISUALIZATIONS	108	DYNAMIC VISUALIZATIONS OF PATTERNS AND ANALYTICS SCATTER RANK, SHOT COMPARISON, CLIP VIEW
	ANALYTICS	110	NEW METRICS BASED ON AI, MACHINE LEARNING SITUATIONAL EFG, VORONOI-BASED RB%
	PATTERNS	112	INFRASTRUCTURE FOR RAPID DISCOVERY OF NEW PATTERNS SCREENS, DEFENSE, FACE-UP, POST-UP, TRANSITION,.....
	EVENTS	114	CREATE NEW EVENTS AND CORRECT CURRENT ERRORS ANALYZE ACCURACY OF MARKINGS [FGA,DRB,...]
	DATA	118	CATALOGUE, CORRECT AND ADAPT RAW DATA [X,Y,Z,T] MANAGE ONLINE WAREHOUSING IN THE CLOUD

FIG. 1

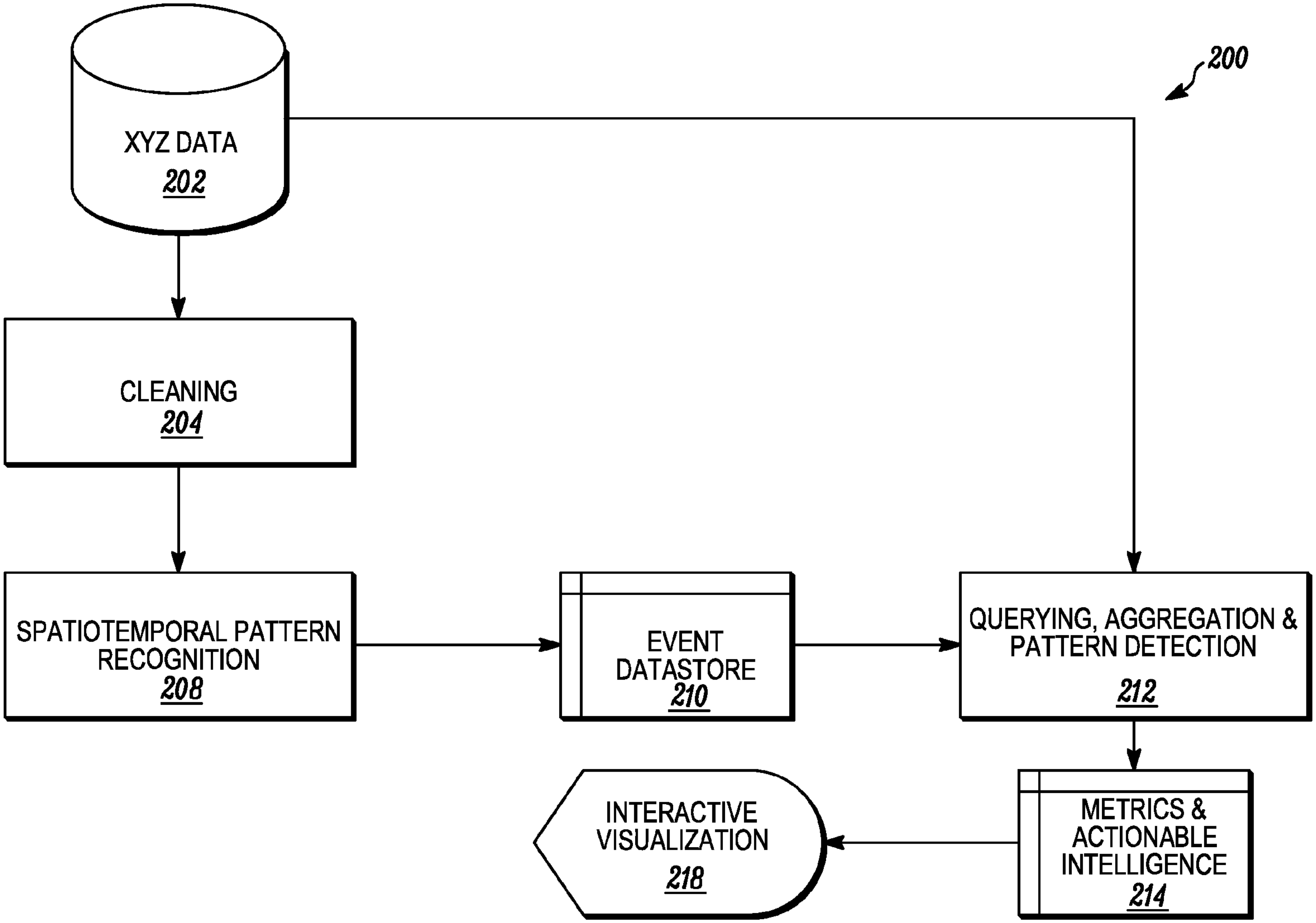

200
XYZ DATA
202
CLEANING
204
SPATIOTEMPORAL PATTERN RECOGNITION
208
EVENT DATASTORE
210
QUERYING, AGGREGATION & PATTERN DETECTION
212
METRICS & ACTIONABLE INTELLIGENCE
214
INTERACTIVE VISUALIZATION
218

FIG. 2

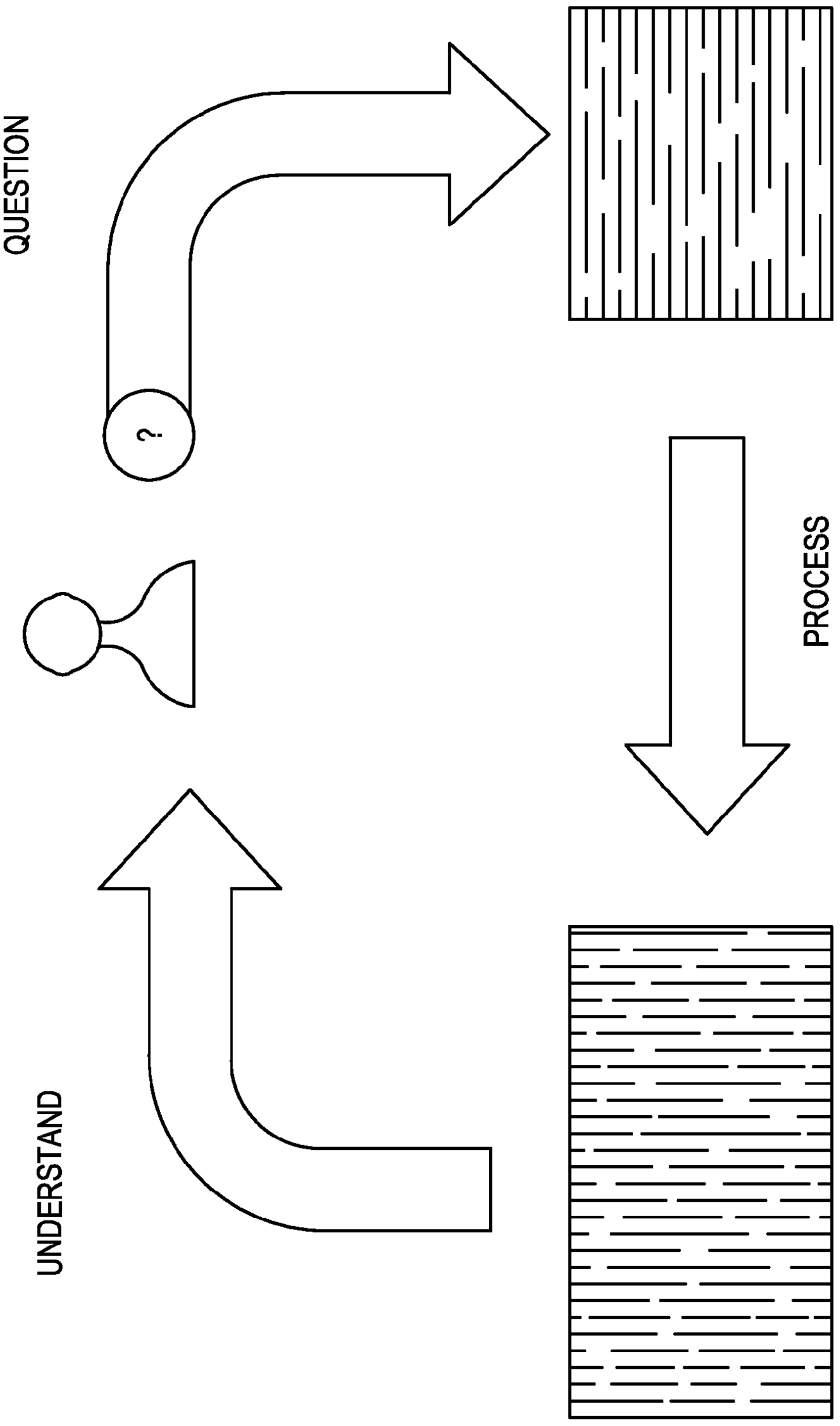
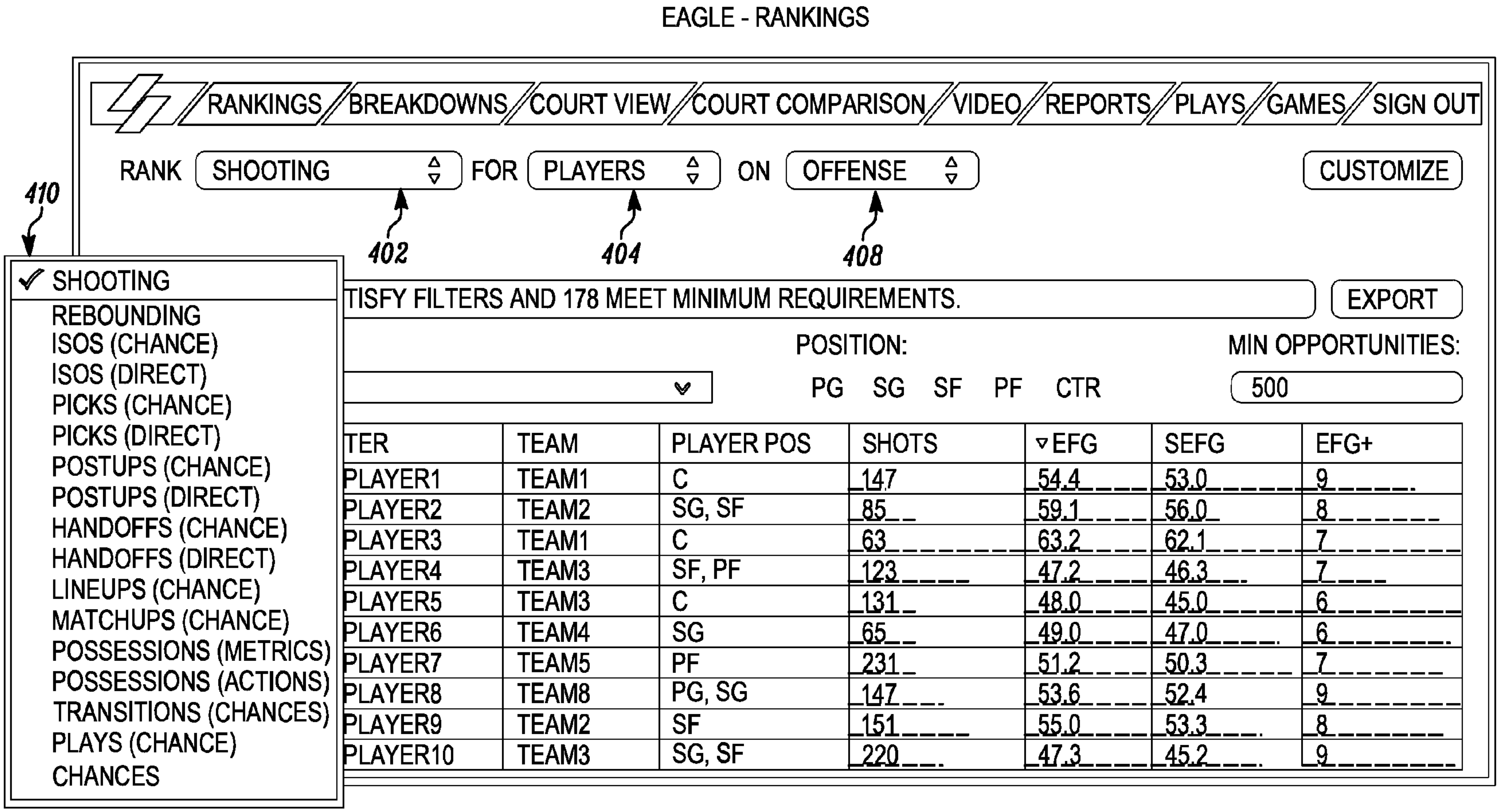


FIG. 3



RANK SHOOTING FOR PLAYERS ON OFFENSE

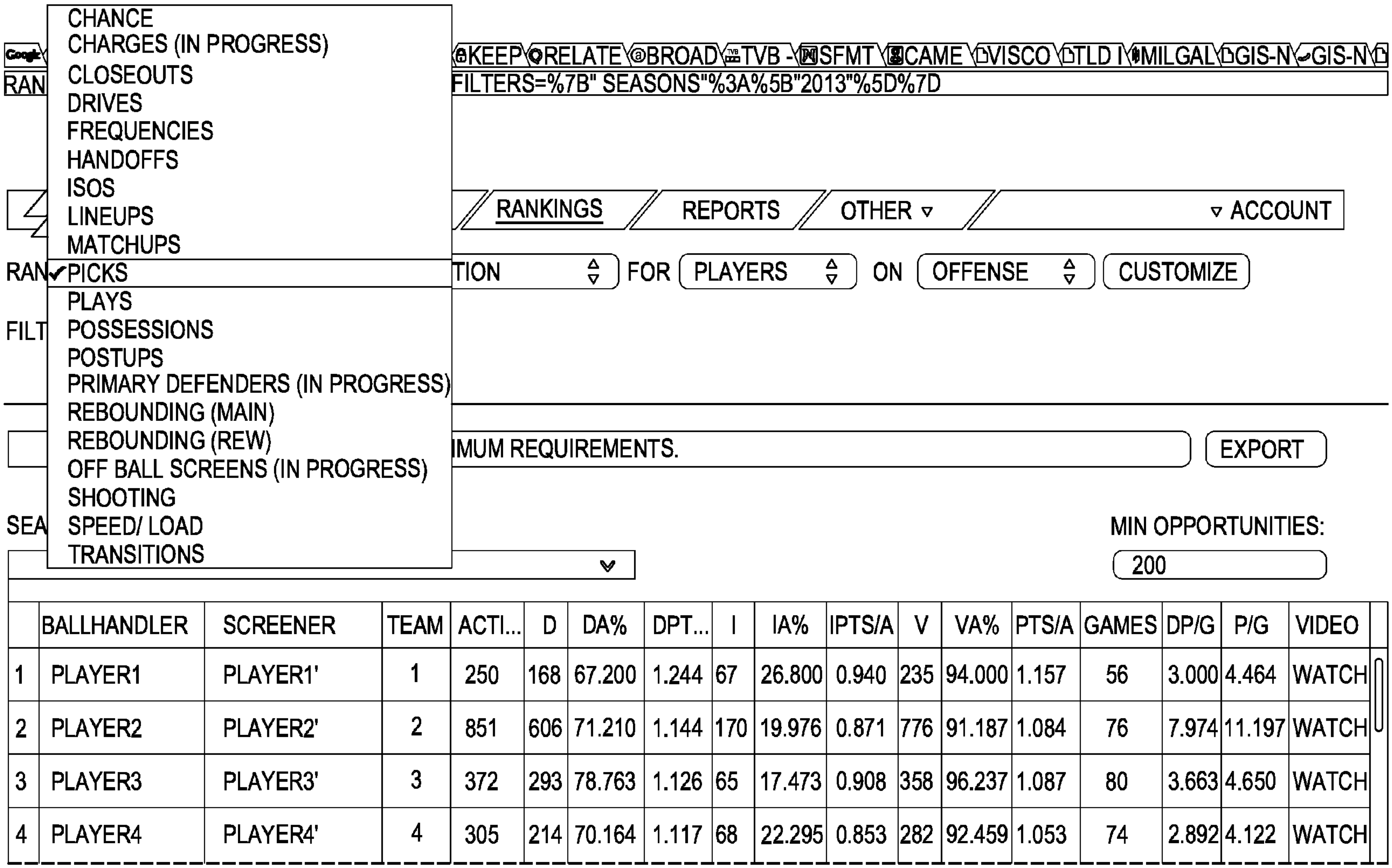
TISFY FILTERS AND 178 MEET MINIMUM REQUIREMENTS.

POSITION: PG SG SF PF CTR

MIN OPPORTUNITIES: 500

TER	TEAM	PLAYER POS	SHOTS	▿EFG	SEFG	EFG+
PLAYER1	TEAM1	C	147	54.4	53.0	9
PLAYER2	TEAM2	SG, SF	85	59.1	56.0	8
PLAYER3	TEAM1	C	63	63.2	62.1	7
PLAYER4	TEAM3	SF, PF	123	47.2	46.3	7
PLAYER5	TEAM3	C	131	48.0	45.0	6
PLAYER6	TEAM4	SG	65	49.0	47.0	6
PLAYER7	TEAM5	PF	231	51.2	50.3	7
PLAYER8	TEAM8	PG, SG	147	53.6	52.4	9
PLAYER9	TEAM2	SF	151	55.0	53.3	8
PLAYER10	TEAM3	SG, SF	220	47.3	45.2	9

FIG. 4



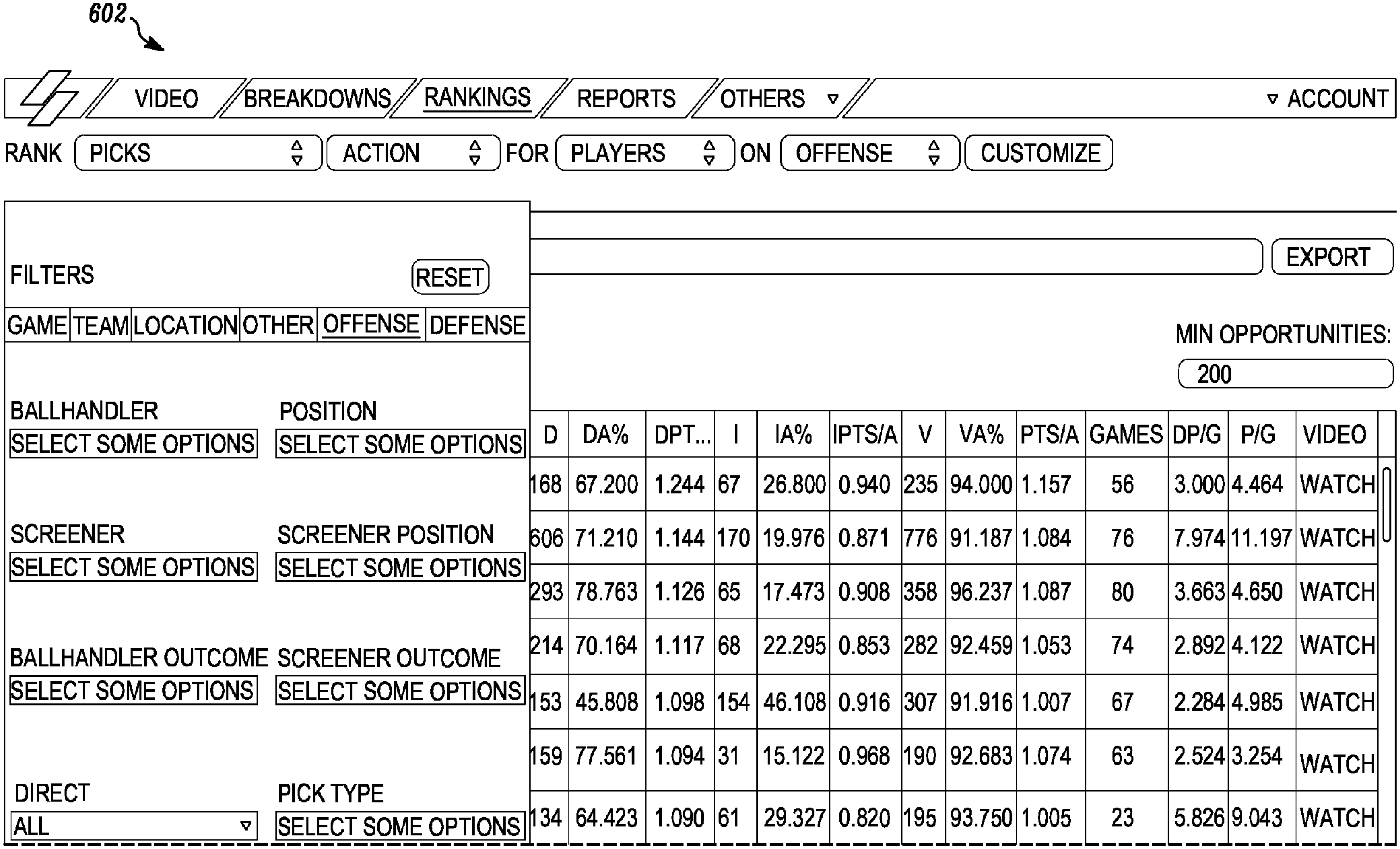
MIN OPPORTUNITIES: 200

	BALLHANDLER	SCREENER	TEAM	ACTI...	D	DA%	DPT...	I	IA%	IPTS/A	V	VA%	PTS/A	GAMES	DP/G	P/G	VIDEO
1	PLAYER1	PLAYER1'	1	250	168	67.200	1.244	67	26.800	0.940	235	94.000	1.157	56	3.000	4.464	WATCH
2	PLAYER2	PLAYER2'	2	851	606	71.210	1.144	170	19.976	0.871	776	91.187	1.084	76	7.974	11.197	WATCH
3	PLAYER3	PLAYER3'	3	372	293	78.763	1.126	65	17.473	0.908	358	96.237	1.087	80	3.663	4.650	WATCH
4	PLAYER4	PLAYER4'	4	305	214	70.164	1.117	68	22.295	0.853	282	92.459	1.053	74	2.892	4.122	WATCH

FIG. 5A

5	PLAYER5	PLAYER5'	5	334	153	45.808	1.098	154	46.108	0.916	307	91.916	1.007	67	2.284	4.985	WATCH	
6	PLAYER6	PLAYER6'	3	205	159	77.561	1.094	31	15.122	0.968	190	92.683	1.074	63	2.524	3.254	WATCH	
7	PLAYER7	PLAYER7'	6	208	134	64.423	1.090	61	29.327	0.820	195	93.750	1.005	23	5.826	9.043	WATCH	
8	PLAYER8	PLAYER8'	7	367	229	62.398	1.087	114	31.063	0.930	343	93.460	1.035	52	4.404	7.058	WATCH	
9	PLAYER9	PLAYER9'	8	330	236	71.515	1.085	71	21.515	0.789	307	93.030	1.016	64	3.688	5.156	WATCH	
10	PLAYER10	PLAYER10'	9	294	189	64.286	1.085	86	29.252	0.930	275	93.537	1.036	46	4.109	6.391	WATCH	
11	PLAYER11	PLAYER11'	10	206	133	64.563	1.083	61	29.612	0.967	194	94.175	1.046	24	5.542	8.538	WATCH	
12	PLAYER12	PLAYER12'	5	253	160	63.241	1.063	78	30.830	1.141	238	94.071	1.088	52	3.077	4.865	WATCH	
13	PLAYER13	PLAYER13'	1	263	150	57.034	1.060	96	36.502	0.813	246	93.536	0.963	51	2.941	5.157	WATCH	
14	PLAYER14	PLAYER14'	11	216	151	69.907	1.060	55	25.463	1.109	206	95.370	1.073	49	3.082	4.408	WATCH	
15	PLAYER15	PLAYER15'	8	272	188	69.118	1.059	65	23.897	0.692	253	93.015	0.964	49	3.837	5.551	WATCH	

FIG. 5B



RANK PICKS ACTION FOR PLAYERS ON OFFENSE

BALLHANDLER: SELECT SOME OPTIONS

POSITION: SELECT SOME OPTIONS

SCREENER: SELECT SOME OPTIONS

SCREENER POSITION: SELECT SOME OPTIONS

BALLHANDLER OUTCOME: SELECT SOME OPTIONS

SCREENER OUTCOME: SELECT SOME OPTIONS

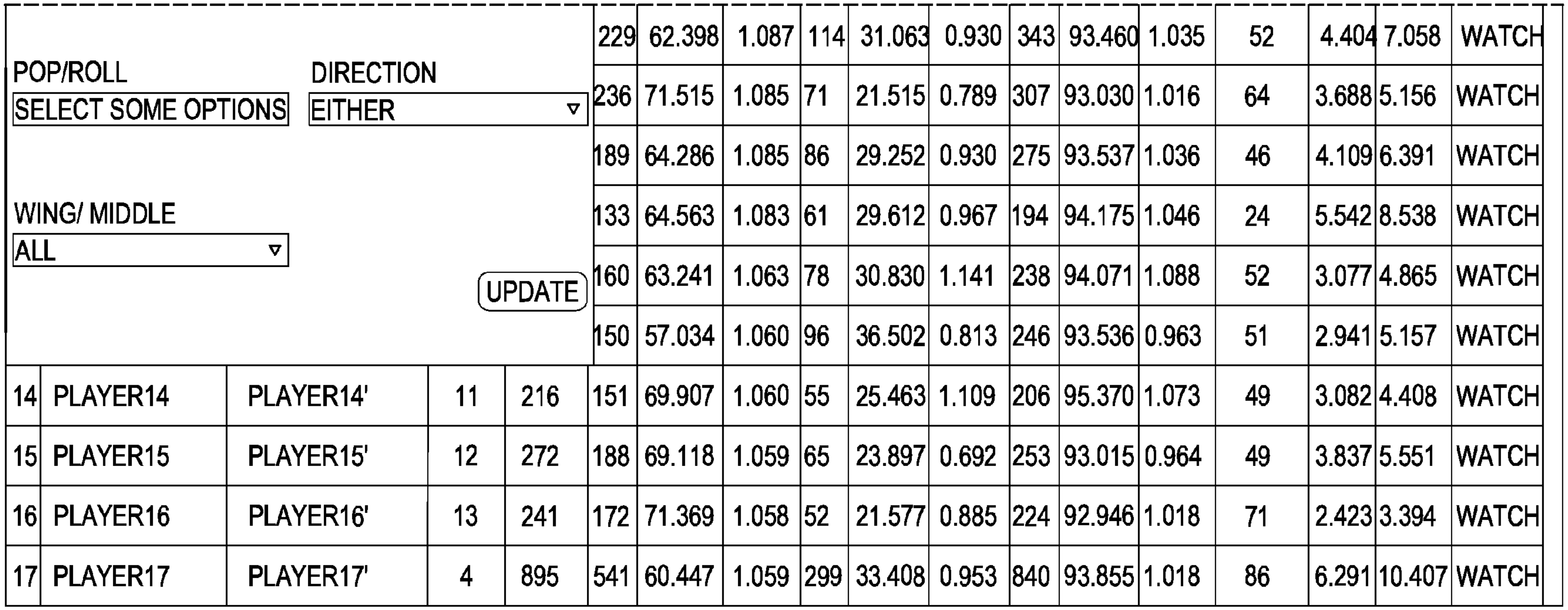
DIRECT: ALL ▿

PICK TYPE: SELECT SOME OPTIONS

MIN OPPORTUNITIES: 200

D	DA%	DPT...	I	IA%	IPTS/A	V	VA%	PTS/A	GAMES	DP/G	P/G	VIDEO
168	67.200	1.244	67	26.800	0.940	235	94.000	1.157	56	3.000	4.464	WATCH
606	71.210	1.144	170	19.976	0.871	776	91.187	1.084	76	7.974	11.197	WATCH
293	78.763	1.126	65	17.473	0.908	358	96.237	1.087	80	3.663	4.650	WATCH
214	70.164	1.117	68	22.295	0.853	282	92.459	1.053	74	2.892	4.122	WATCH
153	45.808	1.098	154	46.108	0.916	307	91.916	1.007	67	2.284	4.985	WATCH
159	77.561	1.094	31	15.122	0.968	190	92.683	1.074	63	2.524	3.254	WATCH
134	64.423	1.090	61	29.327	0.820	195	93.750	1.005	23	5.826	9.043	WATCH

FIG. 6A



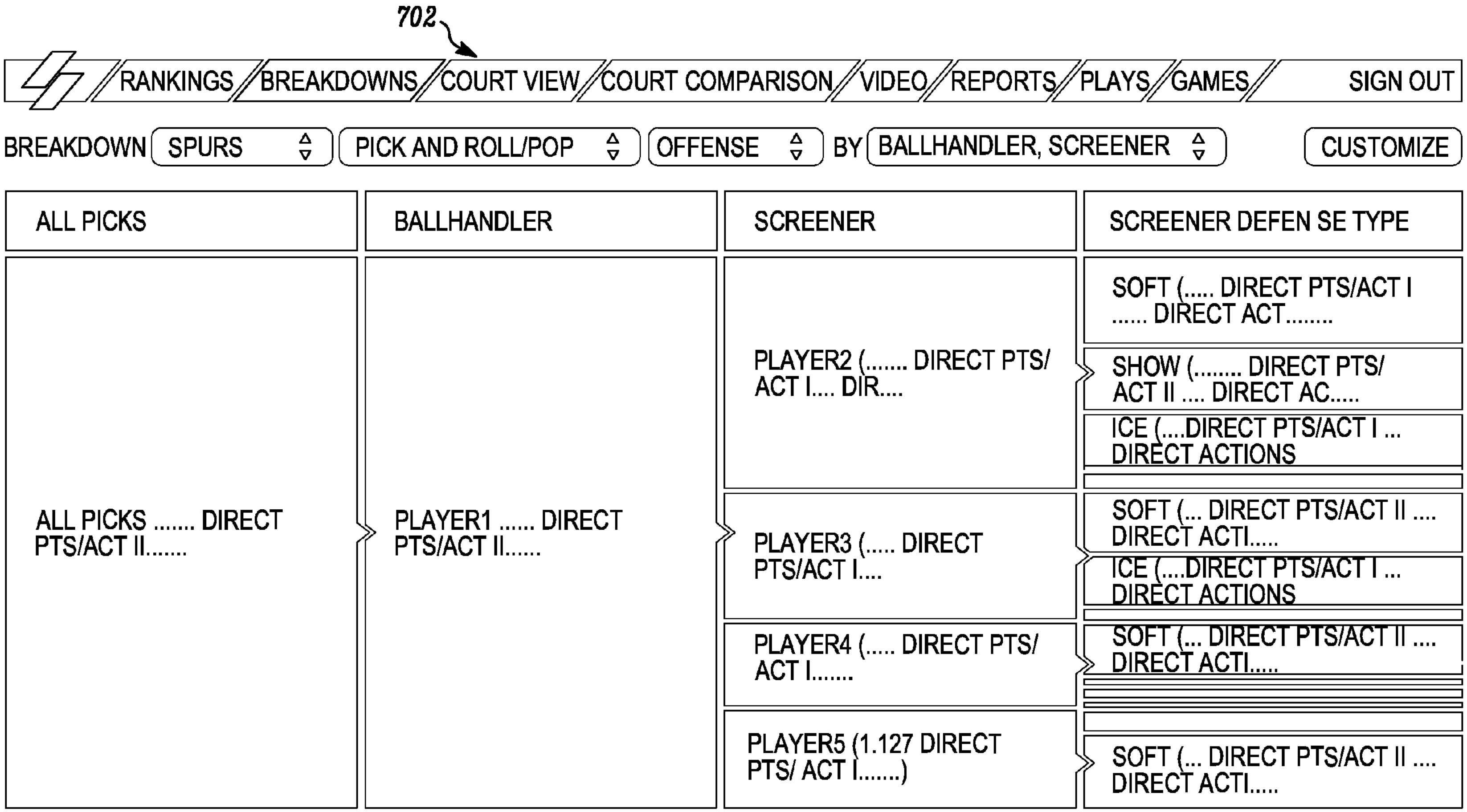
POP/ROLL
SELECT SOME OPTIONS

DIRECTION
EITHER ▽

WING/ MIDDLE
ALL ▽

					229	62.398	1.087	114	31.063	0.930	343	93.460	1.035	52	4.404	7.058	WATCH
					236	71.515	1.085	71	21.515	0.789	307	93.030	1.016	64	3.688	5.156	WATCH
					189	64.286	1.085	86	29.252	0.930	275	93.537	1.036	46	4.109	6.391	WATCH
					133	64.563	1.083	61	29.612	0.967	194	94.175	1.046	24	5.542	8.538	WATCH
					160	63.241	1.063	78	30.830	1.141	238	94.071	1.088	52	3.077	4.865	WATCH
					150	57.034	1.060	96	36.502	0.813	246	93.536	0.963	51	2.941	5.157	WATCH
14	PLAYER14	PLAYER14'	11	216	151	69.907	1.060	55	25.463	1.109	206	95.370	1.073	49	3.082	4.408	WATCH
15	PLAYER15	PLAYER15'	12	272	188	69.118	1.059	65	23.897	0.692	253	93.015	0.964	49	3.837	5.551	WATCH
16	PLAYER16	PLAYER16'	13	241	172	71.369	1.058	52	21.577	0.885	224	92.946	1.018	71	2.423	3.394	WATCH
17	PLAYER17	PLAYER17'	4	895	541	60.447	1.059	299	33.408	0.953	840	93.855	1.018	86	6.291	10.407	WATCH

FIG. 6B



BREAKDOWN SPURS PICK AND ROLL/POP OFFENSE BY BALLHANDLER, SCREENER

ALL PICKS

BALLHANDLER

SCREENER

SCREENER DEFEN SE TYPE

ALL PICKS .... DIRECT PTS/ACT II.......

PLAYER1 ...... DIRECT PTS/ACT II......

PLAYER2 (....... DIRECT PTS/ ACT I.... DIR....

SOFT (..... DIRECT PTS/ACT I ...... DIRECT ACT........

SHOW (........ DIRECT PTS/ ACT II .... DIRECT AC.....

ICE (....DIRECT PTS/ACT I ... DIRECT ACTIONS

PLAYER3 (..... DIRECT PTS/ACT I....

SOFT (... DIRECT PTS/ACT II .... DIRECT ACTI.....

ICE (....DIRECT PTS/ACT I ... DIRECT ACTIONS

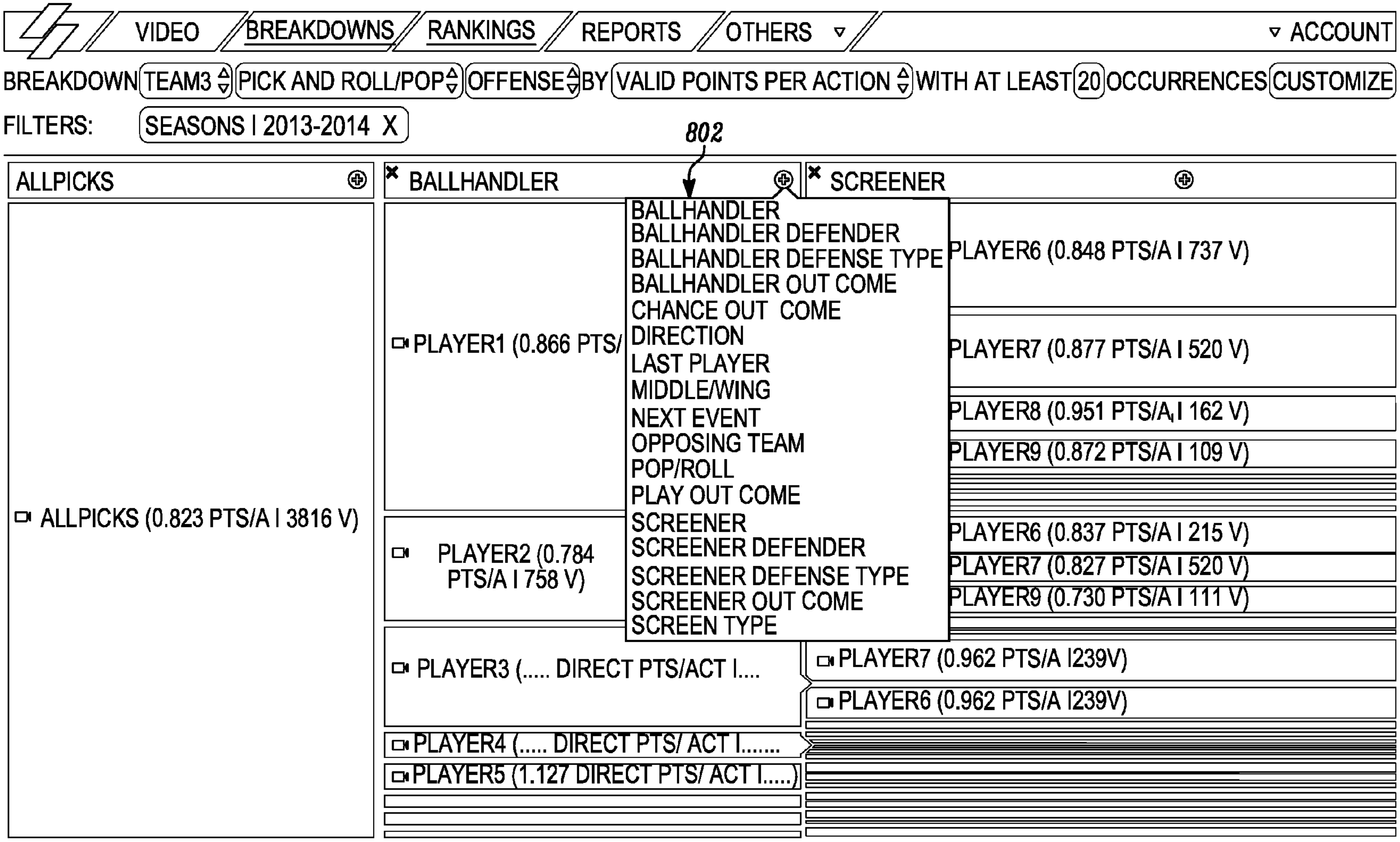
PLAYER4 (..... DIRECT PTS/ ACT I.......

SOFT (... DIRECT PTS/ACT II .... DIRECT ACTI.....

PLAYER5 (1.127 DIRECT PTS/ ACT I.......)

SOFT (... DIRECT PTS/ACT II .... DIRECT ACTI.....

FIG. 7



BREAKDOWN TEAM3 PICK AND ROLL/POP OFFENSE BY VALID POINTS PER ACTION WITH AT LEAST 20 OCCURRENCES

FILTERS: SEASONS I 2013-2014 X

ALLPICKS (0.823 PTS/A I 3816 V)

PLAYER1 (0.866 PTS/

PLAYER2 (0.784 PTS/A I 758 V)

PLAYER3 (..... DIRECT PTS/ACT I....

PLAYER4 (..... DIRECT PTS/ ACT I.......

PLAYER5 (1.127 DIRECT PTS/ ACT I.....)

PLAYER6 (0.848 PTS/A I 737 V)

PLAYER7 (0.877 PTS/A I 520 V)

PLAYER8 (0.951 PTS/A, I 162 V)

PLAYER9 (0.872 PTS/A I 109 V)

PLAYER6 (0.837 PTS/A I 215 V)

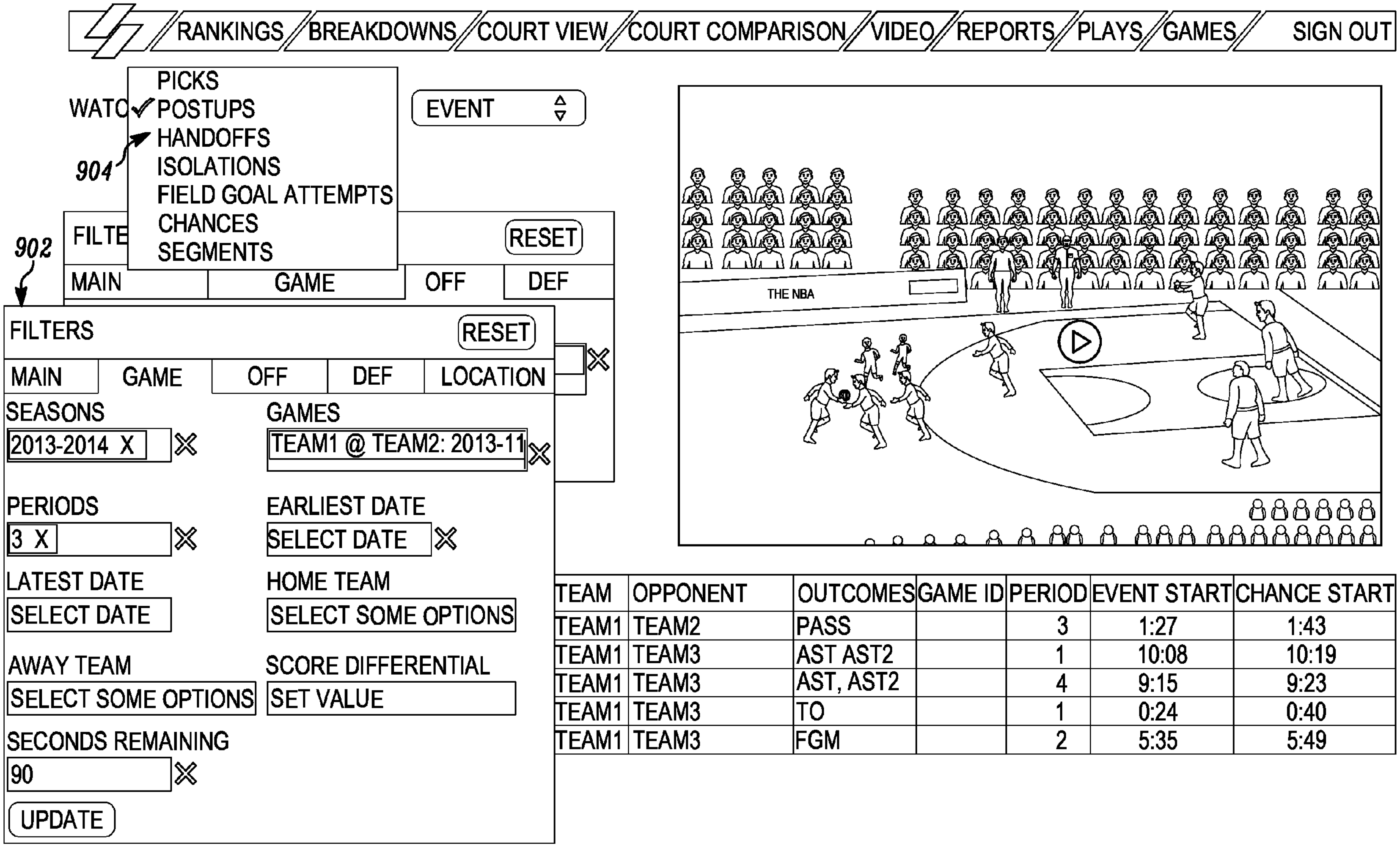
PLAYER7 (0.827 PTS/A I 520 V)

PLAYER9 (0.730 PTS/A I 111 V)

PLAYER7 (0.962 PTS/A I239V)

PLAYER6 (0.962 PTS/A I239V)

FIG. 8



SEASONS: 2013-2014 X

GAMES: TEAM1 @ TEAM2: 2013-11

PERIODS: 3 X

EARLIEST DATE: SELECT DATE

LATEST DATE: SELECT DATE

HOME TEAM: SELECT SOME OPTIONS

AWAY TEAM: SELECT SOME OPTIONS

SCORE DIFFERENTIAL: SET VALUE

SECONDS REMAINING: 90

TEAM	OPPONENT	OUTCOMES	GAME ID	PERIOD	EVENT START	CHANCE START
TEAM1	TEAM2	PASS		3	1:27	1:43
TEAM1	TEAM3	AST AST2		1	10:08	10:19
TEAM1	TEAM3	AST, AST2		4	9:15	9:23
TEAM1	TEAM3	TO		1	0:24	0:40
TEAM1	TEAM3	FGM		2	5:35	5:49

PRECISE GAME SITUATIONS

FIG. 9

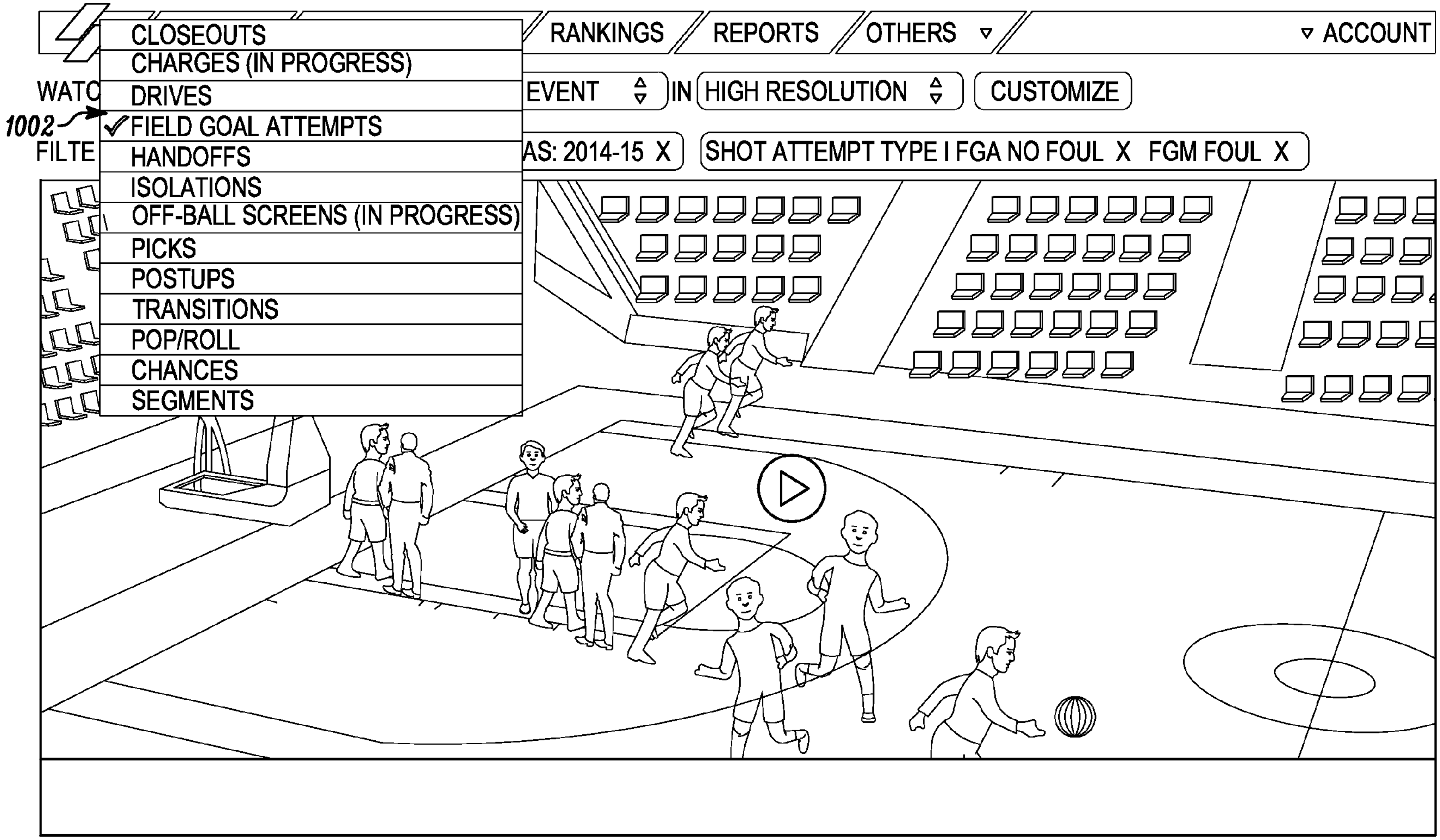
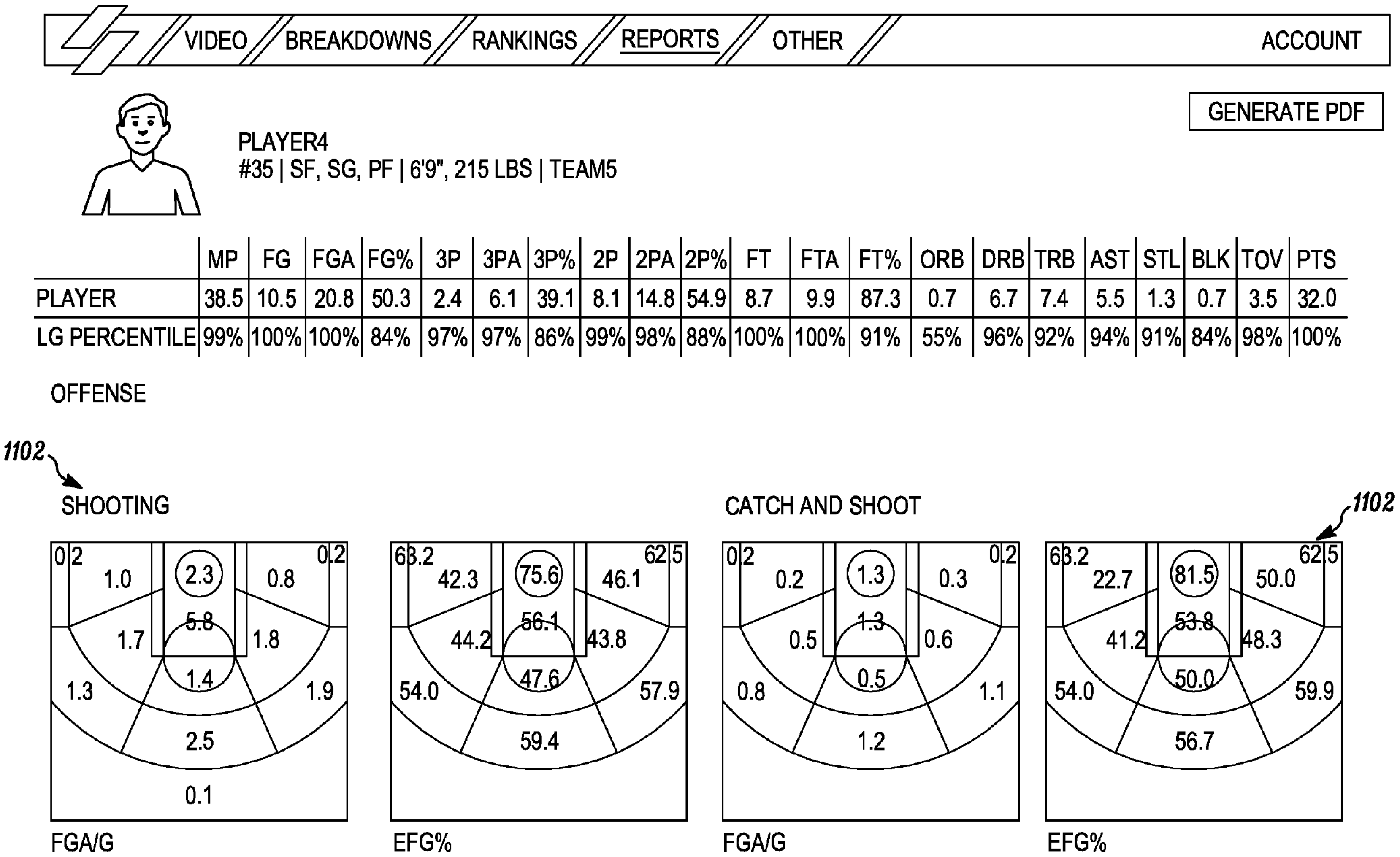

RANKINGS
REPORTS
OTHERS ▽
▽ ACCOUNT
WATC
FILTE
1002
CLOSEOUTS
CHARGES (IN PROGRESS)
DRIVES
✓FIELD GOAL ATTEMPTS
HANDOFFS
ISOLATIONS
OFF-BALL SCREENS (IN PROGRESS)
PICKS
POSTUPS
TRANSITIONS
POP/ROLL
CHANCES
SEGMENTS
EVENT
IN
HIGH RESOLUTION
CUSTOMIZE
AS: 2014-15 X
SHOT ATTEMPT TYPE I FGA NO FOUL X FGM FOUL X

FIG. 10



PLAYER4
#35 | SF, SG, PF | 6'9", 215 LBS | TEAM5

	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PTS
PLAYER	38.5	10.5	20.8	50.3	2.4	6.1	39.1	8.1	14.8	54.9	8.7	9.9	87.3	0.7	6.7	7.4	5.5	1.3	0.7	3.5	32.0
LG PERCENTILE	99%	100%	100%	84%	97%	97%	86%	99%	98%	88%	100%	100%	91%	55%	96%	92%	94%	91%	84%	98%	100%

OFFENSE

SHOOTING

CATCH AND SHOOT

FGA/G

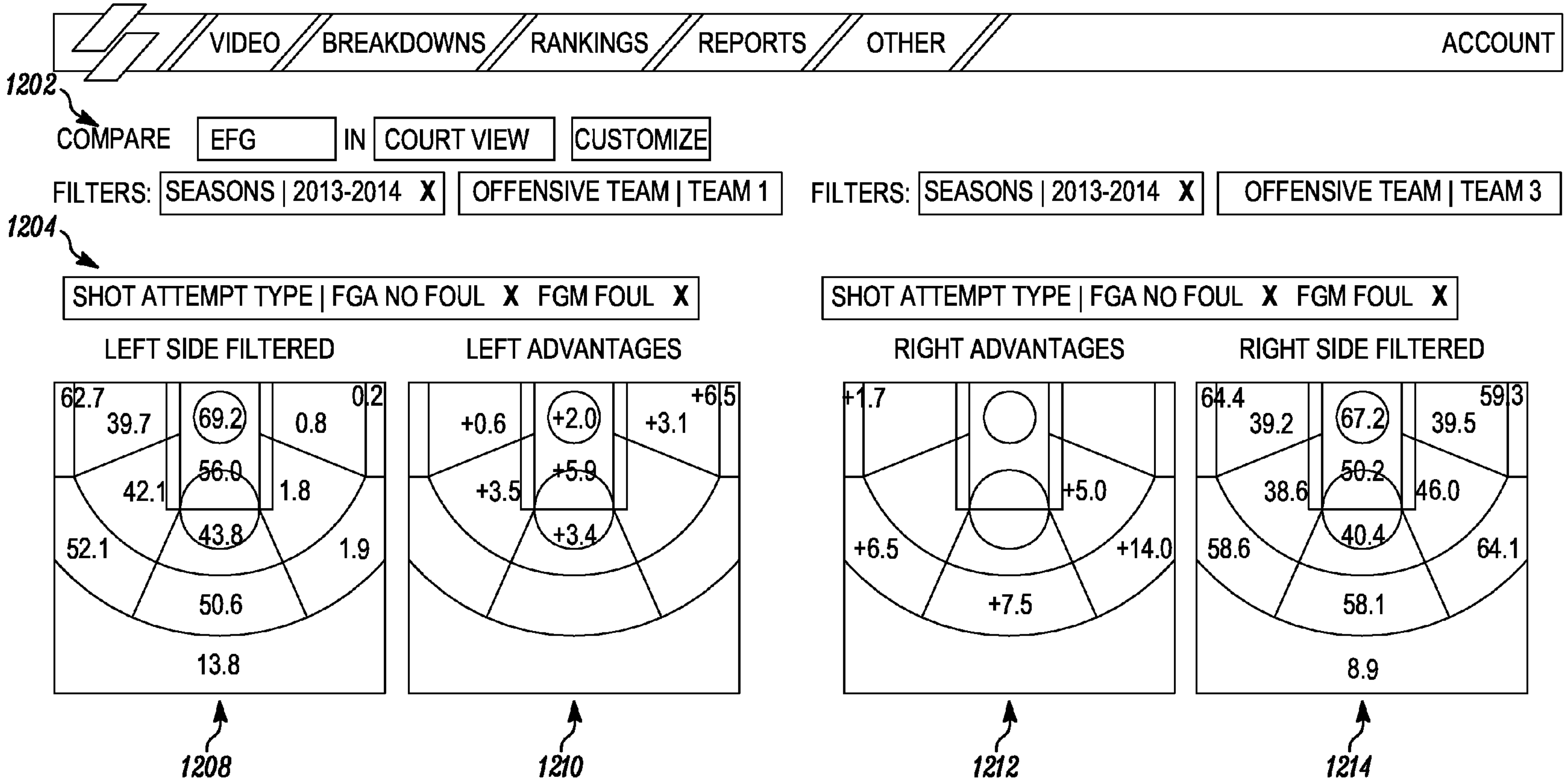
EFG%

FGA/G

EFG%

SHOTS BY LOCATION

FIG. 11



COMPARE EFG IN COURT VIEW CUSTOMIZE

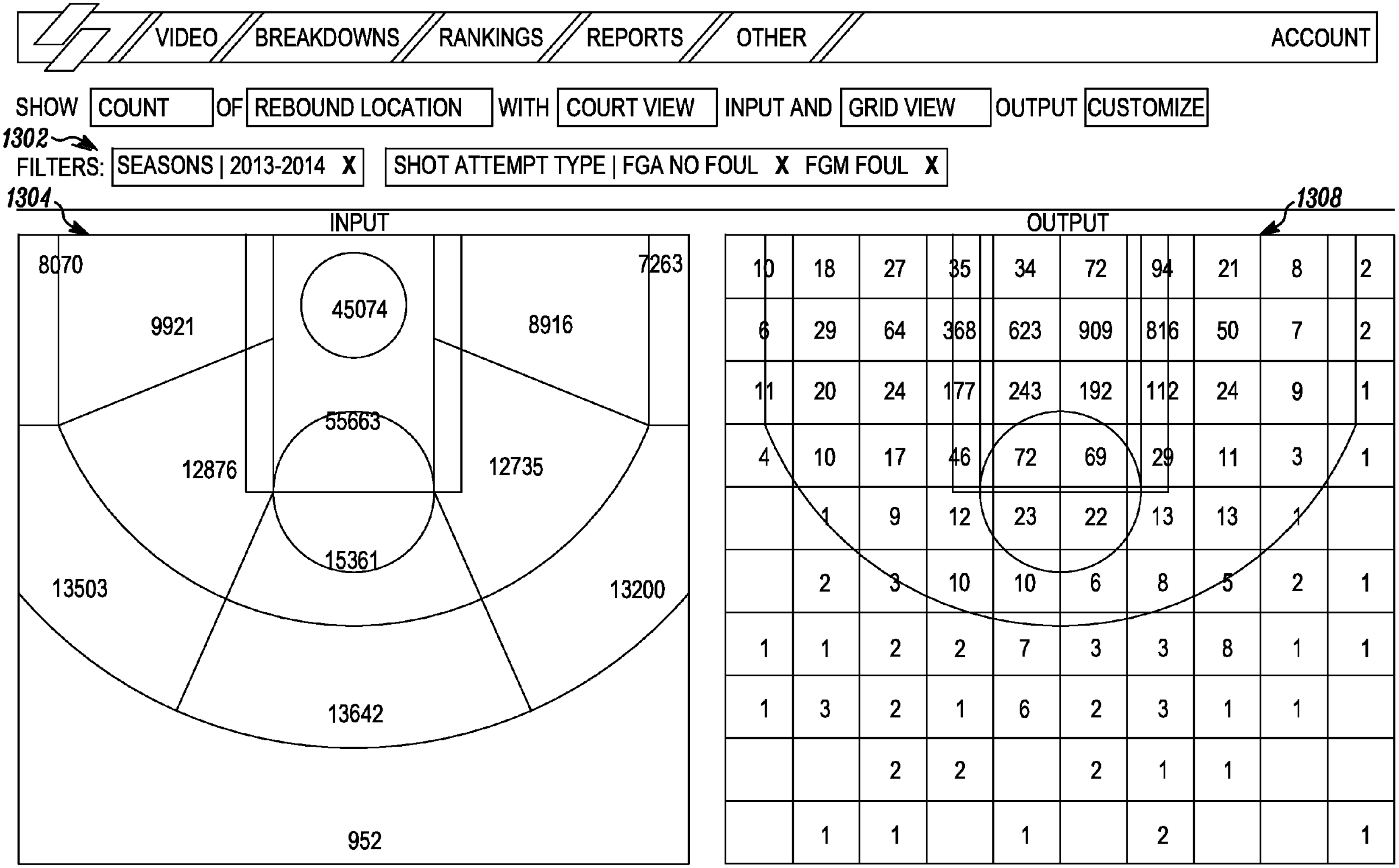
FILTERS: SEASONS | 2013-2014 X OFFENSIVE TEAM | TEAM 1

SHOT ATTEMPT TYPE | FGA NO FOUL X FGM FOUL X

FILTERS: SEASONS | 2013-2014 X OFFENSIVE TEAM | TEAM 3

SHOT ATTEMPT TYPE | FGA NO FOUL X FGM FOUL X

FIG. 12

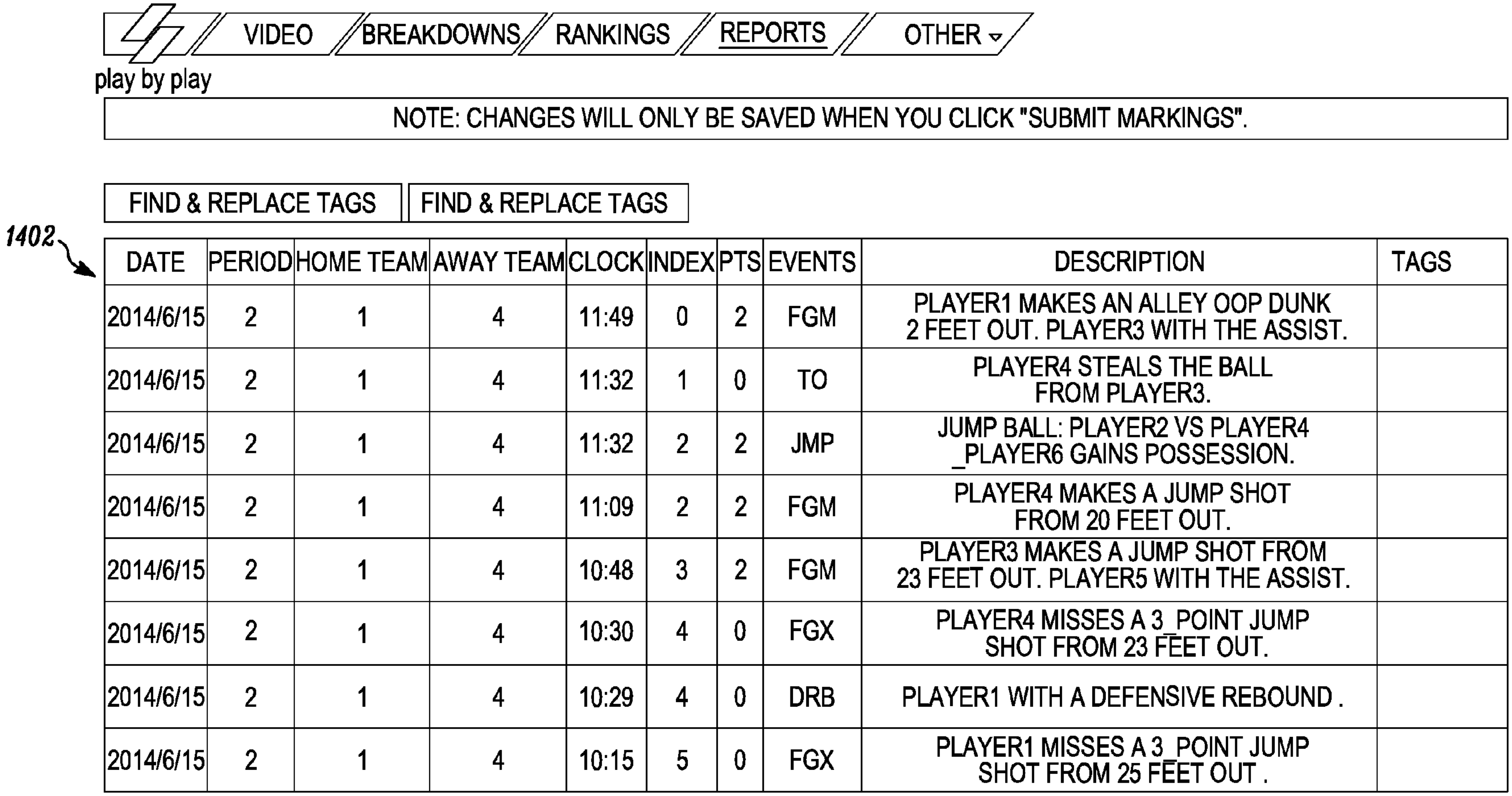


SHOW COUNT OF REBOUND LOCATION WITH COURT VIEW INPUT AND GRID VIEW OUTPUT CUSTOMIZE

FILTERS: SEASONS | 2013-2014 X SHOT ATTEMPT TYPE | FGA NO FOUL X FGM FOUL X

10	18	27	35	34	72	94	21	8	2
6	29	64	368	623	909	816	50	7	2
11	20	24	177	243	192	112	24	9	1
4	10	17	46	72	69	29	11	3	1
	1	9	12	23	22	13	13	1	
	2	3	10	10	6	8	5	2	1
1	1	2	2	7	3	3	8	1	1
1	3	2	1	6	2	3	1	1	
		2	2		2	1	1		
	1	1		1		2			1

FIG. 13



NOTE: CHANGES WILL ONLY BE SAVED WHEN YOU CLICK "SUBMIT MARKINGS".

DATE	PERIOD	HOME TEAM	AWAY TEAM	CLOCK	INDEX	PTS	EVENTS	DESCRIPTION	TAGS
2014/6/15	2	1	4	11:49	0	2	FGM	PLAYER1 MAKES AN ALLEY OOP DUNK 2 FEET OUT. PLAYER3 WITH THE ASSIST.	
2014/6/15	2	1	4	11:32	1	0	TO	PLAYER4 STEALS THE BALL FROM PLAYER3.	
2014/6/15	2	1	4	11:32	2	2	JMP	JUMP BALL: PLAYER2 VS PLAYER4 _PLAYER6 GAINS POSSESSION.	
2014/6/15	2	1	4	11:09	2	2	FGM	PLAYER4 MAKES A JUMP SHOT FROM 20 FEET OUT.	
2014/6/15	2	1	4	10:48	3	2	FGM	PLAYER3 MAKES A JUMP SHOT FROM 23 FEET OUT. PLAYER5 WITH THE ASSIST.	
2014/6/15	2	1	4	10:30	4	0	FGX	PLAYER4 MISSES A 3_POINT JUMP SHOT FROM 23 FEET OUT.	
2014/6/15	2	1	4	10:29	4	0	DRB	PLAYER1 WITH A DEFENSIVE REBOUND .	
2014/6/15	2	1	4	10:15	5	0	FGX	PLAYER1 MISSES A 3_POINT JUMP SHOT FROM 25 FEET OUT .	

FIG. 14

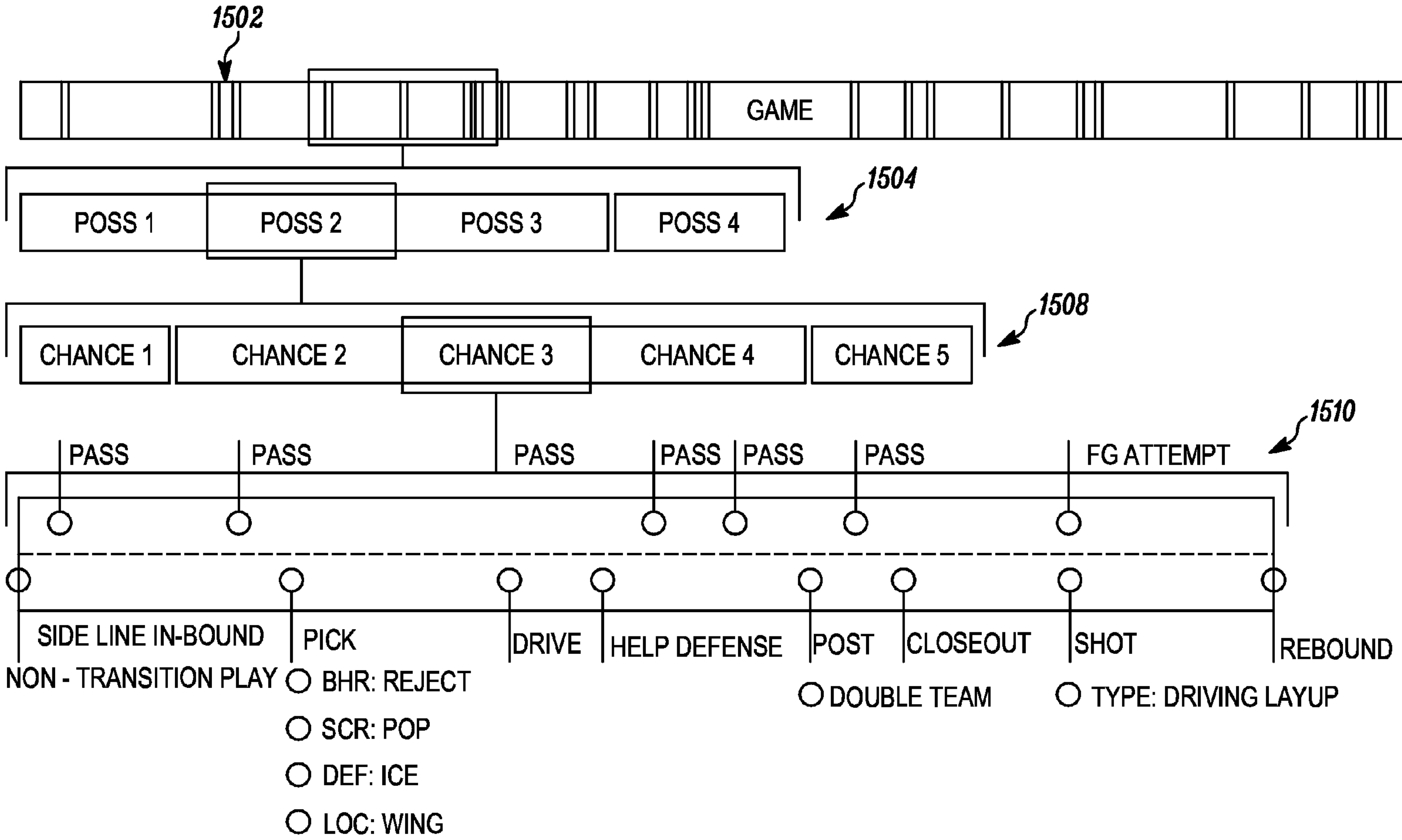

1502
GAME
1504
POSS 1
POSS 2
POSS 3
POSS 4
1508
CHANCE 1
CHANCE 2
CHANCE 3
CHANCE 4
CHANCE 5
1510
PASS
PASS
PASS
PASS
PASS
PASS
FG ATTEMPT
SIDE LINE IN-BOUND
NON - TRANSITION PLAY
PICK
BHR: REJECT
SCR: POP
DEF: ICE
LOC: WING
DRIVE
HELP DEFENSE
POST
DOUBLE TEAM
CLOSEOUT
SHOT
TYPE: DRIVING LAYUP
REBOUND

FIG. 15

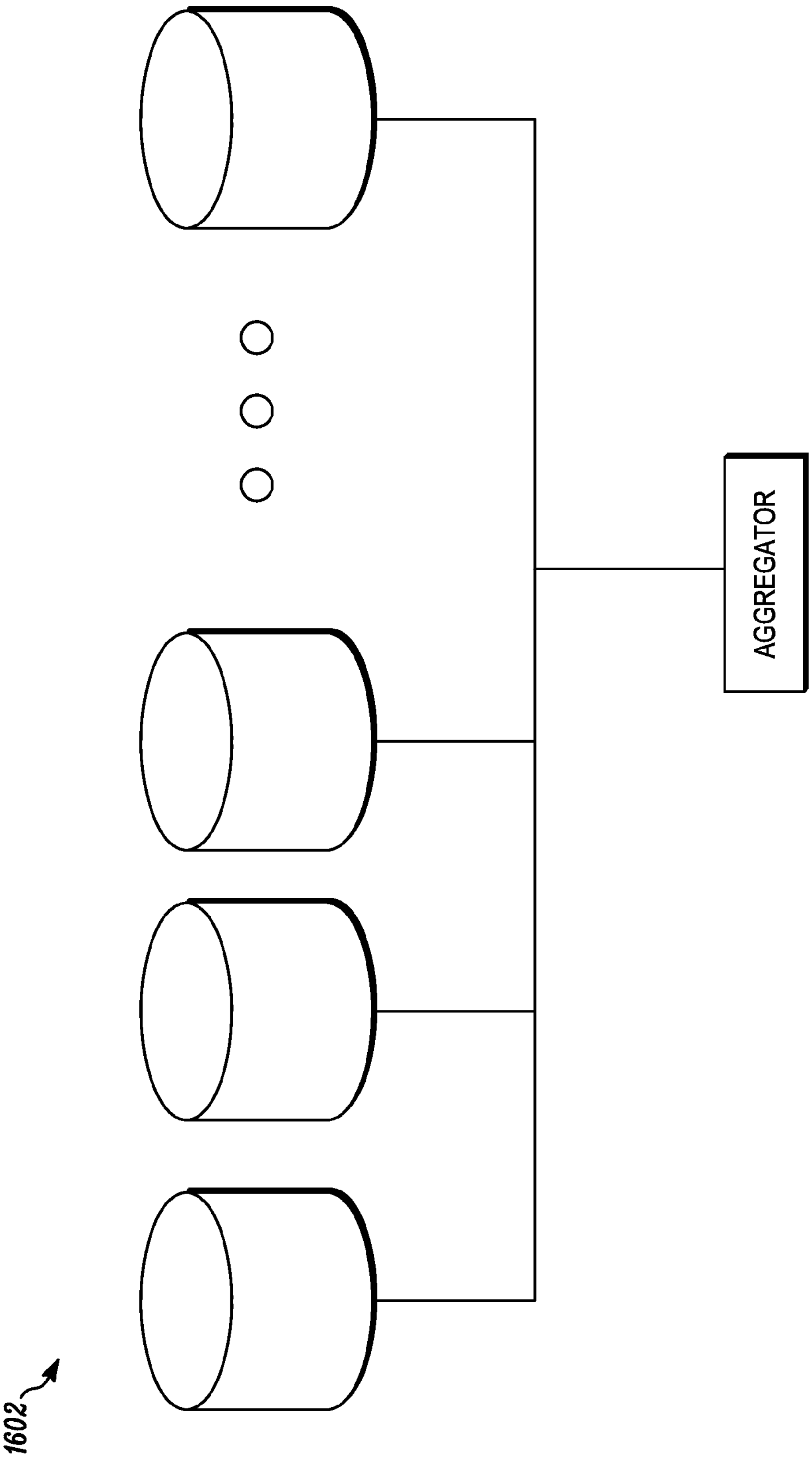


FIG. 16

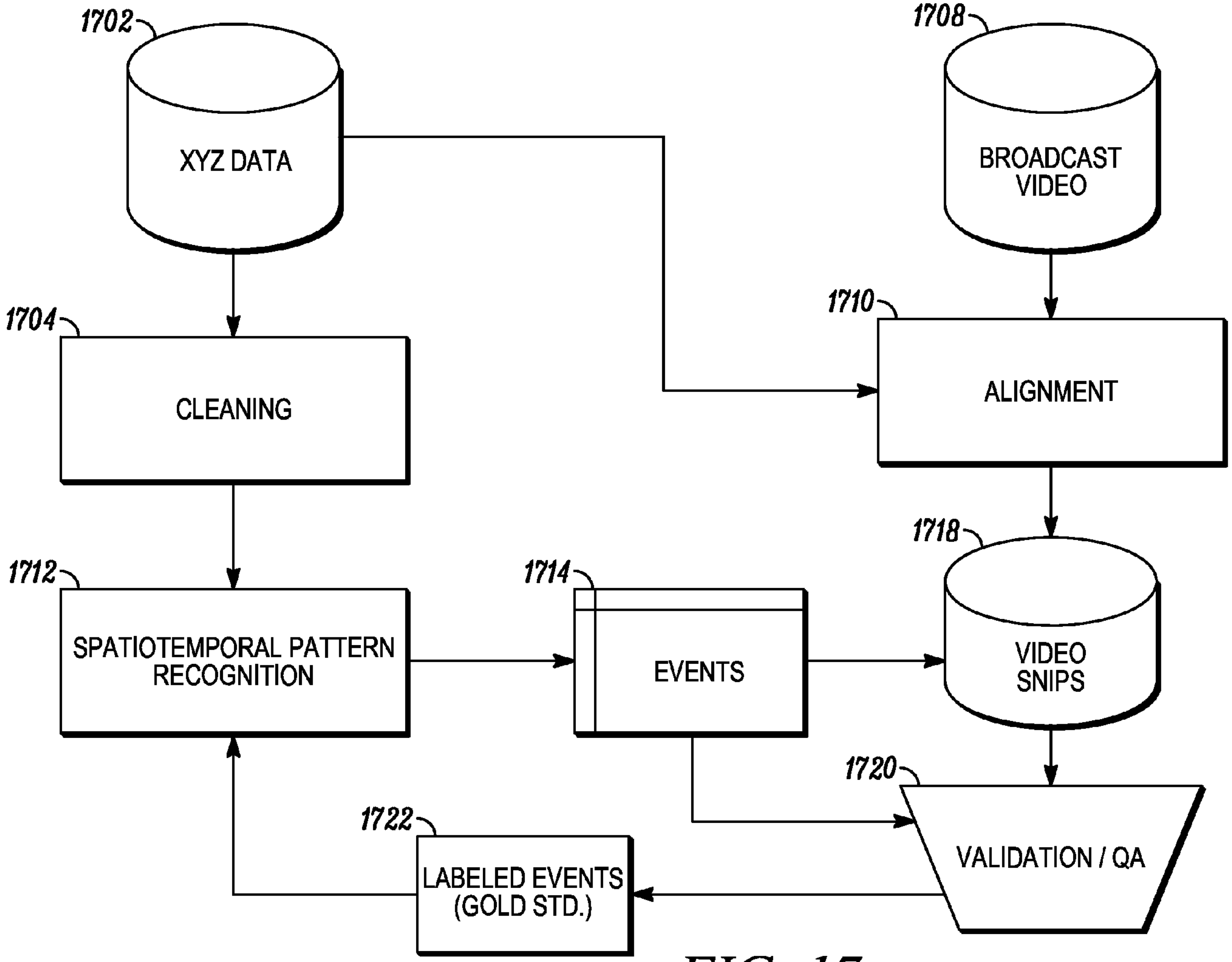

1702
XYZ DATA
1708
BROADCAST VIDEO
1704
CLEANING
1710
ALIGNMENT
1718
VIDEO SNIPS
1712
SPATIOTEMPORAL PATTERN RECOGNITION
1714
EVENTS
1720
VALIDATION / QA
1722
LABELED EVENTS (GOLD STD.)

FIG. 17

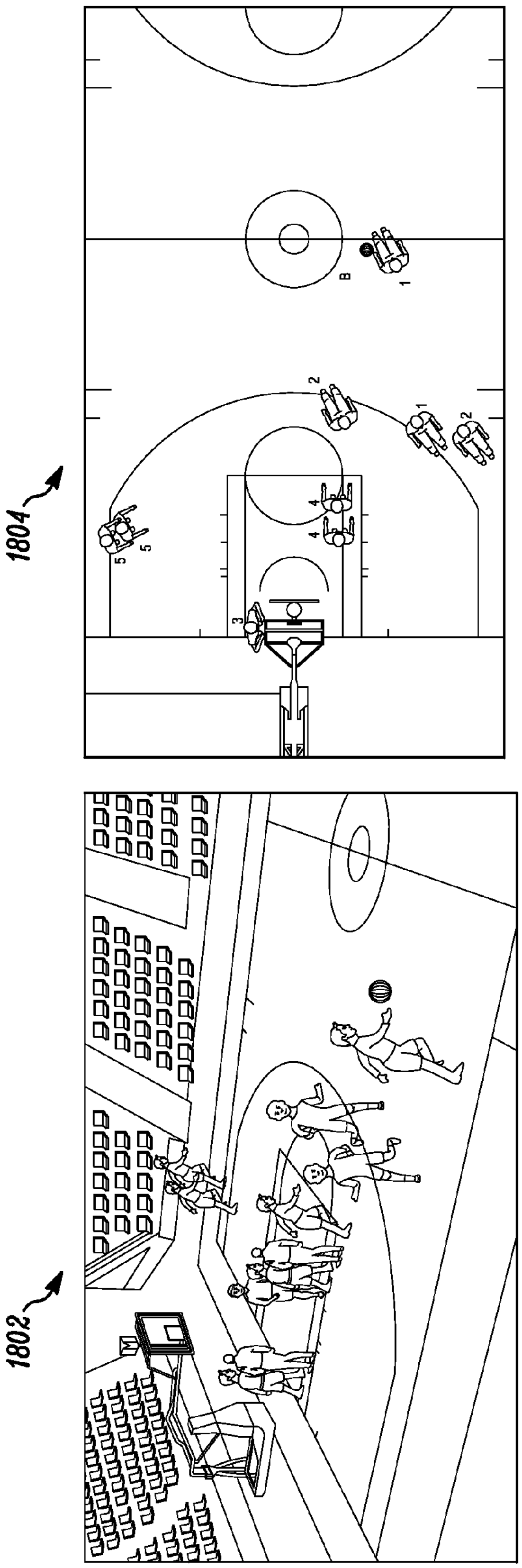


FIG. 18

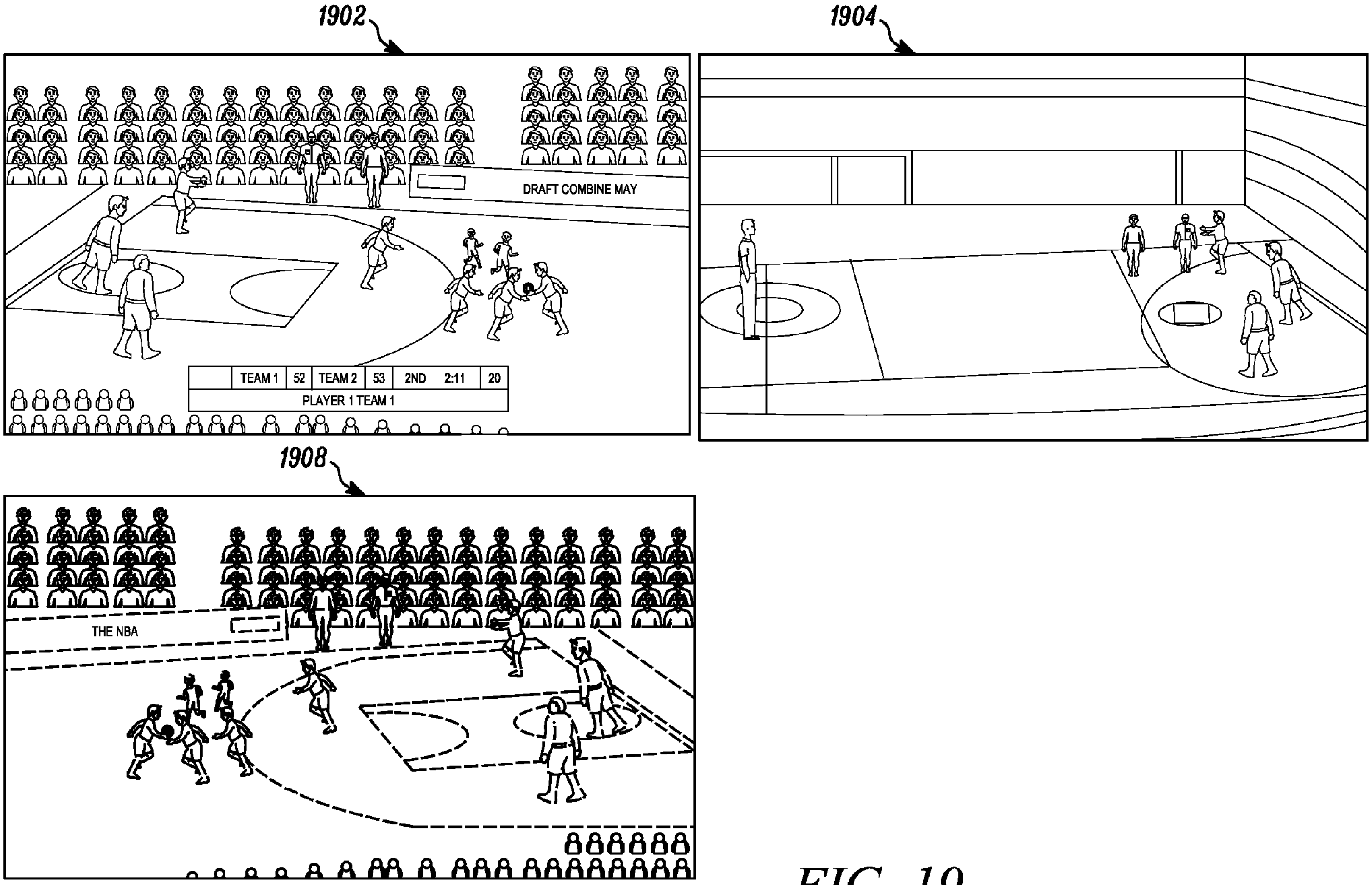

1902
1904
DRAFT COMBINE MAY
TEAM 1
52
TEAM 2
53
2ND
2:11
20
PLAYER 1 TEAM 1
1908
THE NBA

FIG. 19

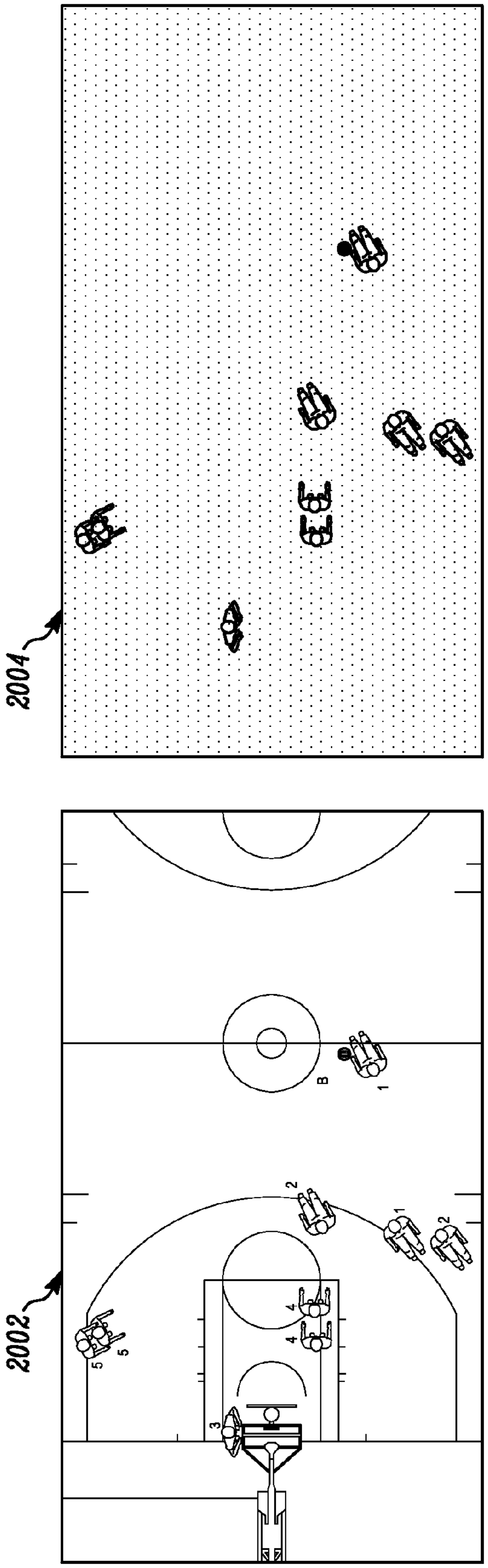


FIG. 20

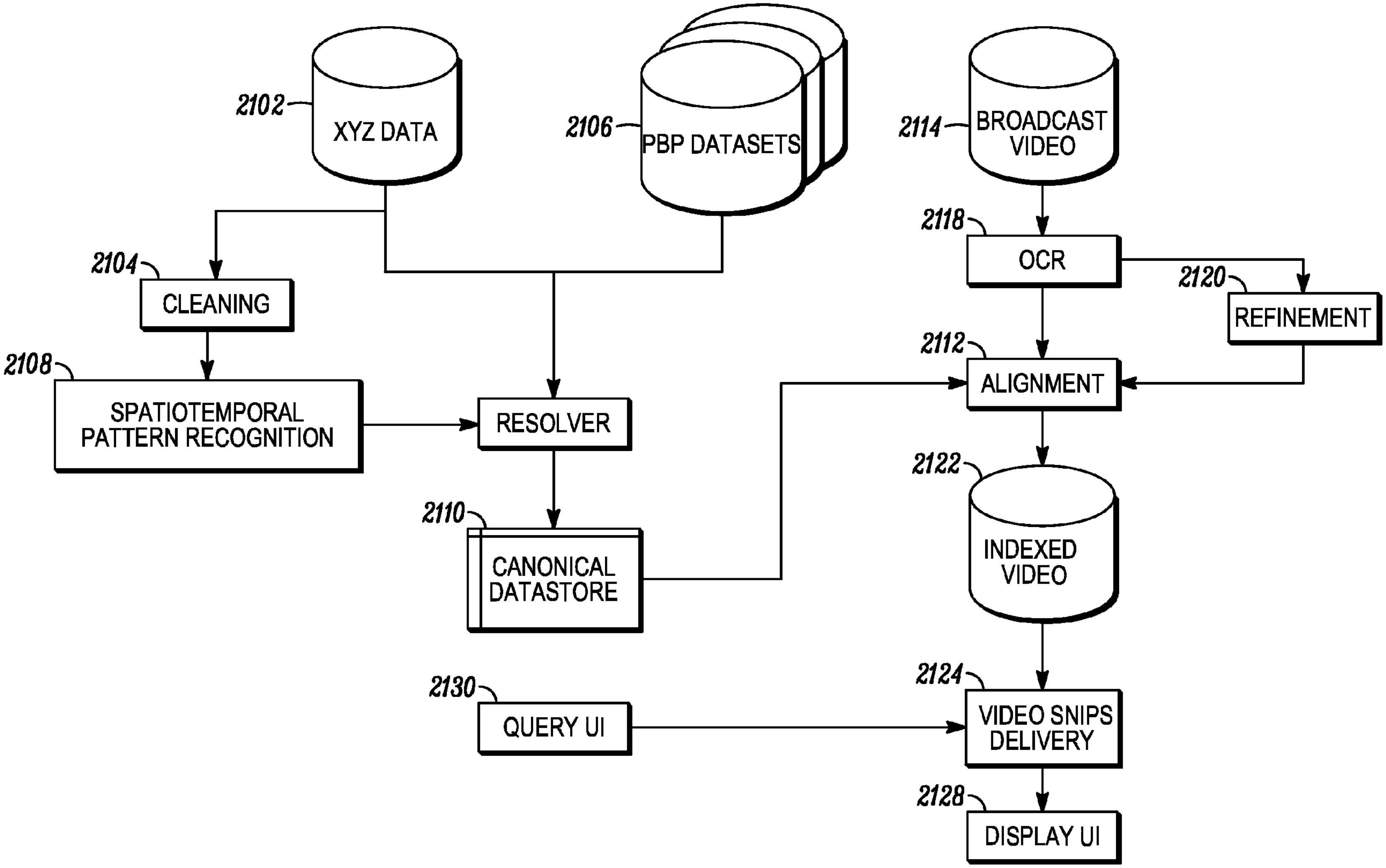

2102
XYZ DATA
2106
PBP DATASETS
2114
BROADCAST VIDEO
2118
OCR
2120
REFINEMENT
2104
CLEANING
2108
SPATIOTEMPORAL PATTERN RECOGNITION
RESOLVER
2112
ALIGNMENT
2122
INDEXED VIDEO
2110
CANONICAL DATASTORE
2124
VIDEO SNIPS DELIVERY
2130
QUERY UI
2128
DISPLAY UI

FIG. 21

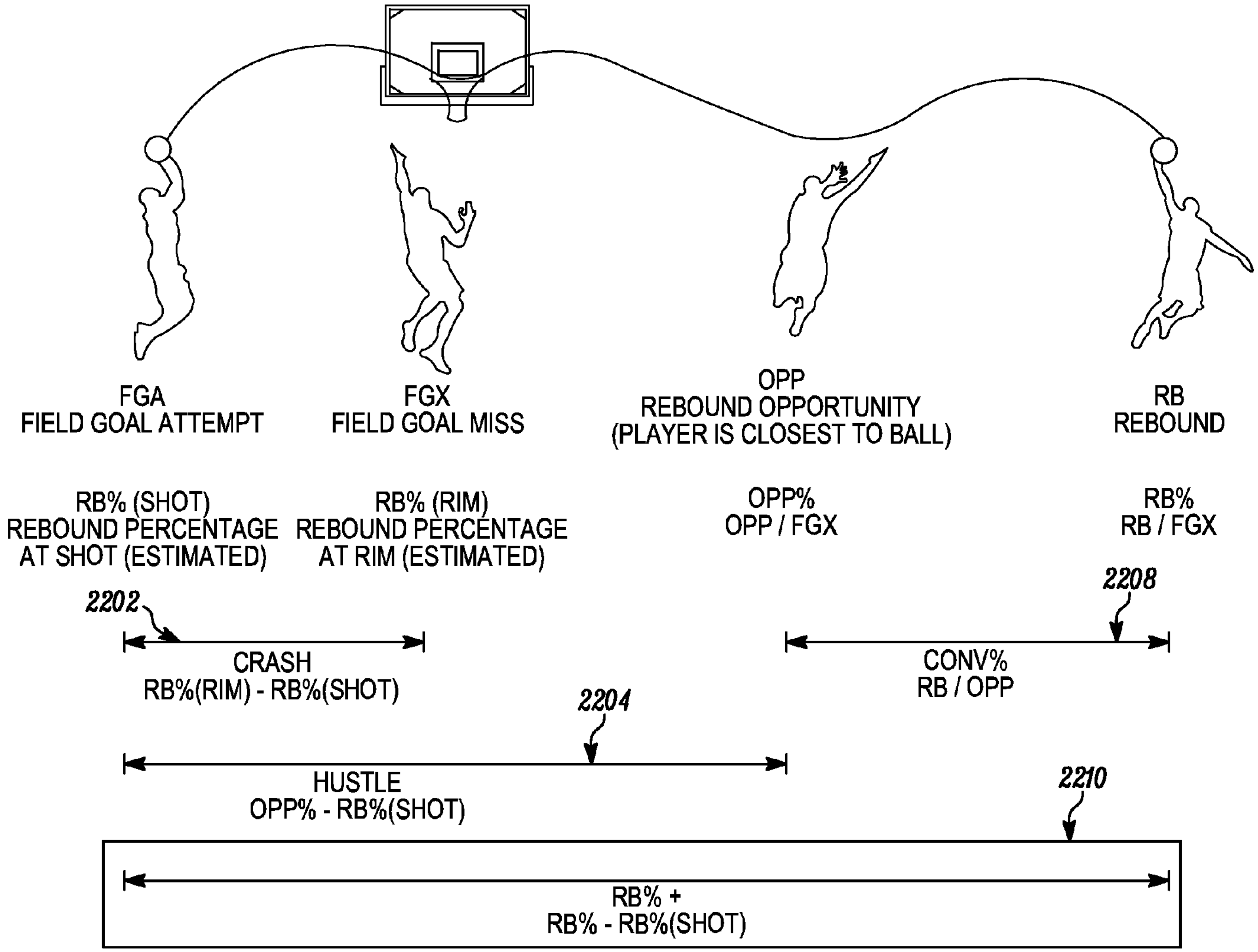
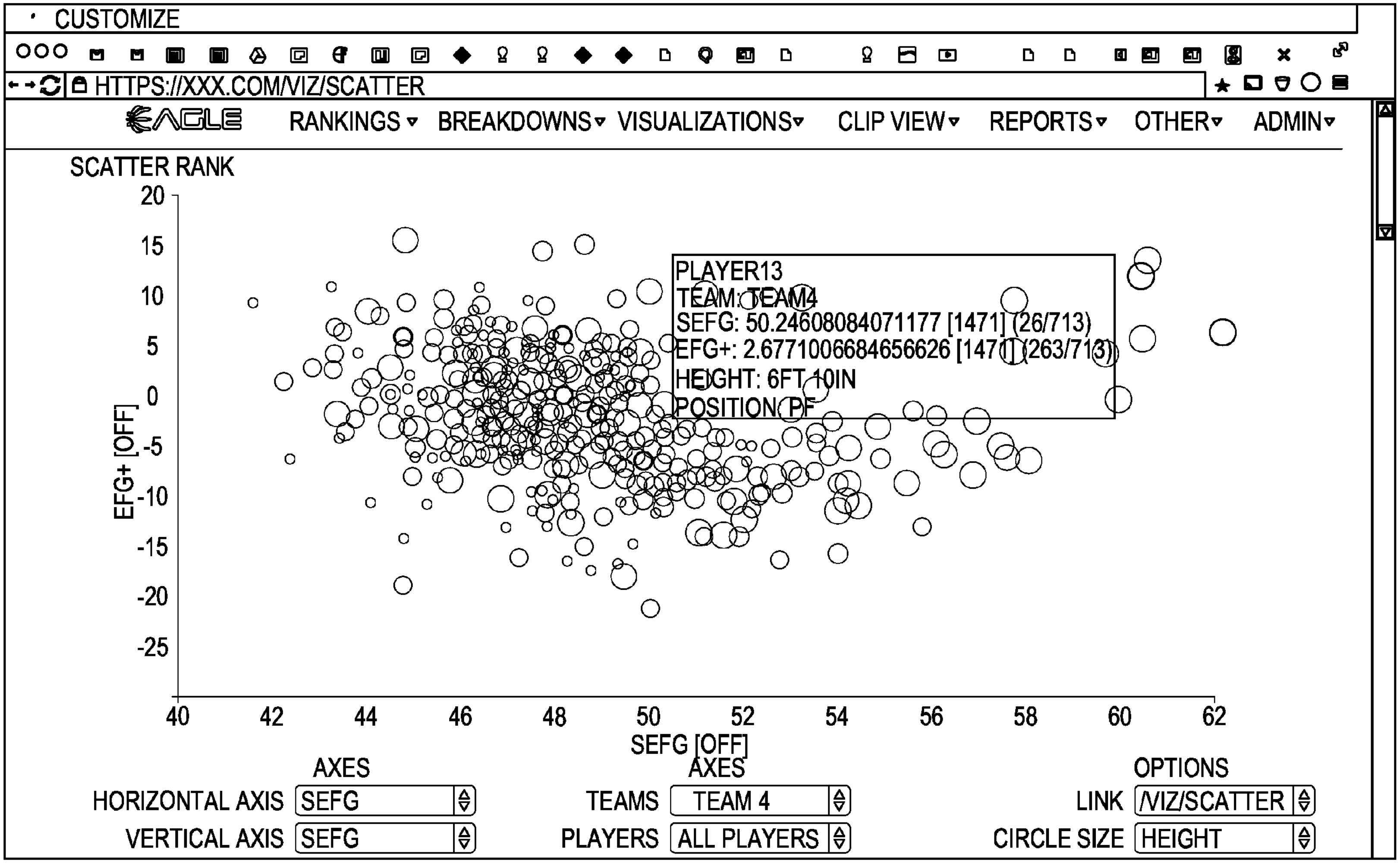

FGA
FIELD GOAL ATTEMPT
FGX
FIELD GOAL MISS
OPP
REBOUND OPPORTUNITY
(PLAYER IS CLOSEST TO BALL)
RB
REBOUND
RB% (SHOT)
REBOUND PERCENTAGE
AT SHOT (ESTIMATED)
RB% (RIM)
REBOUND PERCENTAGE
AT RIM (ESTIMATED)
OPP%
OPP / FGX
RB%
RB / FGX
2202
CRASH
RB%(RIM) - RB%(SHOT)
2208
CONV%
RB / OPP
2204
HUSTLE
OPP% - RB%(SHOT)
2210
RB% +
RB% - RB%(SHOT)

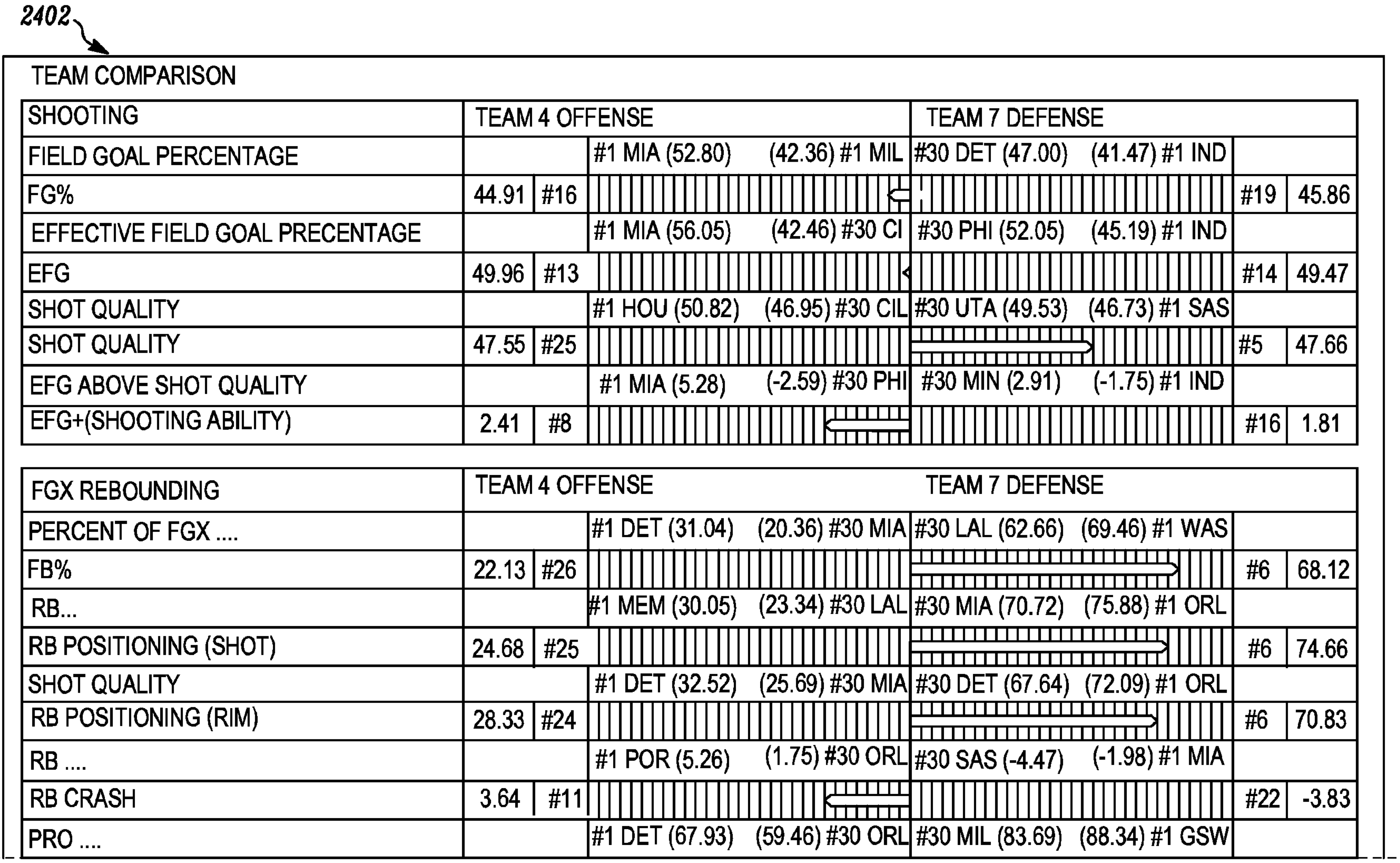
FIG. 22



PLAYER13
TEAM: TEAM4
SEFG: 50.24608084071177 [1471] (26/713)
EFG+: 2.6771006684656626 [1471] (263/713)
HEIGHT: 6FT 10IN
POSITION: PF

HORIZONTAL AXIS: SEFG
VERTICAL AXIS: SEFG
TEAMS: TEAM 4
PLAYERS: ALL PLAYERS
LINK: /VIZ/SCATTER
CIRCLE SIZE: HEIGHT

FIG. 23

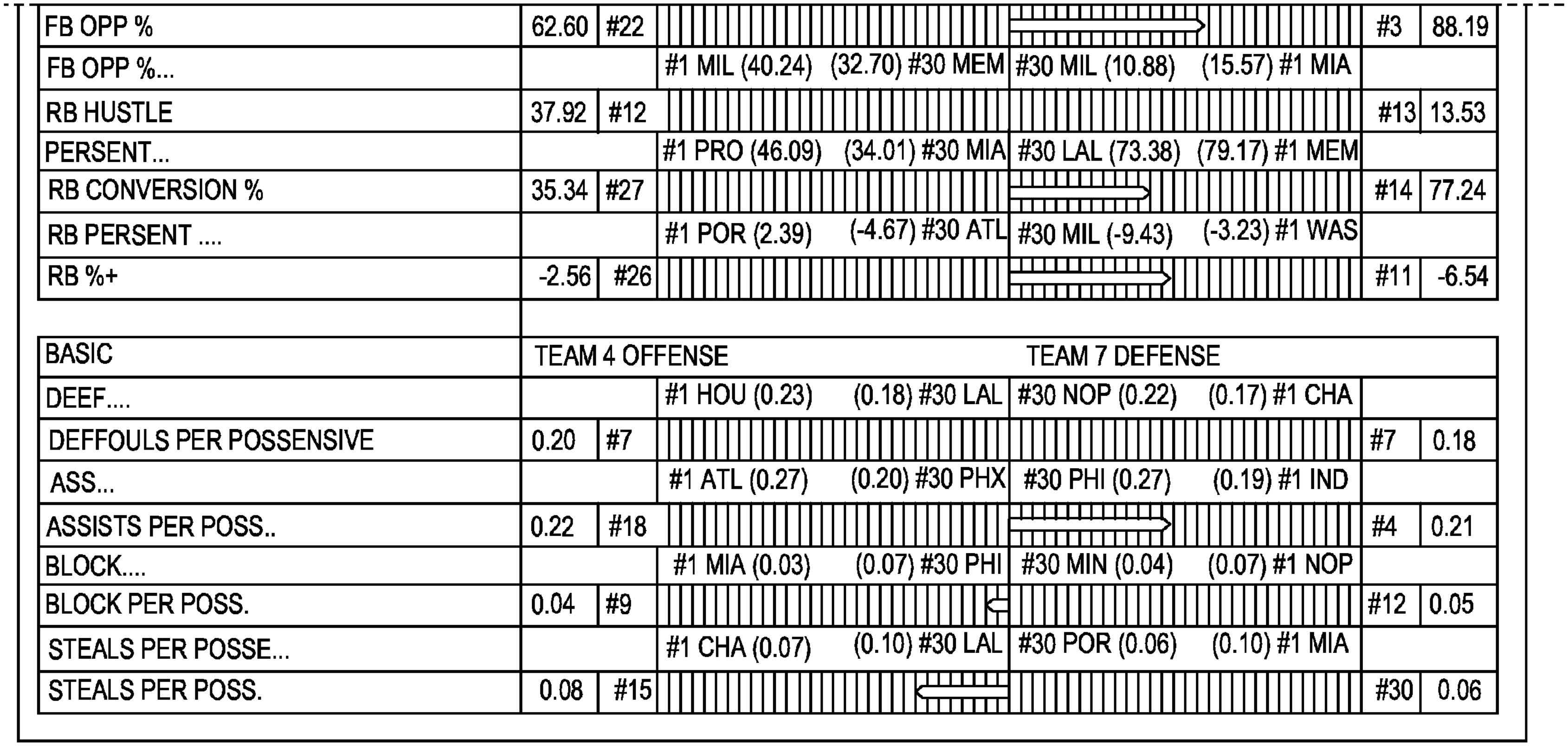


TEAM COMPARISON

SHOOTING	TEAM 4 OFFENSE			TEAM 7 DEFENSE		
FIELD GOAL PERCENTAGE			#1 MIA (52.80) (42.36) #1 MIL	#30 DET (47.00) (41.47) #1 IND		
FG%	44.91	#16			#19	45.86
EFFECTIVE FIELD GOAL PRECENTAGE			#1 MIA (56.05) (42.46) #30 CI	#30 PHI (52.05) (45.19) #1 IND		
EFG	49.96	#13			#14	49.47
SHOT QUALITY			#1 HOU (50.82) (46.95) #30 CIL	#30 UTA (49.53) (46.73) #1 SAS		
SHOT QUALITY	47.55	#25			#5	47.66
EFG ABOVE SHOT QUALITY			#1 MIA (5.28) (-2.59) #30 PHI	#30 MIN (2.91) (-1.75) #1 IND		
EFG+(SHOOTING ABILITY)	2.41	#8			#16	1.81

FGX REBOUNDING	TEAM 4 OFFENSE			TEAM 7 DEFENSE		
PERCENT OF FGX			#1 DET (31.04) (20.36) #30 MIA	#30 LAL (62.66) (69.46) #1 WAS		
FB%	22.13	#26			#6	68.12
RB...			#1 MEM (30.05) (23.34) #30 LAL	#30 MIA (70.72) (75.88) #1 ORL		
RB POSITIONING (SHOT)	24.68	#25			#6	74.66
SHOT QUALITY			#1 DET (32.52) (25.69) #30 MIA	#30 DET (67.64) (72.09) #1 ORL		
RB POSITIONING (RIM)	28.33	#24			#6	70.83
RB			#1 POR (5.26) (1.75) #30 ORL	#30 SAS (-4.47) (-1.98) #1 MIA		
RB CRASH	3.64	#11			#22	-3.83
PRO			#1 DET (67.93) (59.46) #30 ORL	#30 MIL (83.69) (88.34) #1 GSW		

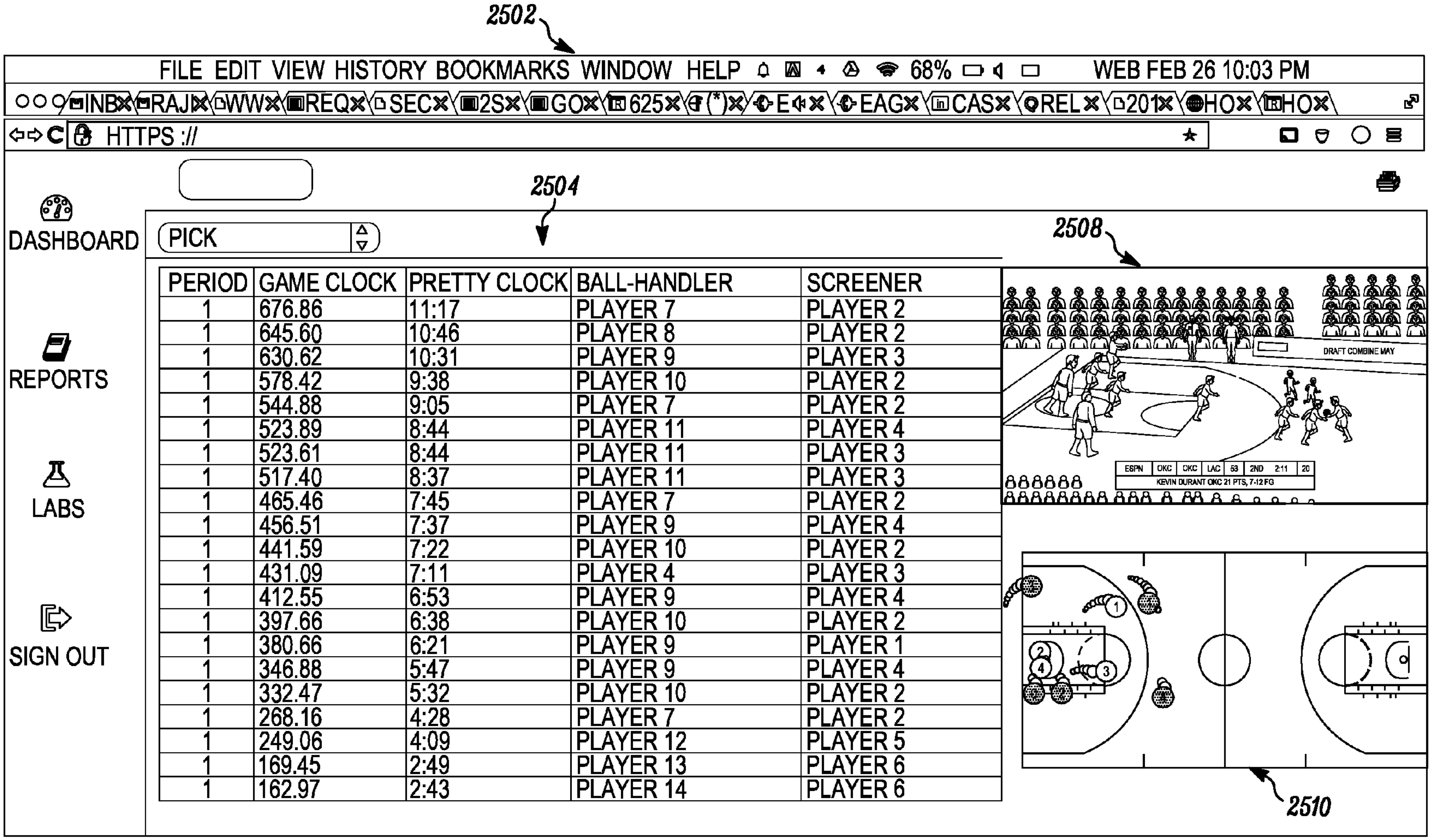
FIG. 24A



FB OPP %	62.60	#22			#3	88.19
FB OPP %...			#1 MIL (40.24) (32.70) #30 MEM	#30 MIL (10.88) (15.57) #1 MIA		
RB HUSTLE	37.92	#12			#13	13.53
PERSENT...			#1 PRO (46.09) (34.01) #30 MIA	#30 LAL (73.38) (79.17) #1 MEM		
RB CONVERSION %	35.34	#27			#14	77.24
RB PERSENT			#1 POR (2.39) (-4.67) #30 ATL	#30 MIL (-9.43) (-3.23) #1 WAS		
RB %+	-2.56	#26			#11	-6.54

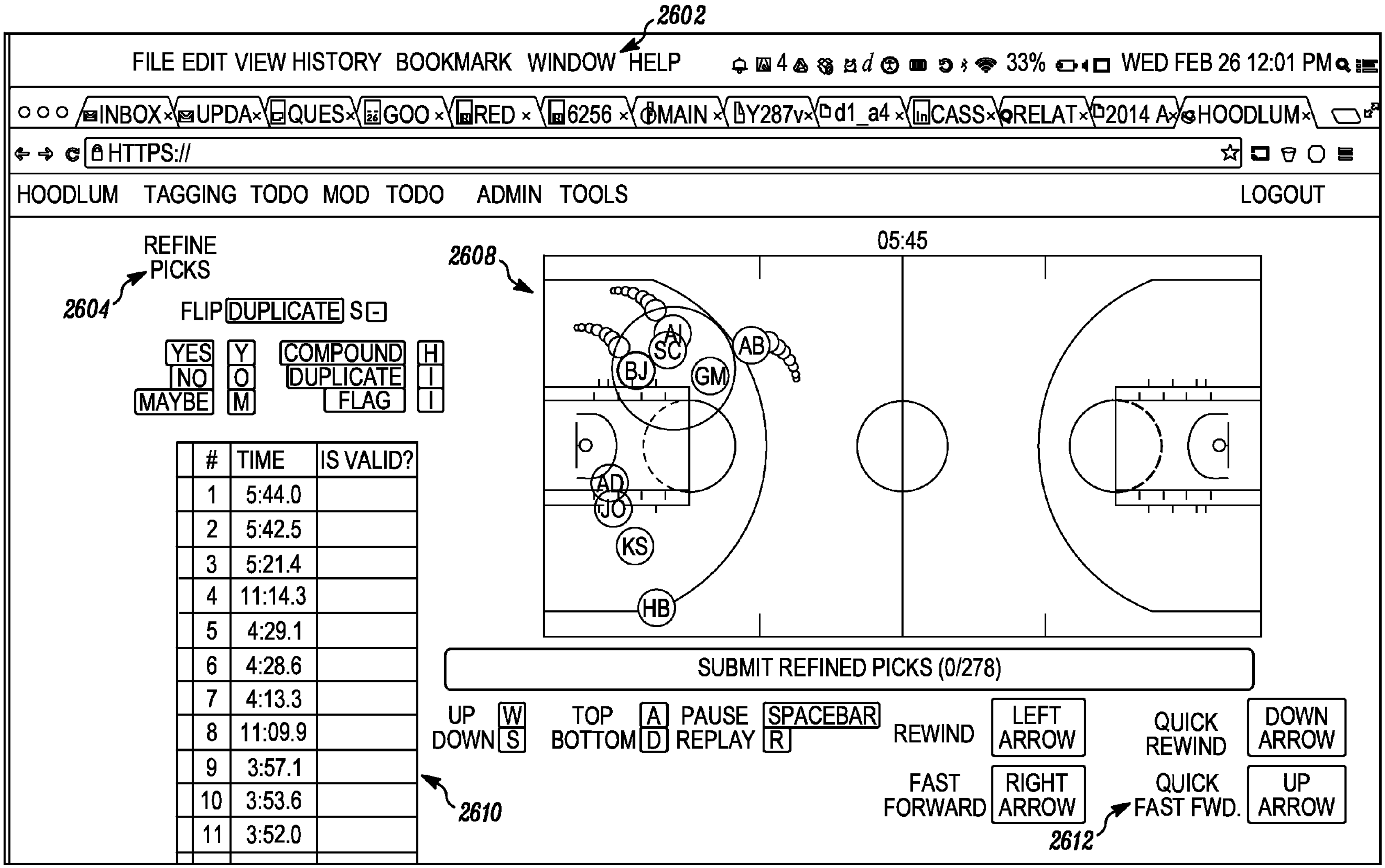
BASIC	TEAM 4 OFFENSE			TEAM 7 DEFENSE		
DEEF....			#1 HOU (0.23) (0.18) #30 LAL	#30 NOP (0.22) (0.17) #1 CHA		
DEFFOULS PER POSSENSIVE	0.20	#7			#7	0.18
ASS...			#1 ATL (0.27) (0.20) #30 PHX	#30 PHI (0.27) (0.19) #1 IND		
ASSISTS PER POSS..	0.22	#18			#4	0.21
BLOCK....			#1 MIA (0.03) (0.07) #30 PHI	#30 MIN (0.04) (0.07) #1 NOP		
BLOCK PER POSS.	0.04	#9			#12	0.05
STEALS PER POSSE...			#1 CHA (0.07) (0.10) #30 LAL	#30 POR (0.06) (0.10) #1 MIA		
STEALS PER POSS.	0.08	#15			#30	0.06

FIG. 24B


2502
FILE EDIT VIEW HISTORY BOOKMARKS WINDOW HELP 68% WEB FEB 26 10:03 PM
INB RAJ WW REQ SEC 2S GO 625 (*) E EAG CAS REL 201 HO HO
HTTPS ://
DASHBOARD
REPORTS
LABS
SIGN OUT
2504
PICK
2508
DRAFT COMBINE MAY
ESPN OKC OKC LAC 53 2ND 2:11 20
KEVIN DURANT OKC 21 PTS, 7-12 FG
1
2
3
4
2510

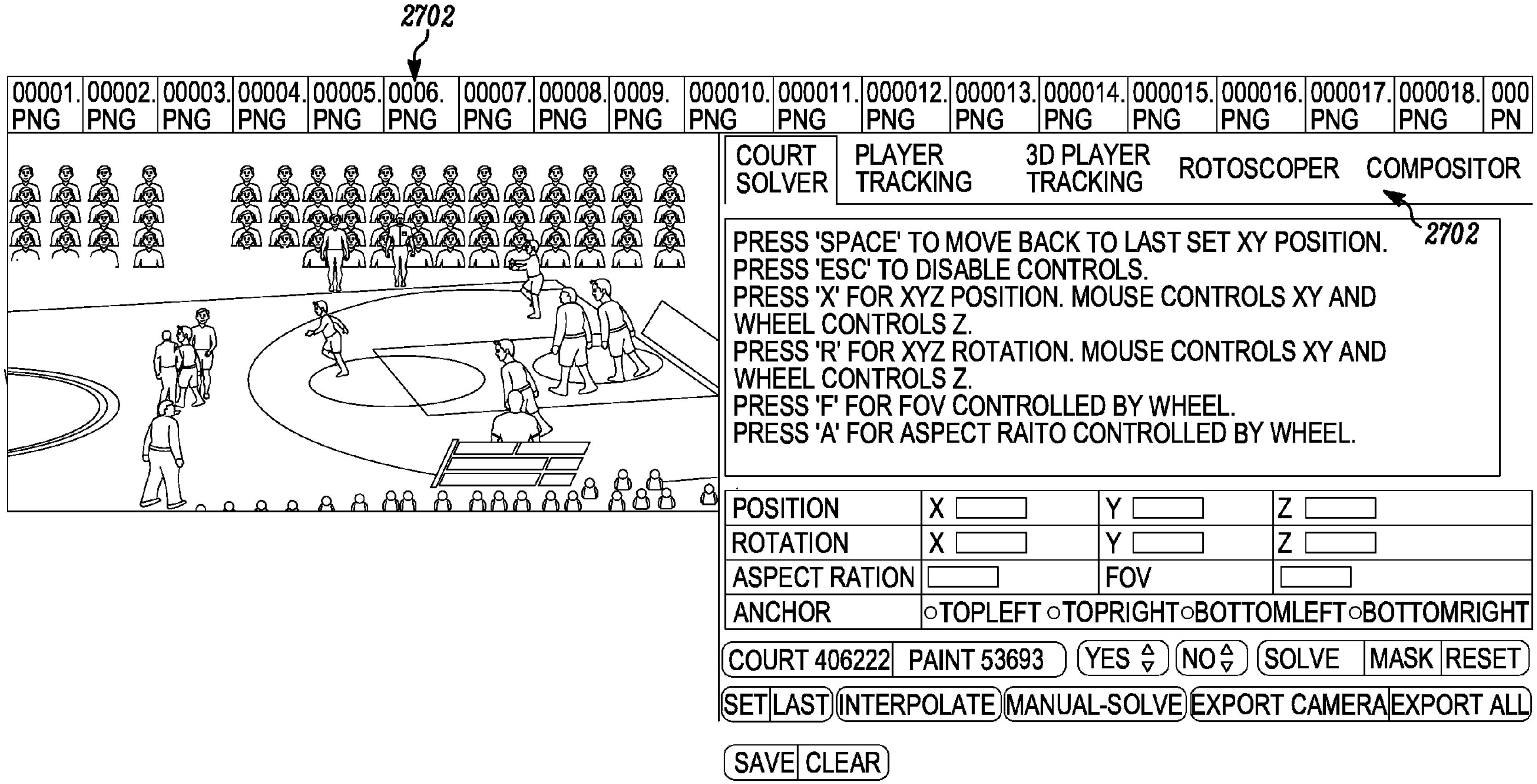
PERIOD	GAME CLOCK	PRETTY CLOCK	BALL-HANDLER	SCREENER
1	676.86	11:17	PLAYER 7	PLAYER 2
1	645.60	10:46	PLAYER 8	PLAYER 2
1	630.62	10:31	PLAYER 9	PLAYER 3
1	578.42	9:38	PLAYER 10	PLAYER 2
1	544.88	9:05	PLAYER 7	PLAYER 2
1	523.89	8:44	PLAYER 11	PLAYER 4
1	523.61	8:44	PLAYER 11	PLAYER 3
1	517.40	8:37	PLAYER 11	PLAYER 3
1	465.46	7:45	PLAYER 7	PLAYER 2
1	456.51	7:37	PLAYER 9	PLAYER 4
1	441.59	7:22	PLAYER 10	PLAYER 2
1	431.09	7:11	PLAYER 4	PLAYER 3
1	412.55	6:53	PLAYER 9	PLAYER 4
1	397.66	6:38	PLAYER 10	PLAYER 2
1	380.66	6:21	PLAYER 9	PLAYER 1
1	346.88	5:47	PLAYER 9	PLAYER 4
1	332.47	5:32	PLAYER 10	PLAYER 2
1	268.16	4:28	PLAYER 7	PLAYER 2
1	249.06	4:09	PLAYER 12	PLAYER 5
1	169.45	2:49	PLAYER 13	PLAYER 6
1	162.97	2:43	PLAYER 14	PLAYER 6

FIG. 25



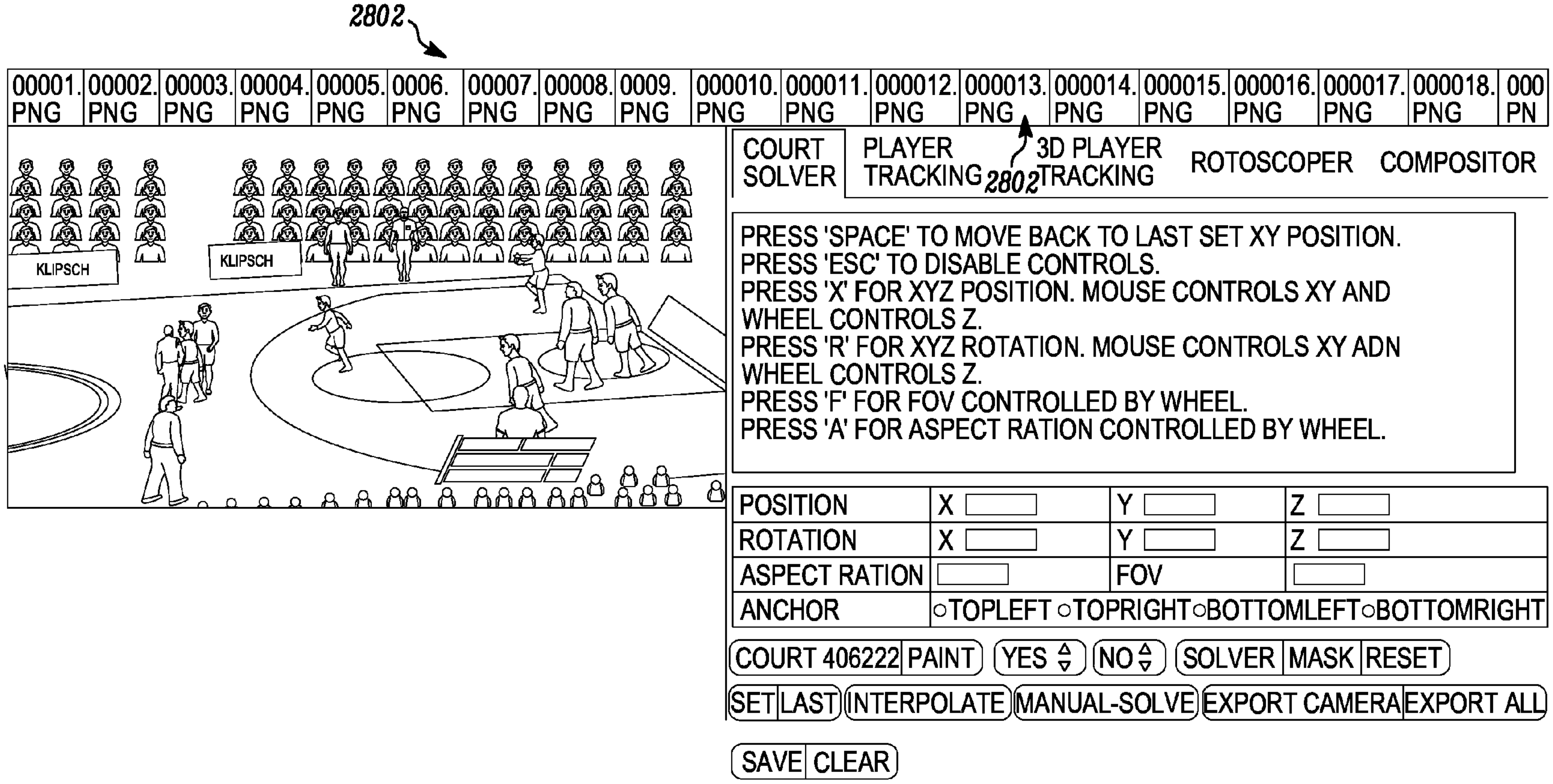
	#	TIME	IS VALID?
	1	5:44.0	
	2	5:42.5	
	3	5:21.4	
	4	11:14.3	
	5	4:29.1	
	6	4:28.6	
	7	4:13.3	
	8	11:09.9	
	9	3:57.1	
	10	3:53.6	
	11	3:52.0	

FIG. 26



PRESS 'SPACE' TO MOVE BACK TO LAST SET XY POSITION.
PRESS 'ESC' TO DISABLE CONTROLS.
PRESS 'X' FOR XYZ POSITION. MOUSE CONTROLS XY AND WHEEL CONTROLS Z.
PRESS 'R' FOR XYZ ROTATION. MOUSE CONTROLS XY AND WHEEL CONTROLS Z.
PRESS 'F' FOR FOV CONTROLLED BY WHEEL.
PRESS 'A' FOR ASPECT RAITO CONTROLLED BY WHEEL.

FIG. 27



PRESS 'SPACE' TO MOVE BACK TO LAST SET XY POSITION.
PRESS 'ESC' TO DISABLE CONTROLS.
PRESS 'X' FOR XYZ POSITION. MOUSE CONTROLS XY AND WHEEL CONTROLS Z.
PRESS 'R' FOR XYZ ROTATION. MOUSE CONTROLS XY ADN WHEEL CONTROLS Z.
PRESS 'F' FOR FOV CONTROLLED BY WHEEL.
PRESS 'A' FOR ASPECT RATION CONTROLLED BY WHEEL.

POSITION	X	Y	Z
ROTATION	X	Y	Z
ASPECT RATION		FOV	
ANCHOR	○TOPLEFT ○TOPRIGHT ○BOTTOMLEFT ○BOTTOMRIGHT		

FIG. 28

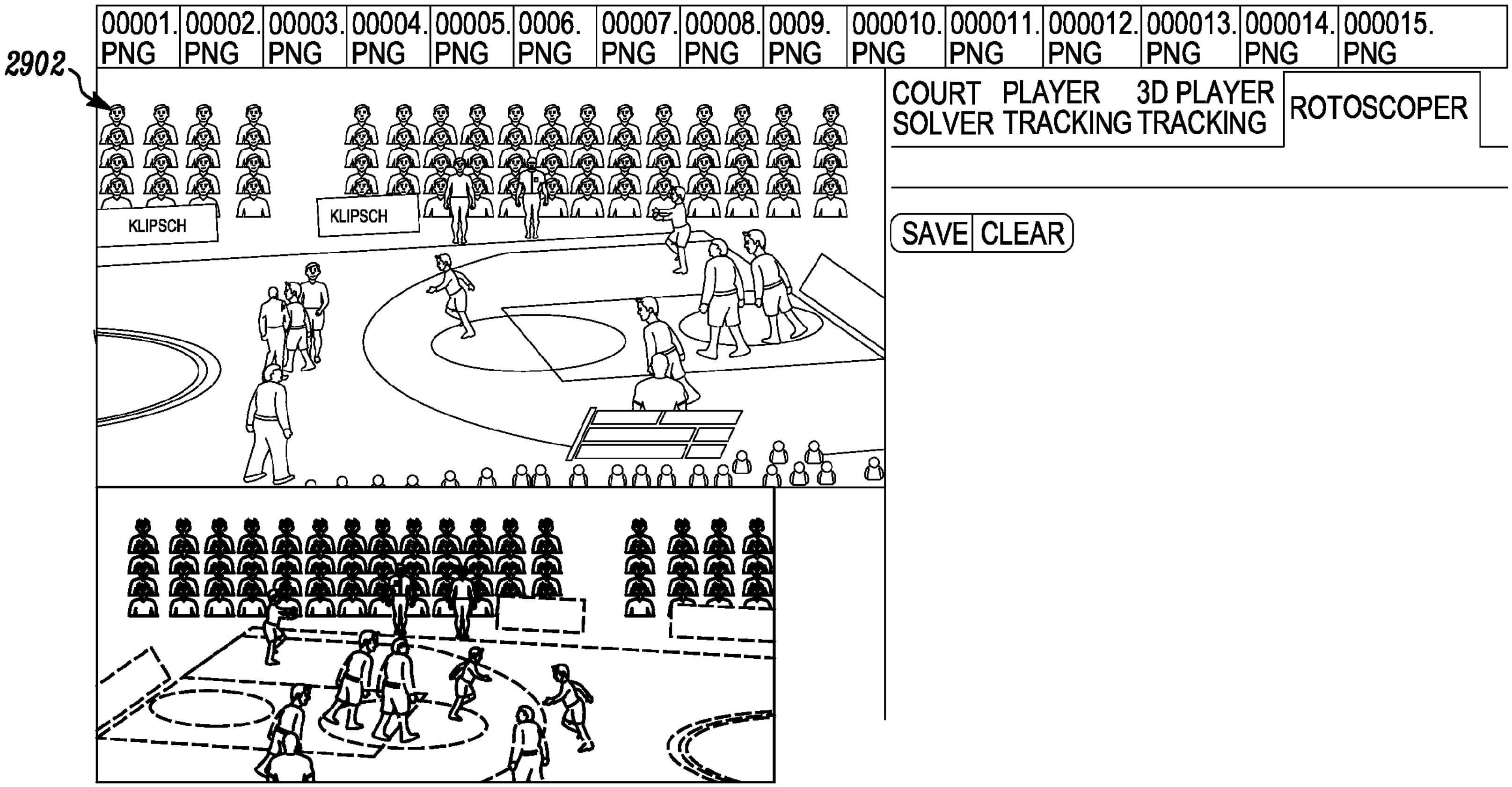
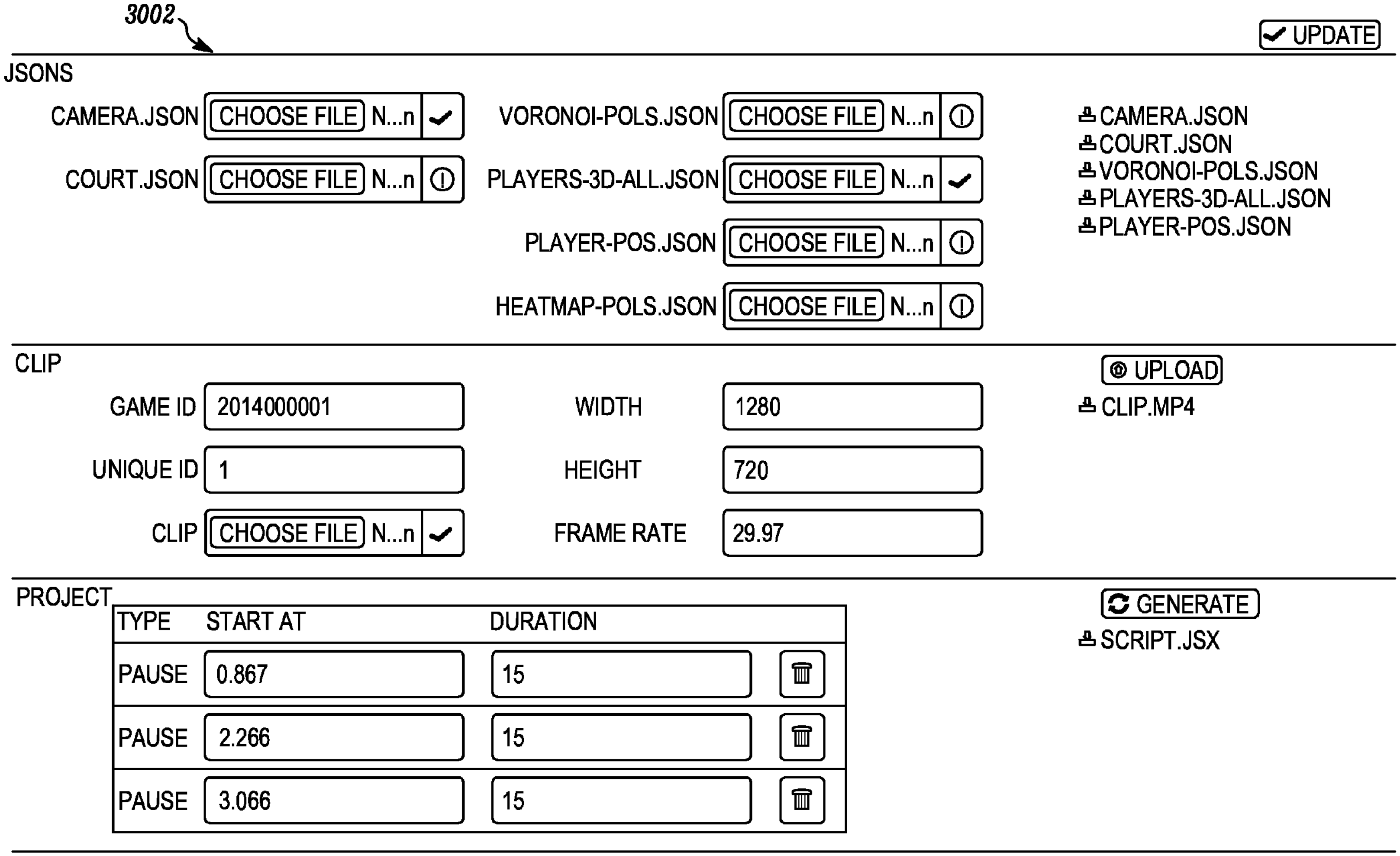

00001. PNG
00002. PNG
00003. PNG
00004. PNG
00005. PNG
0006. PNG
00007. PNG
00008. PNG
0009. PNG
000010. PNG
000011. PNG
000012. PNG
000013. PNG
000014. PNG
000015. PNG
2902
KLIPSCH
KLIPSCH
COURT SOLVER
PLAYER TRACKING
3D PLAYER TRACKING
ROTOSCOPER
SAVE
CLEAR

FIG. 29



JSONS

CAMERA.JSON [CHOOSE FILE] N...n ✓

COURT.JSON [CHOOSE FILE] N...n ⓘ

VORONOI-POLS.JSON [CHOOSE FILE] N...n ⓘ

PLAYERS-3D-ALL.JSON [CHOOSE FILE] N...n ✓

PLAYER-POS.JSON [CHOOSE FILE] N...n ⓘ

HEATMAP-POLS.JSON [CHOOSE FILE] N...n ⓘ

- CAMERA.JSON
- COURT.JSON
- VORONOI-POLS.JSON
- PLAYERS-3D-ALL.JSON
- PLAYER-POS.JSON

CLIP

GAME ID 2014000001

UNIQUE ID 1

CLIP [CHOOSE FILE] N...n ✓

WIDTH 1280

HEIGHT 720

FRAME RATE 29.97

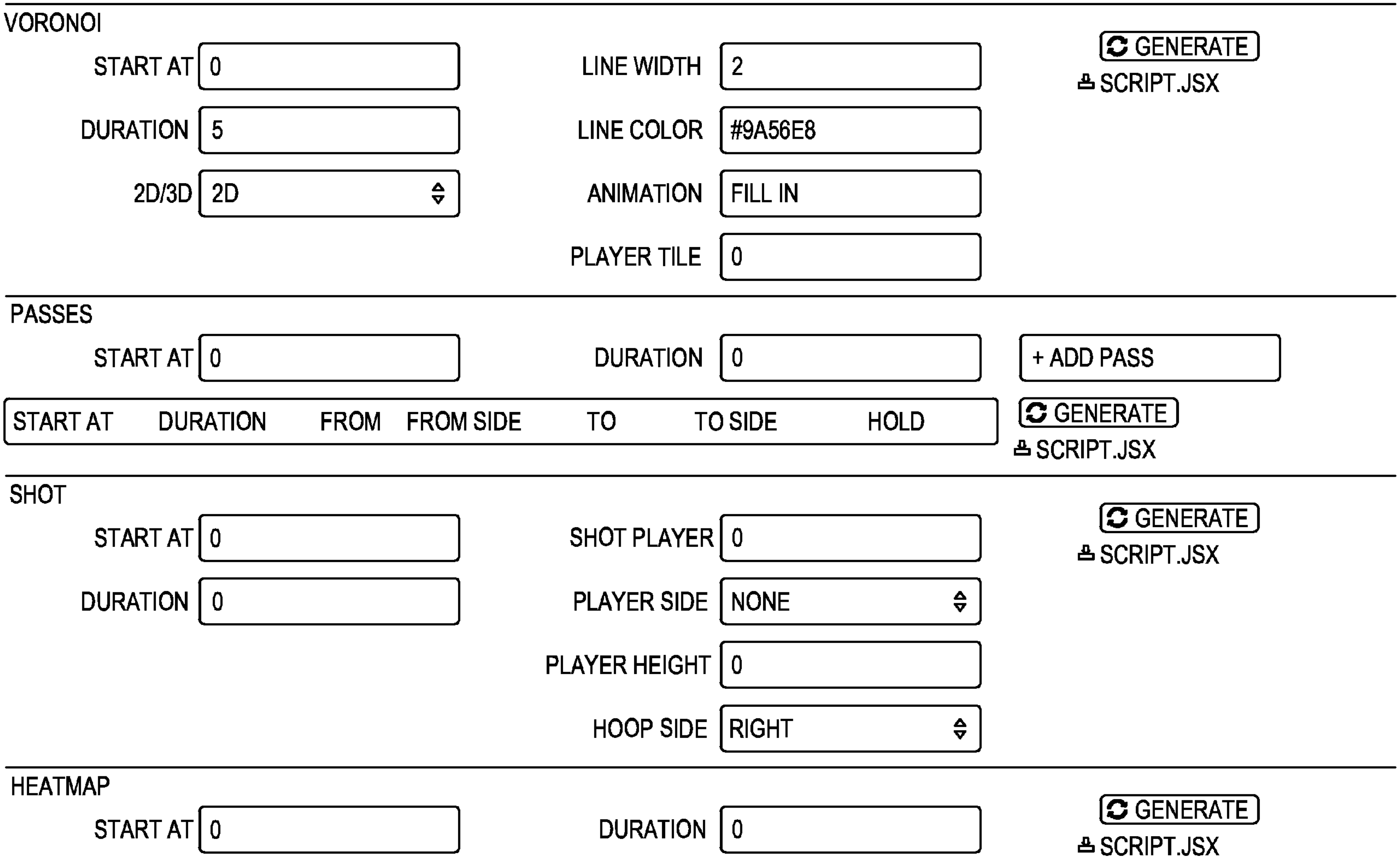
- CLIP.MP4

PROJECT

TYPE	START AT	DURATION	
PAUSE	0.867	15	🗑
PAUSE	2.266	15	🗑
PAUSE	3.066	15	🗑

- SCRIPT.JSX

FIG. 30A



VORONOI

START AT 0

DURATION 5

2D/3D 2D

LINE WIDTH 2

LINE COLOR #9A56E8

ANIMATION FILL IN

PLAYER TILE 0

PASSES

START AT 0

DURATION 0

START AT	DURATION	FROM	FROM SIDE	TO	TO SIDE	HOLD

SHOT

START AT 0

DURATION 0

SHOT PLAYER 0

PLAYER SIDE NONE

PLAYER HEIGHT 0

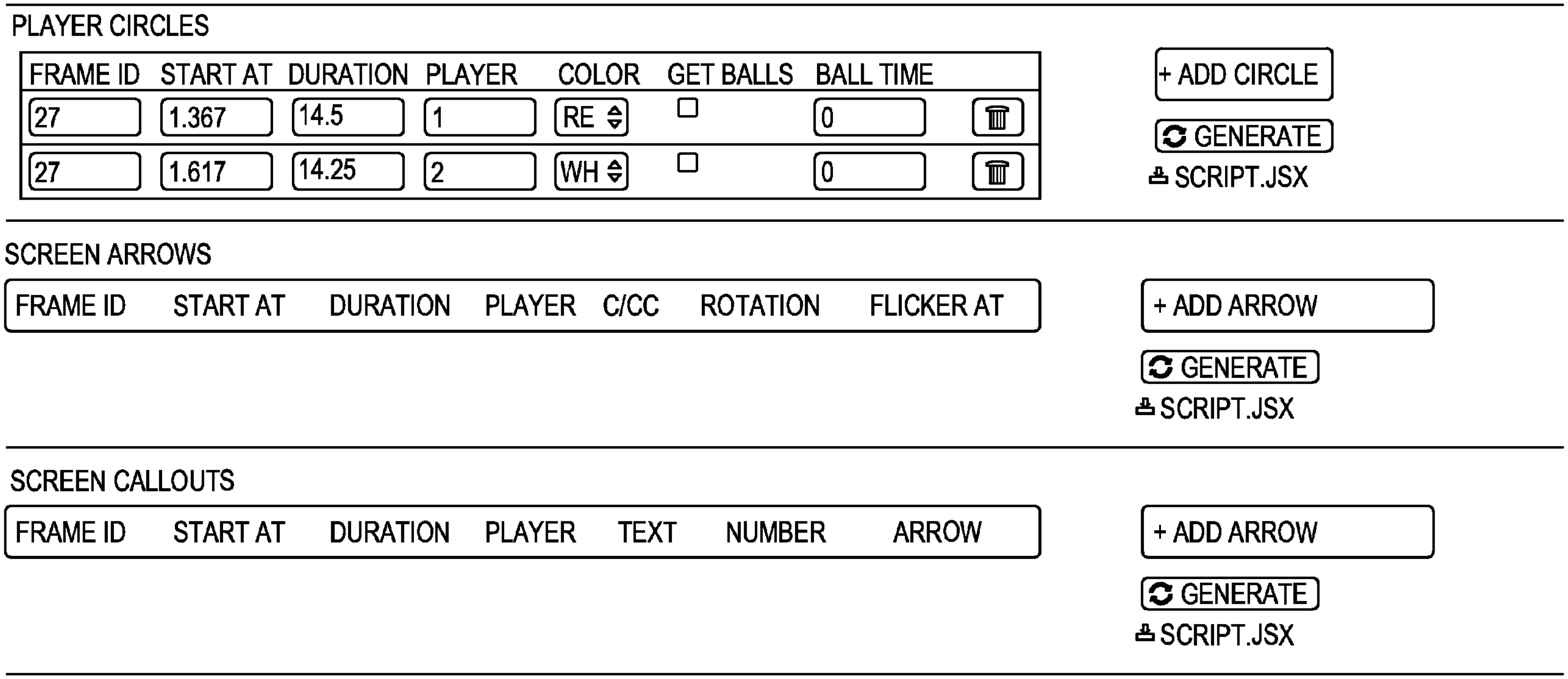
HOOP SIDE RIGHT

HEATMAP

START AT 0

DURATION 0

FIG. 30B



PLAYER CIRCLES

FRAME ID	START AT	DURATION	PLAYER	COLOR	GET BALLS	BALL TIME	
27	1.367	14.5	1	RE ⇕	□	0	🗑
27	1.617	14.25	2	WH ⇕	□	0	🗑

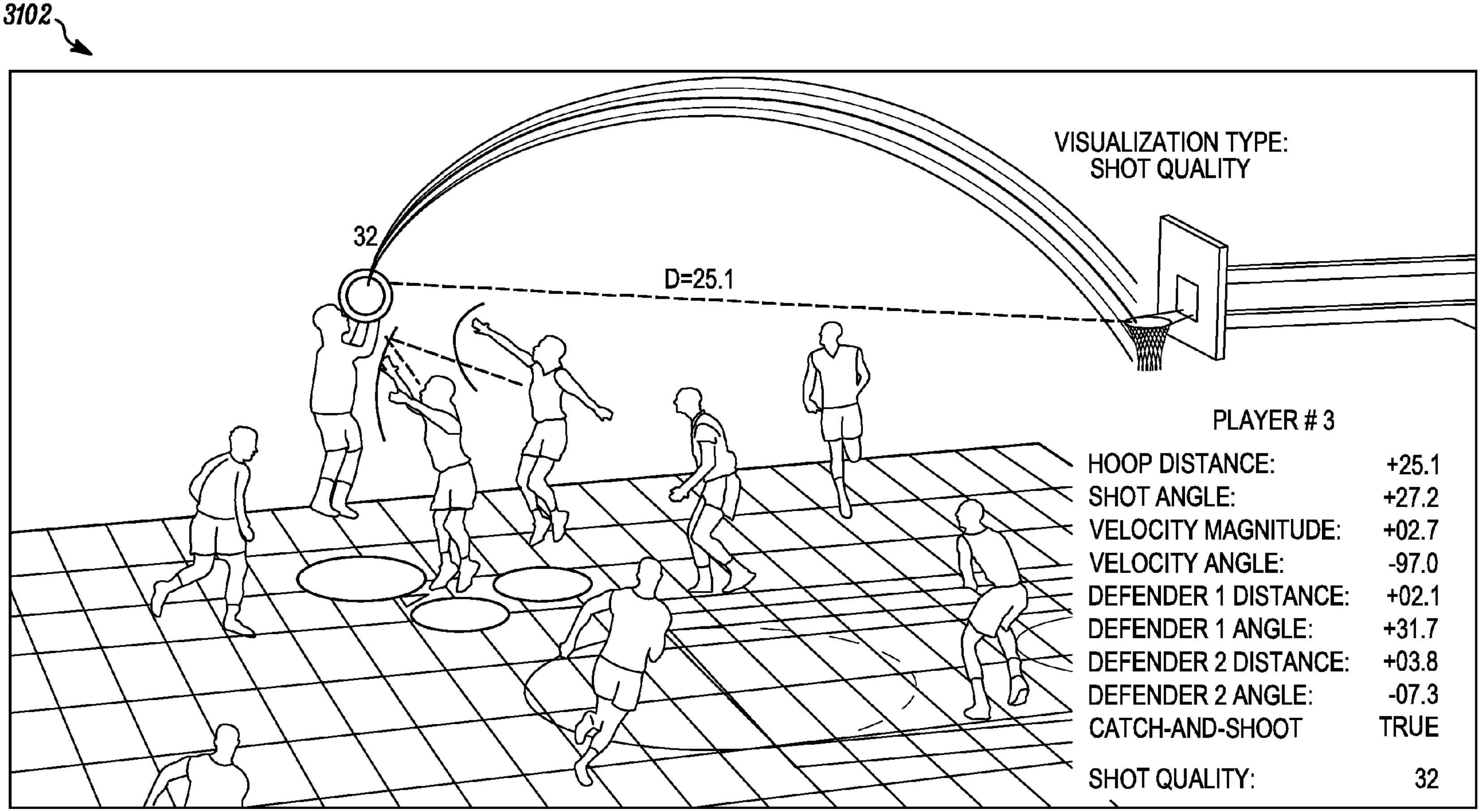
SCREEN ARROWS

FRAME ID	START AT	DURATION	PLAYER	C/CC	ROTATION	FLICKER AT

SCREEN CALLOUTS

FRAME ID	START AT	DURATION	PLAYER	TEXT	NUMBER	ARROW

FIG. 30C



VISUALIZATION TYPE:
SHOT QUALITY

PLAYER # 3

HOOP DISTANCE:	+25.1
SHOT ANGLE:	+27.2
VELOCITY MAGNITUDE:	+02.7
VELOCITY ANGLE:	-97.0
DEFENDER 1 DISTANCE:	+02.1
DEFENDER 1 ANGLE:	+31.7
DEFENDER 2 DISTANCE:	+03.8
DEFENDER 2 ANGLE:	-07.3
CATCH-AND-SHOOT	TRUE
SHOT QUALITY:	32

FIG. 31

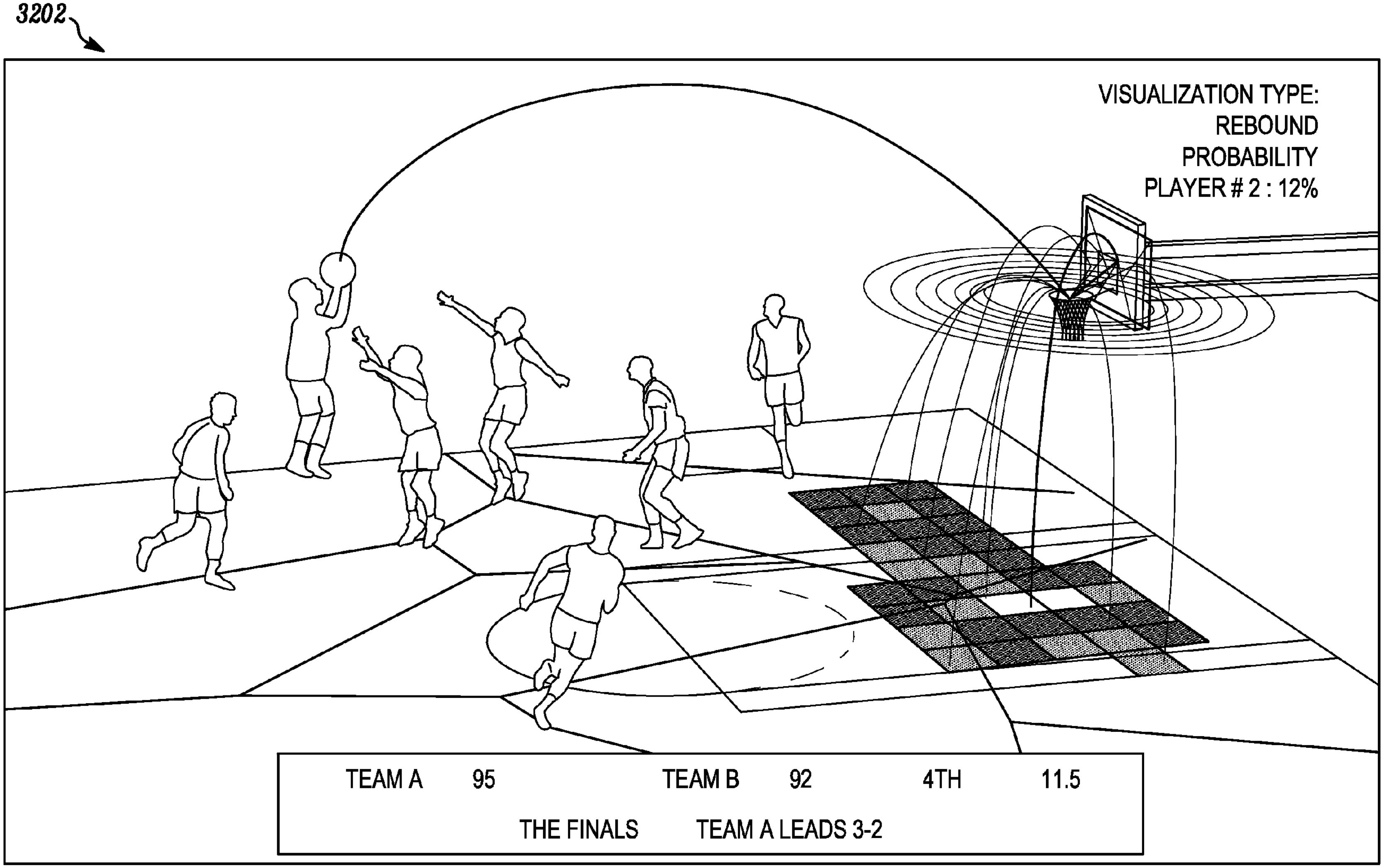

3202
VISUALIZATION TYPE:
REBOUND
PROBABILITY
PLAYER # 2 : 12%
TEAM A 95
TEAM B 92
4TH 11.5
THE FINALS
TEAM A LEADS 3-2

FIG. 32

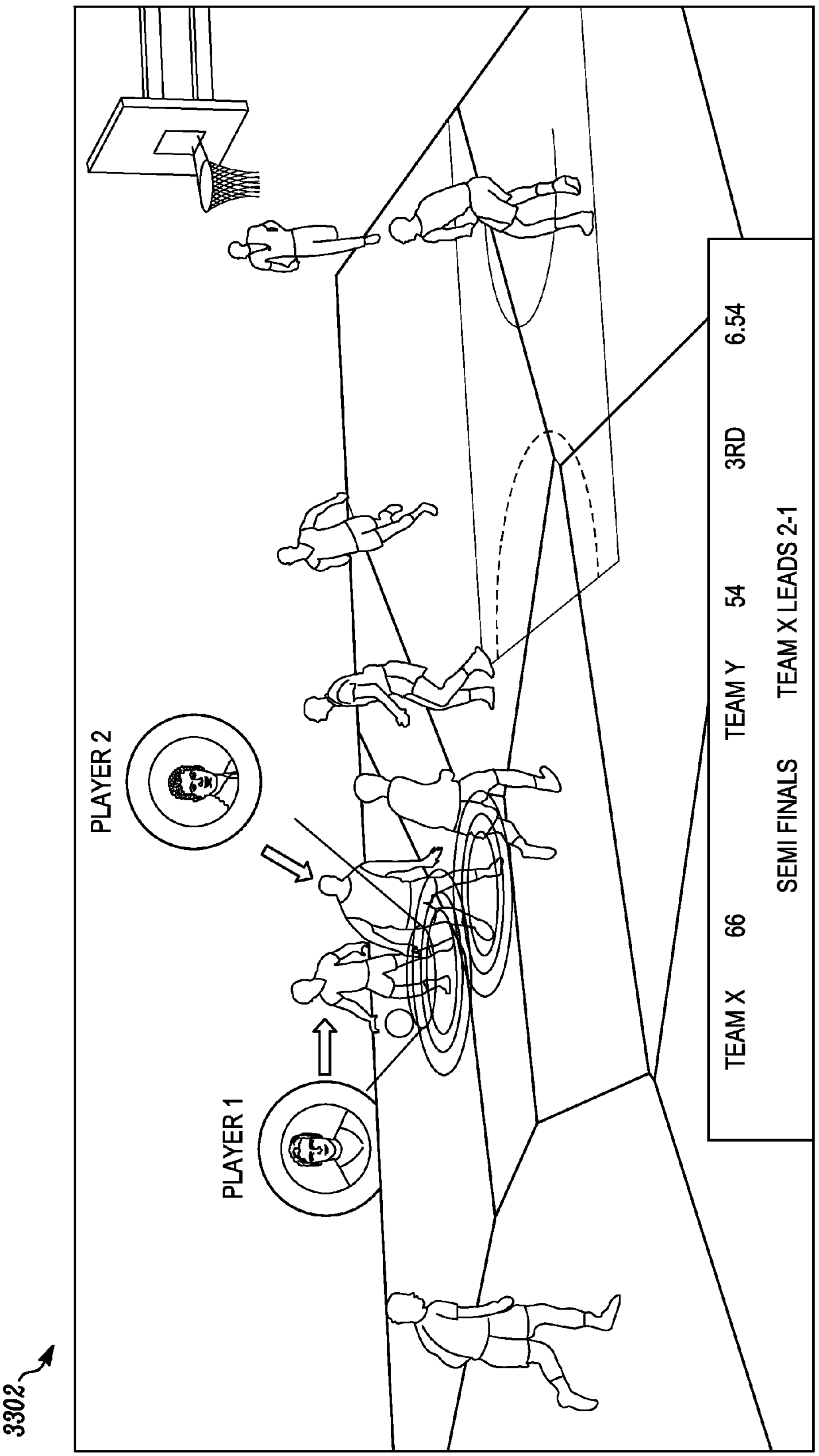

3302
PLAYER 1
PLAYER 2
TEAM X 66
TEAM Y 54
3RD
6.54
SEMI FINALS
TEAM X LEADS 2-1

FIG. 33

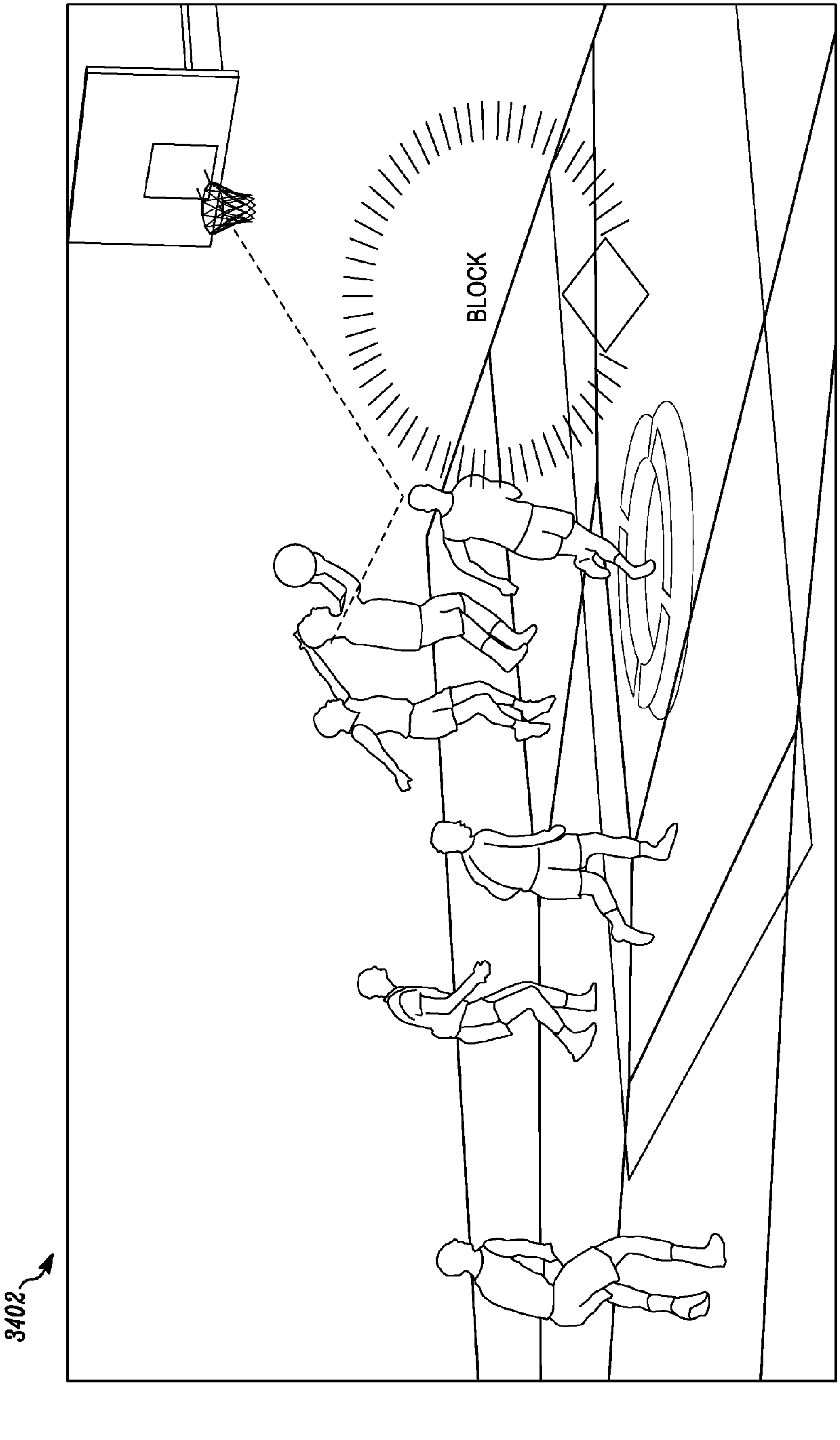


FIG. 34

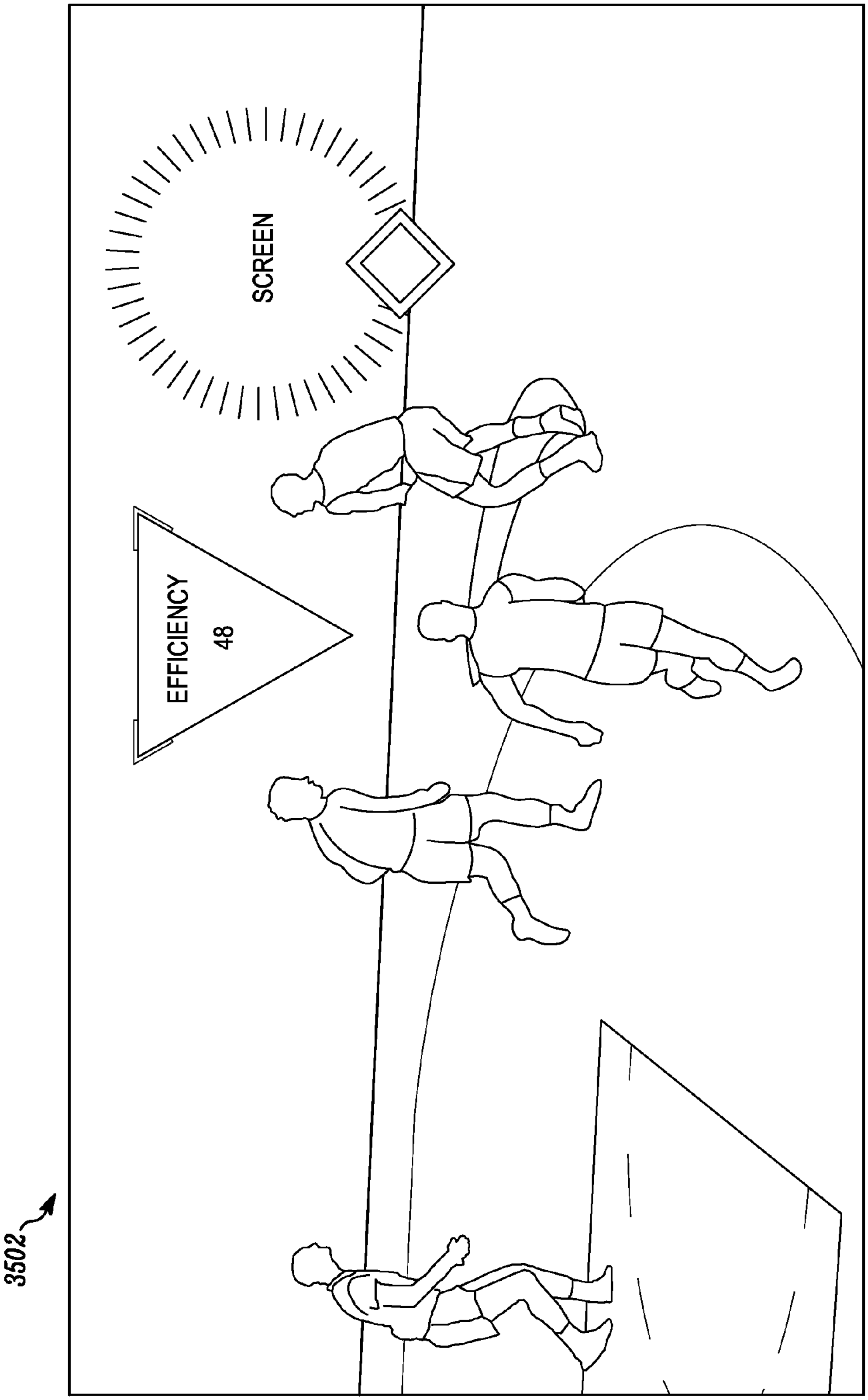


FIG. 35

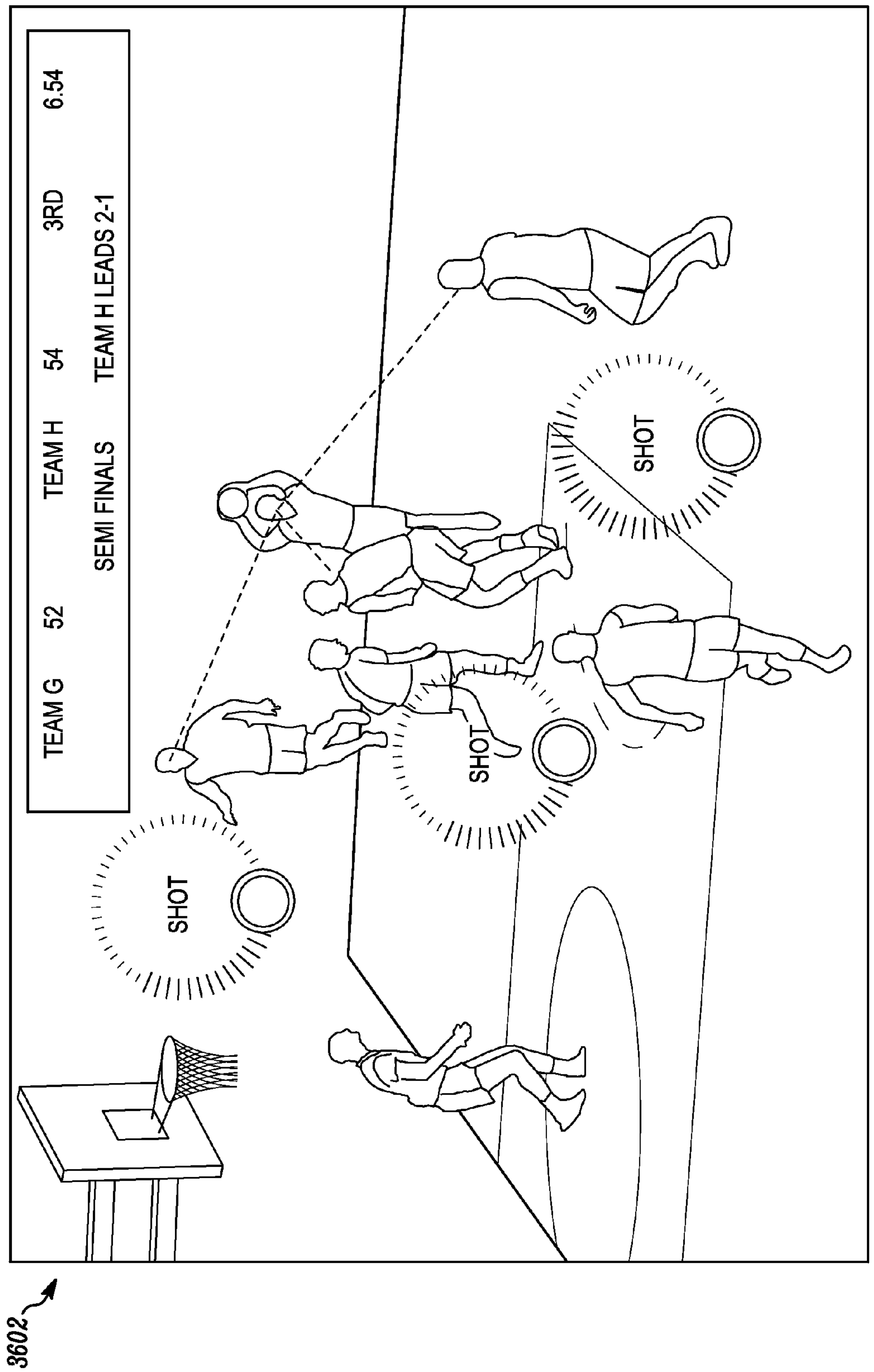

TEAM G
52
TEAM H
54
3RD
6.54
SEMI FINALS
TEAM H LEADS 2-1
SHOT
SHOT
SHOT
3602

FIG. 36

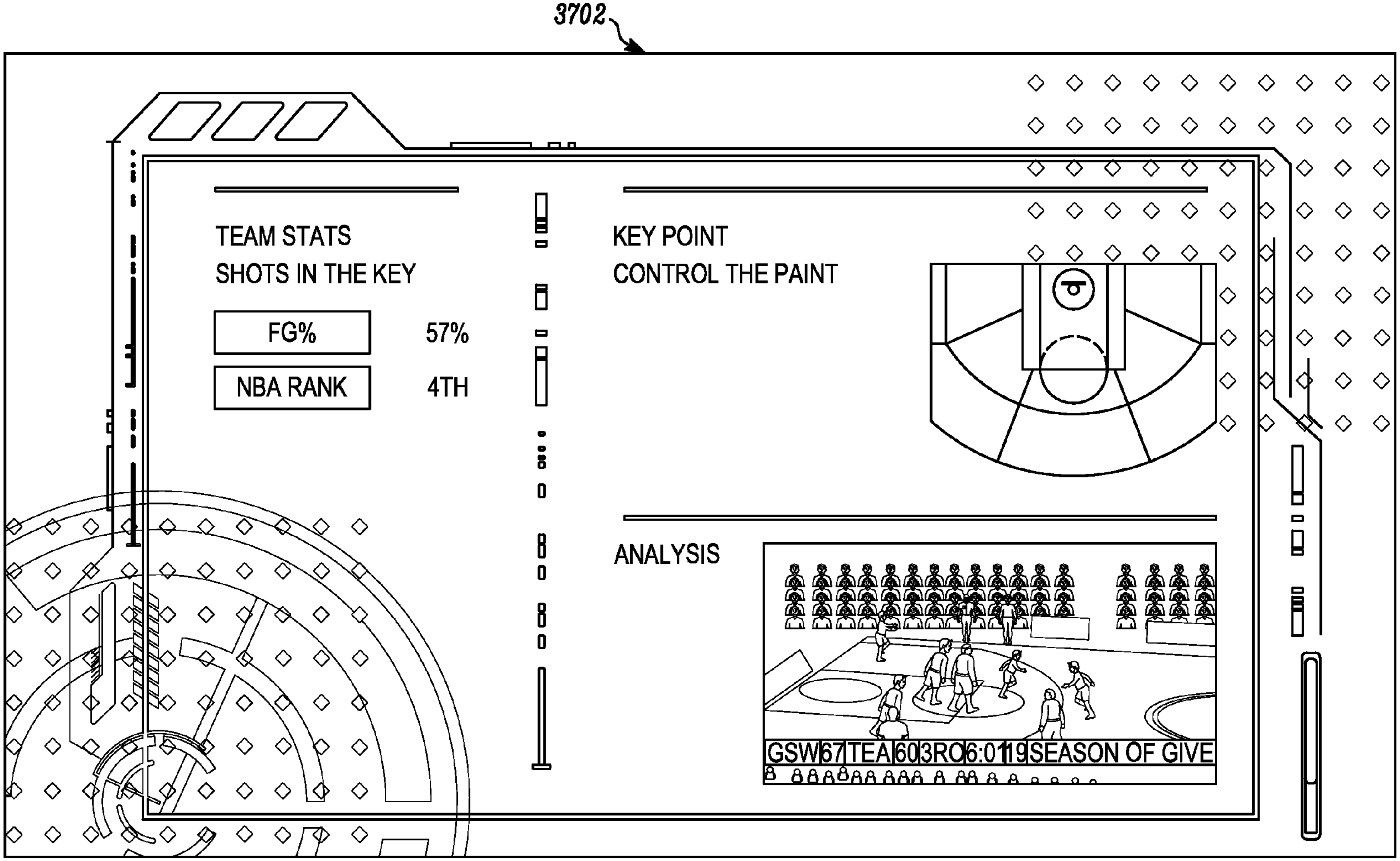

3702
TEAM STATS
SHOTS IN THE KEY
FG% 57%
NBA RANK 4TH
KEY POINT
CONTROL THE PAINT
ANALYSIS
GSW 67 TEA 60 3RD 6:01 19 SEASON OF GIVE

FIG. 37

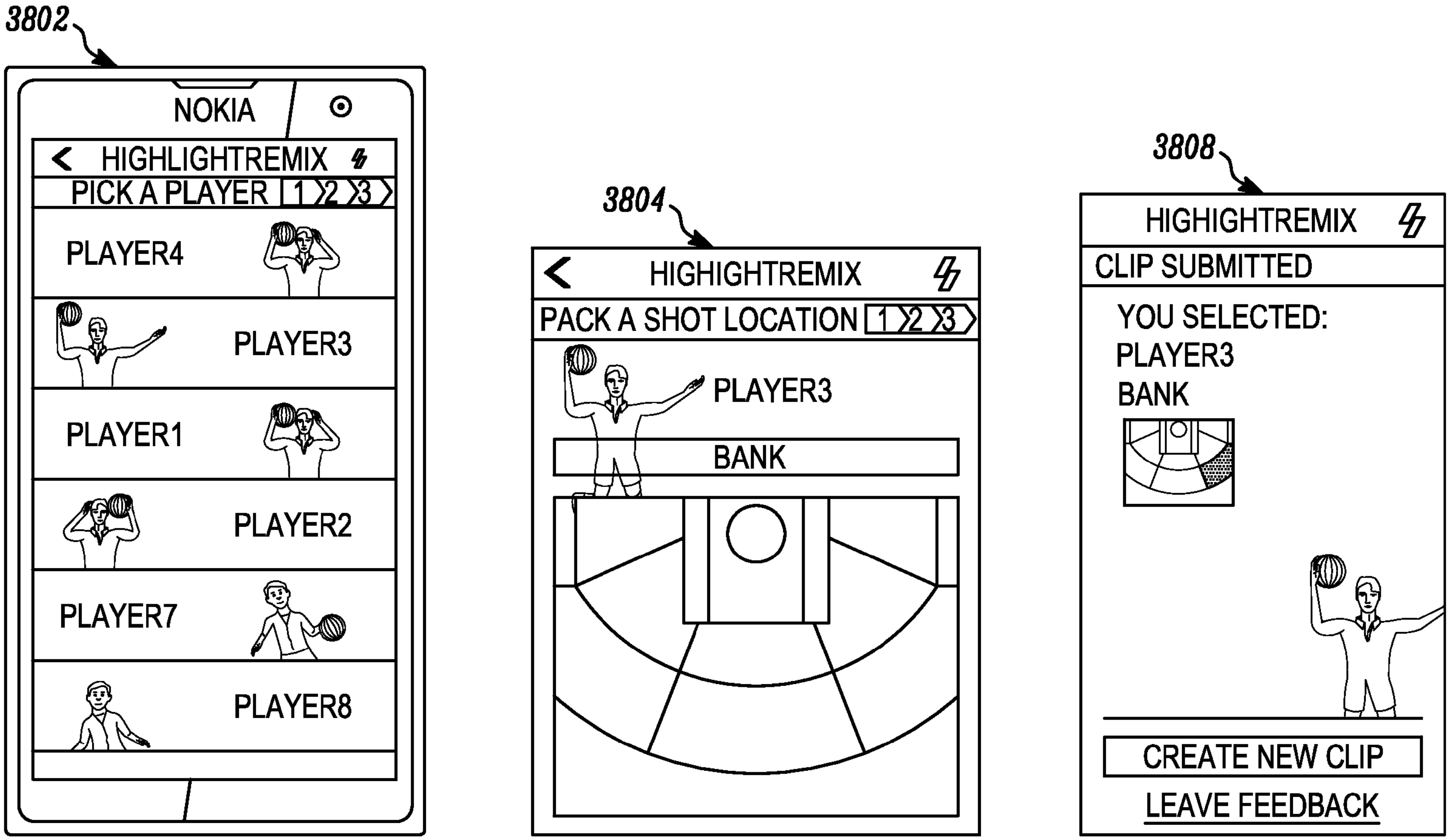
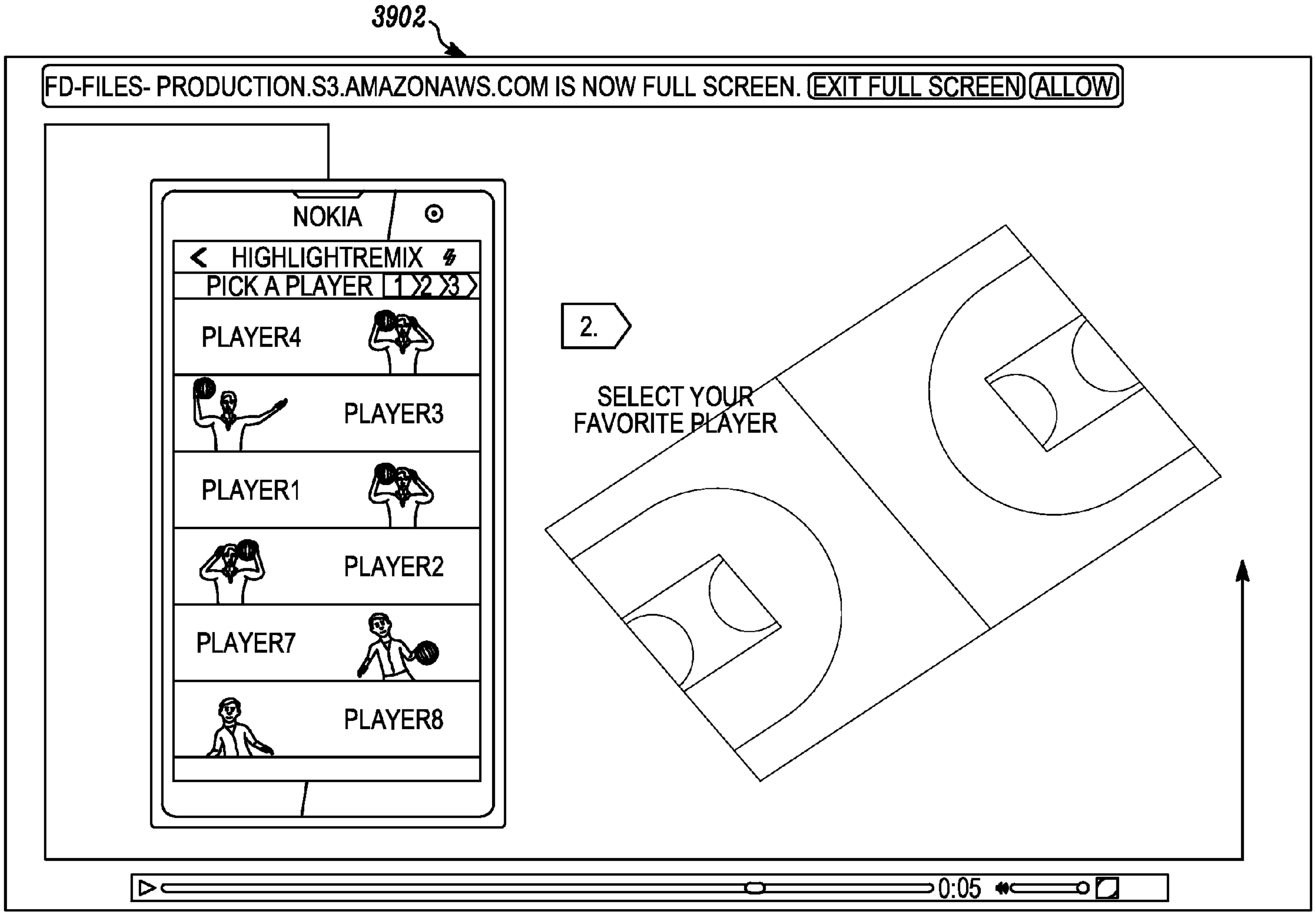

3802
NOKIA
HIGHLIGHTREMIX
PICK A PLAYER 1 2 3
PLAYER4
PLAYER3
PLAYER1
PLAYER2
PLAYER7
PLAYER8
3804
HIGHIGHTREMIX
PACK A SHOT LOCATION 1 2 3
PLAYER3
BANK
3808
HIGHIGHTREMIX
CLIP SUBMITTED
YOU SELECTED:
PLAYER3
BANK
CREATE NEW CLIP
LEAVE FEEDBACK

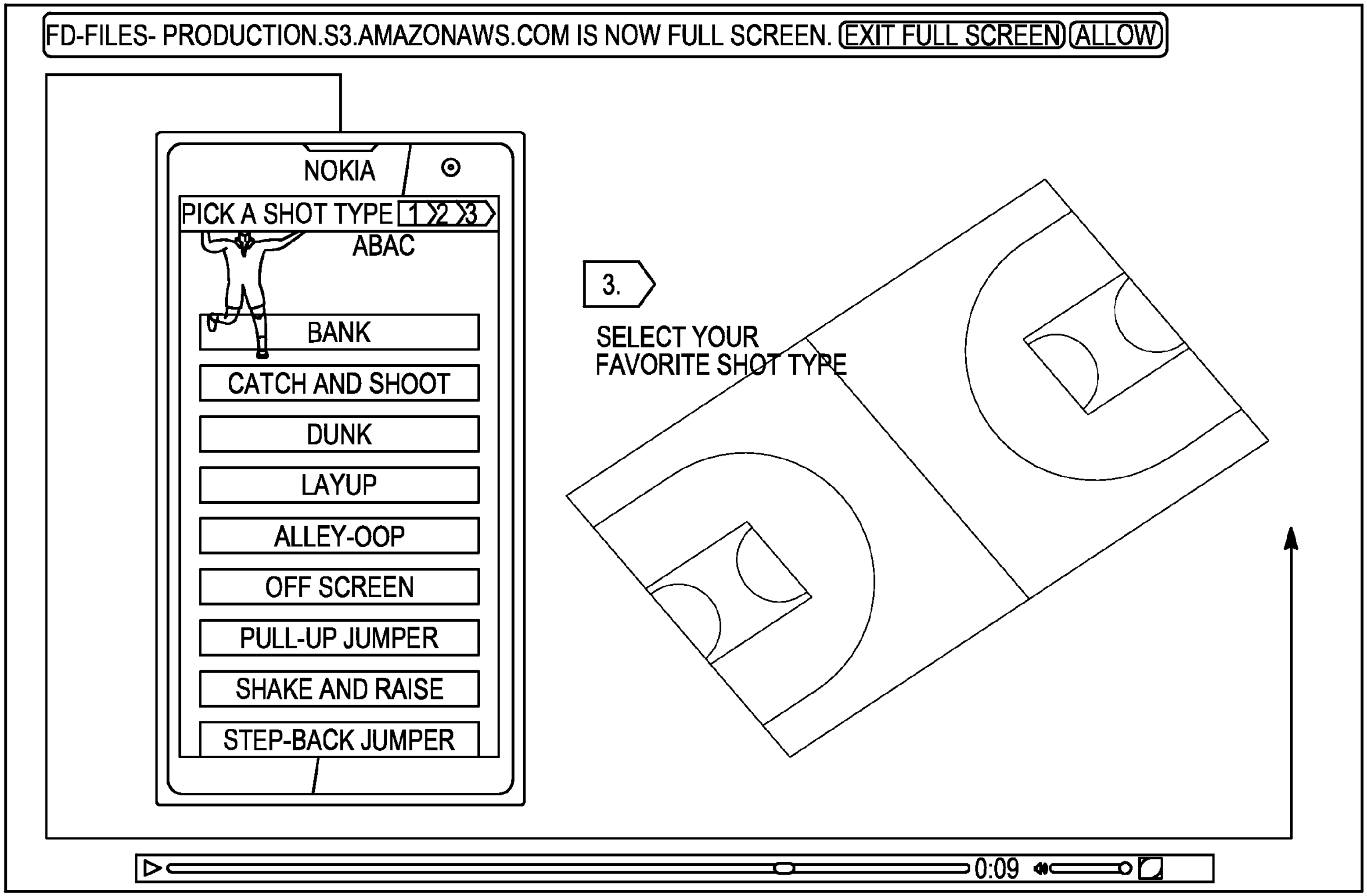
FIG. 38



FD-FILES- PRODUCTION.S3.AMAZONAWS.COM IS NOW FULL SCREEN.

SELECT YOUR FAVORITE PLAYER

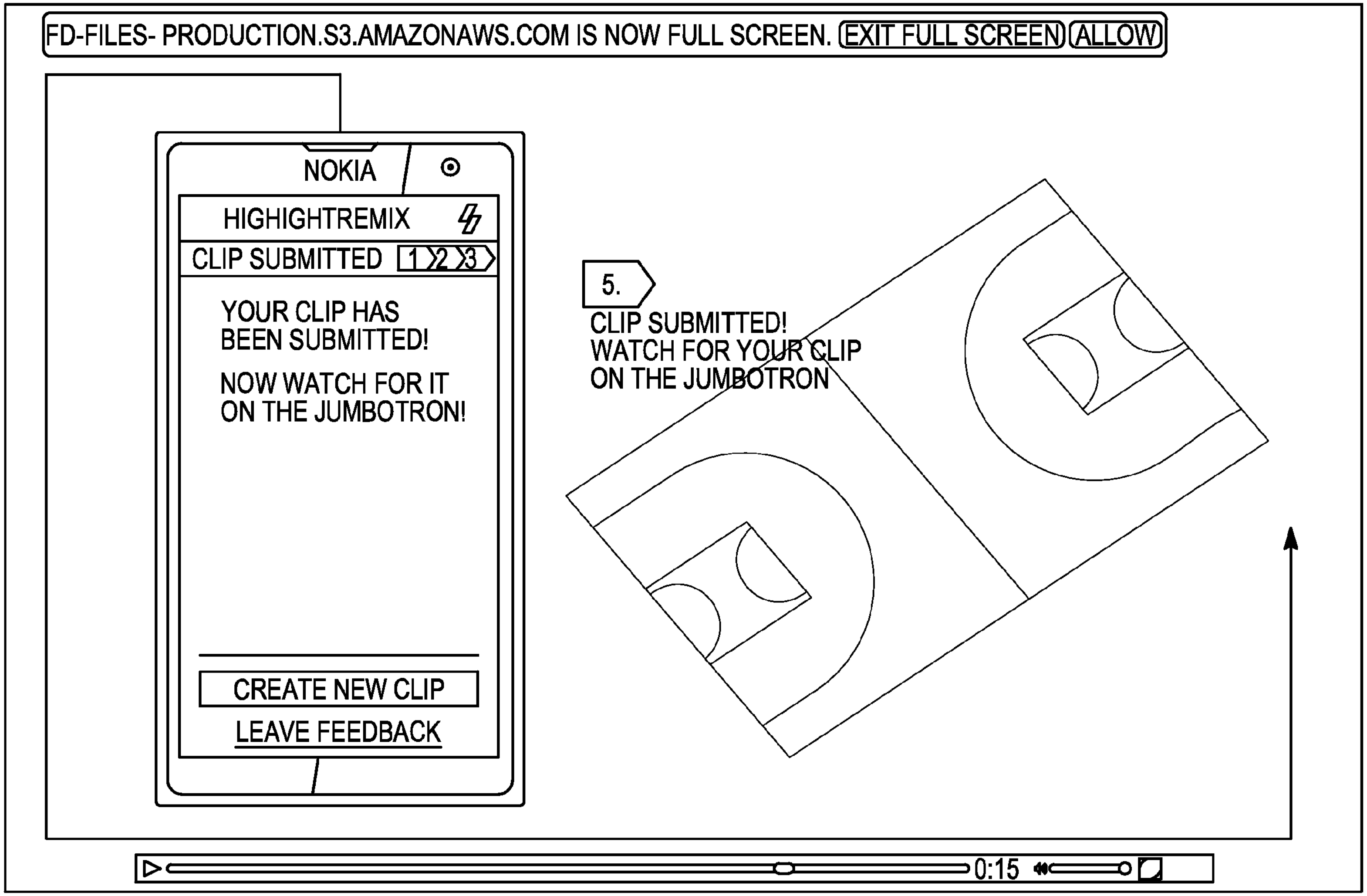
FIG. 39A



FD-FILES- PRODUCTION.S3.AMAZONAWS.COM IS NOW FULL SCREEN.

SELECT YOUR FAVORITE SHOT TYPE

FIG. 39B



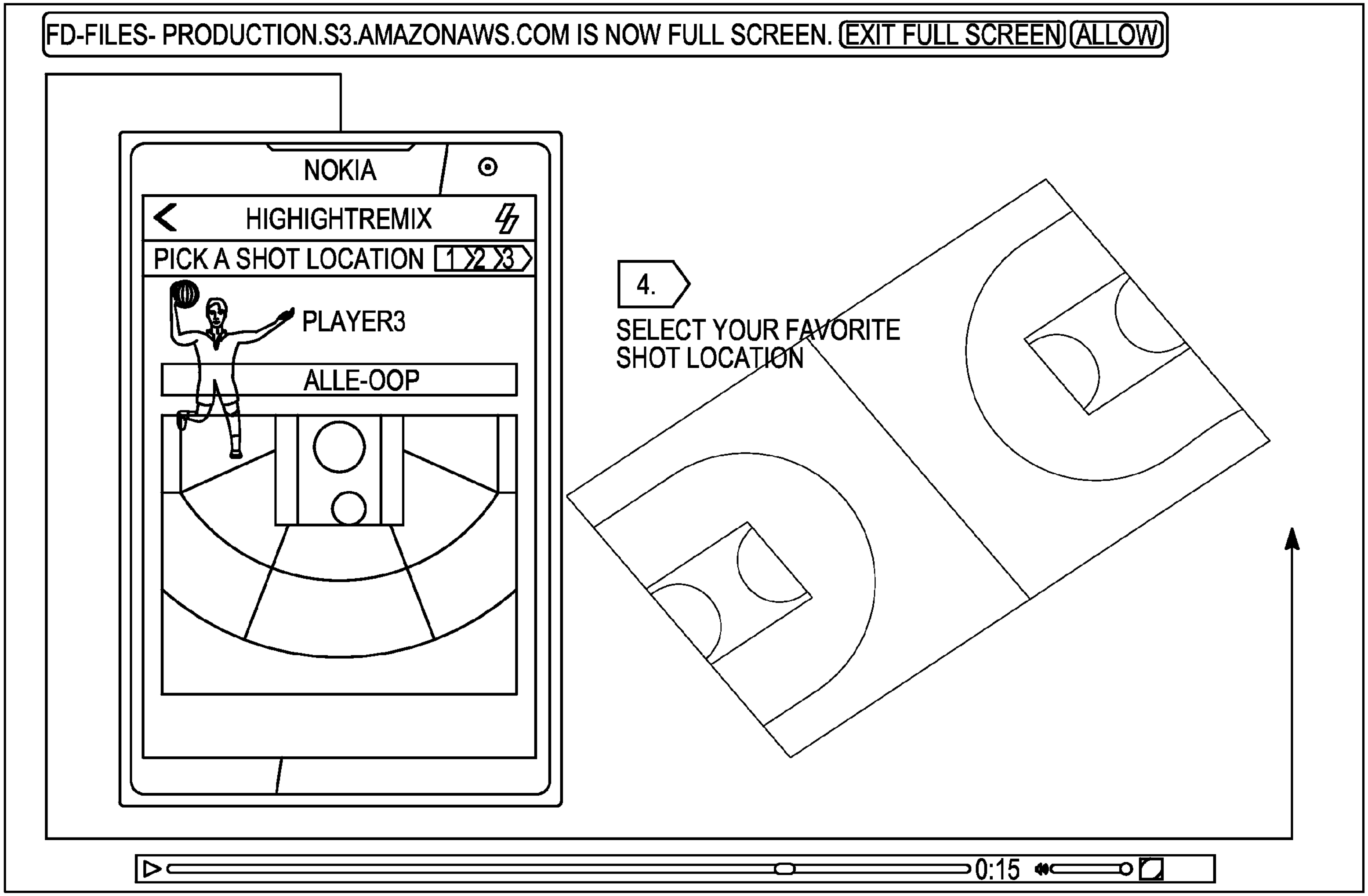
FD-FILES- PRODUCTION.S3.AMAZONAWS.COM IS NOW FULL SCREEN.

YOUR CLIP HAS BEEN SUBMITTED!

NOW WATCH FOR IT ON THE JUMBOTRON!

CLIP SUBMITTED! WATCH FOR YOUR CLIP ON THE JUMBOTRON

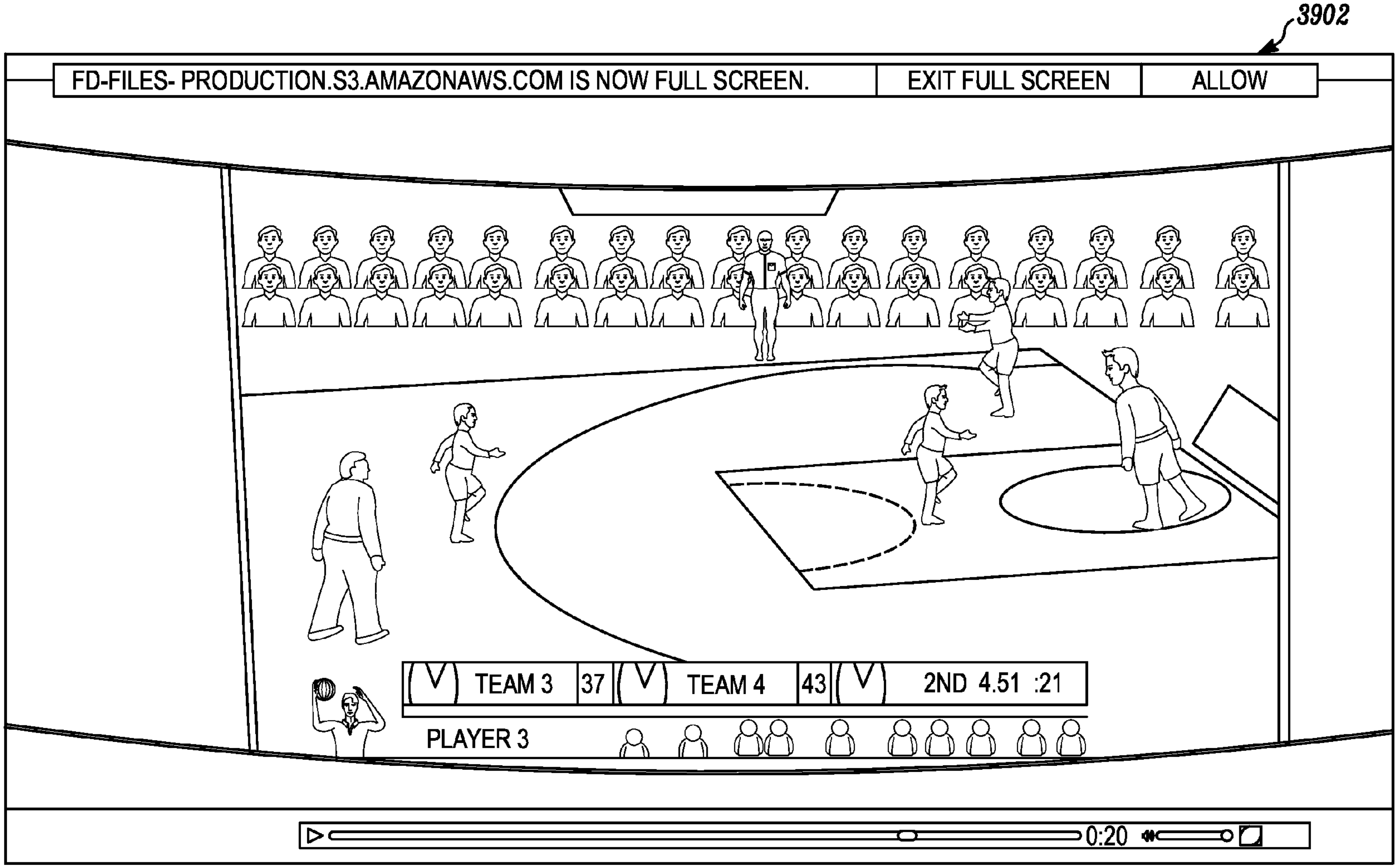
FIG. 39C



FD-FILES- PRODUCTION.S3.AMAZONAWS.COM IS NOW FULL SCREEN.

SELECT YOUR FAVORITE SHOT LOCATION

FIG. 39D



FD-FILES- PRODUCTION.S3.AMAZONAWS.COM IS NOW FULL SCREEN.

FIG. 39E

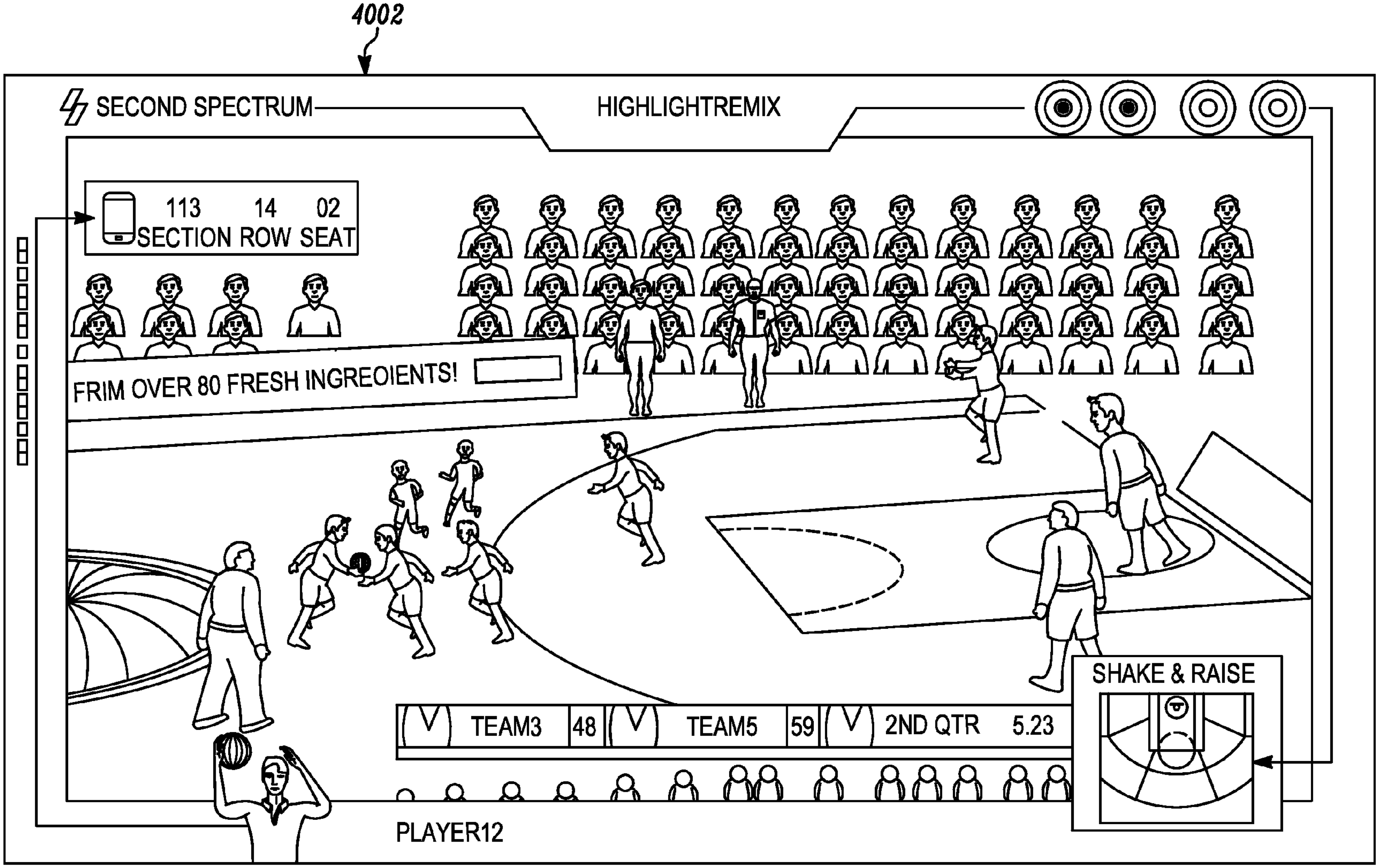
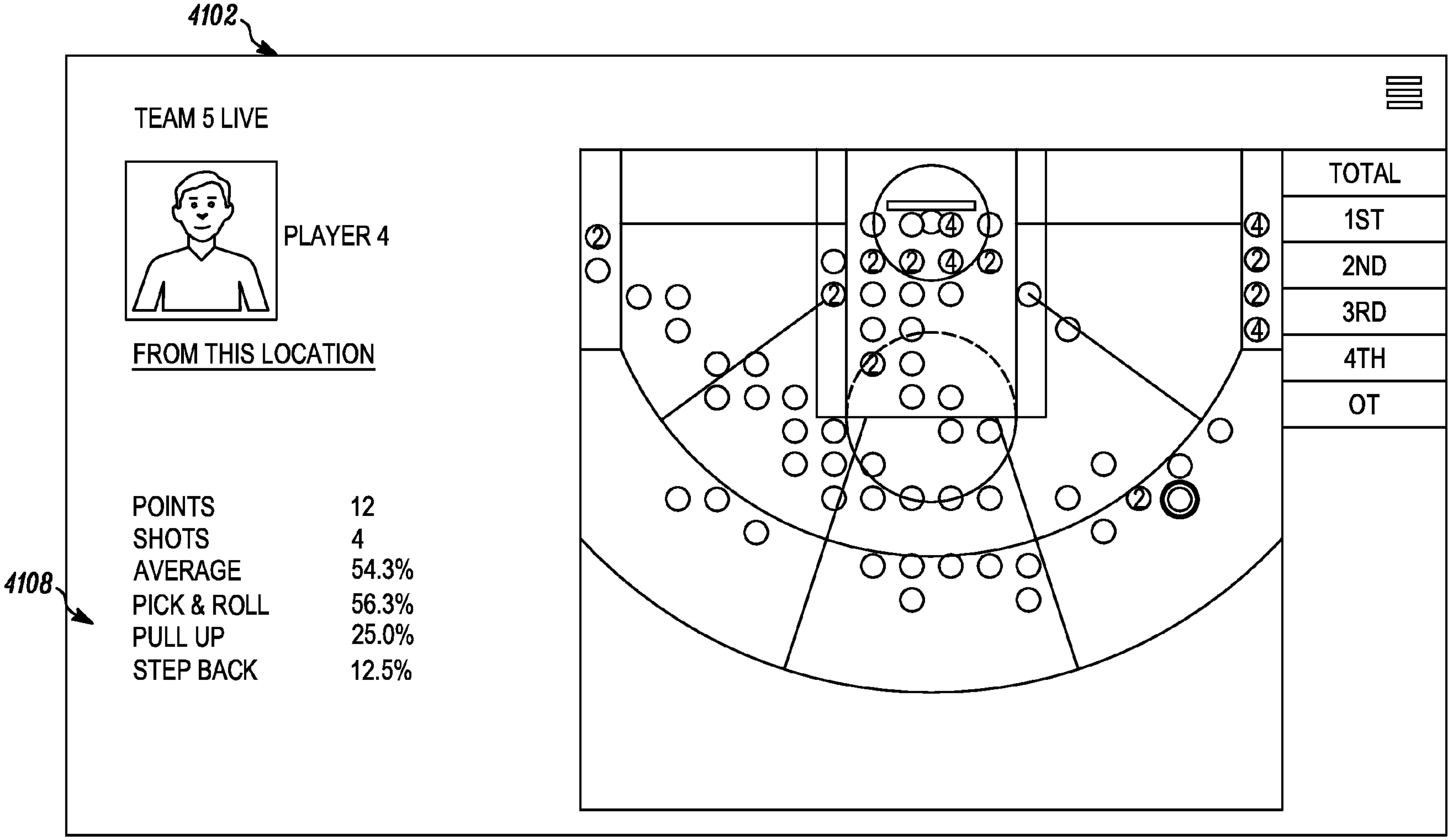

4002
SECOND SPECTRUM
HIGHLIGHTREMIX
113 14 02
SECTION ROW SEAT
FRIM OVER 80 FRESH INGREOIENTS!
SHAKE & RAISE
TEAM3
48
TEAM5
59
2ND QTR
5.23
PLAYER12

FIG. 40



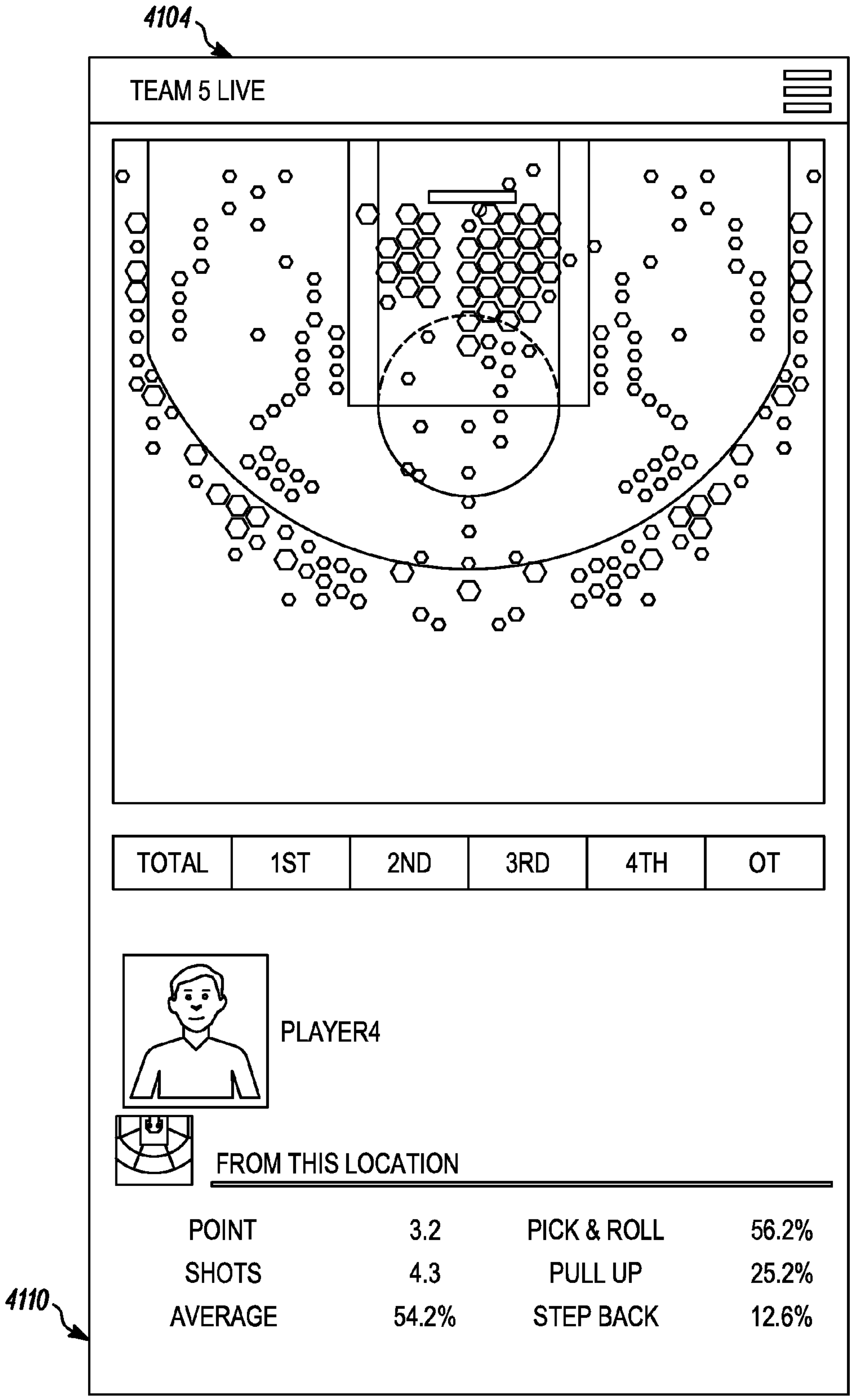
TEAM 5 LIVE

PLAYER 4

FROM THIS LOCATION

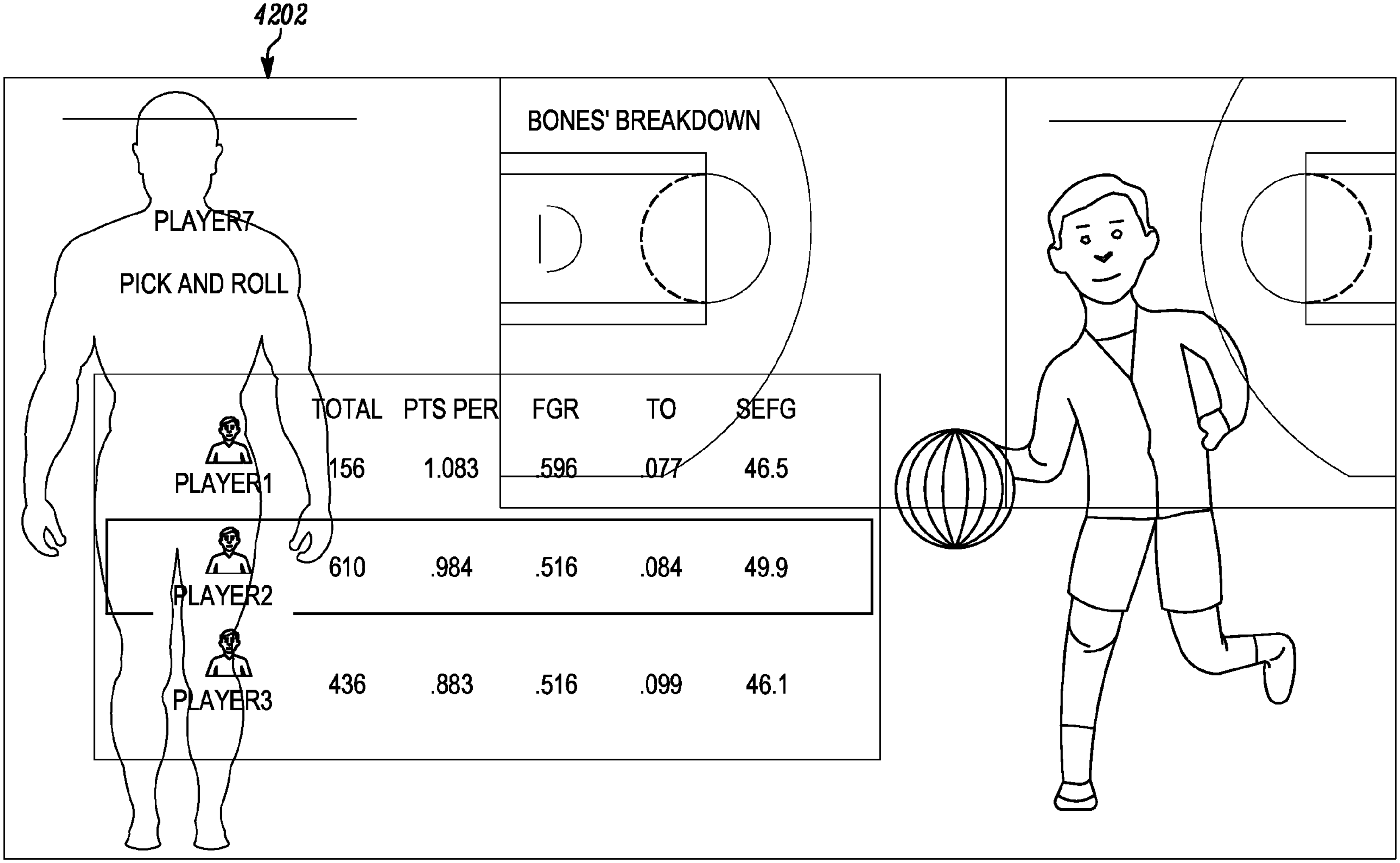
POINTS	12
SHOTS	4
AVERAGE	54.3%
PICK & ROLL	56.3%
PULL UP	25.0%
STEP BACK	12.5%

FIG. 41A



POINT	3.2	PICK & ROLL	56.2%
SHOTS	4.3	PULL UP	25.2%
AVERAGE	54.2%	STEP BACK	12.6%

FIG. 41B



	TOTAL	PTS PER	FGR	TO	SEFG
PLAYER1	156	1.083	.596	.077	46.5
PLAYER2	610	.984	.516	.084	49.9
PLAYER3	436	.883	.516	.099	46.1

FIG. 42A

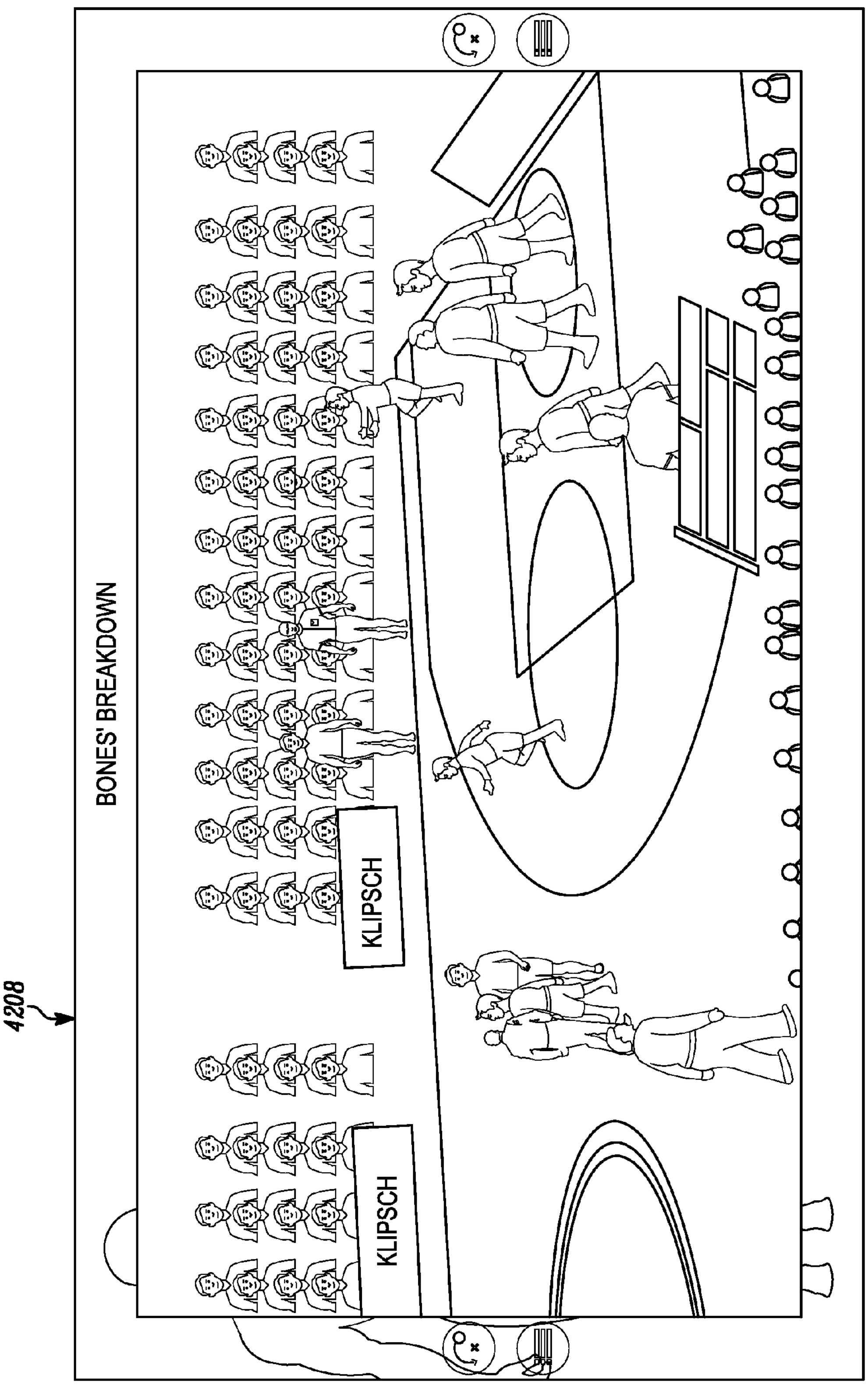


FIG. 42B

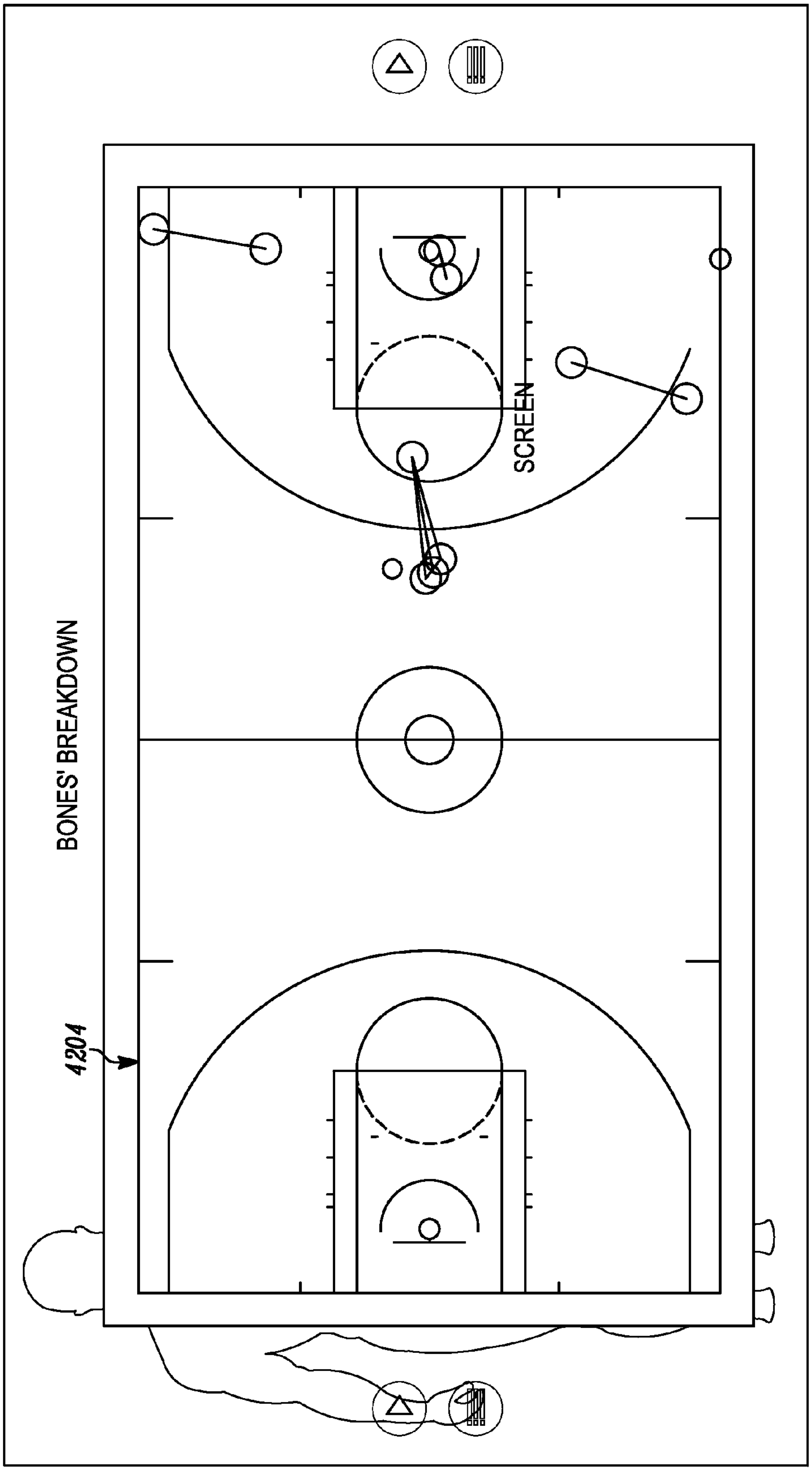
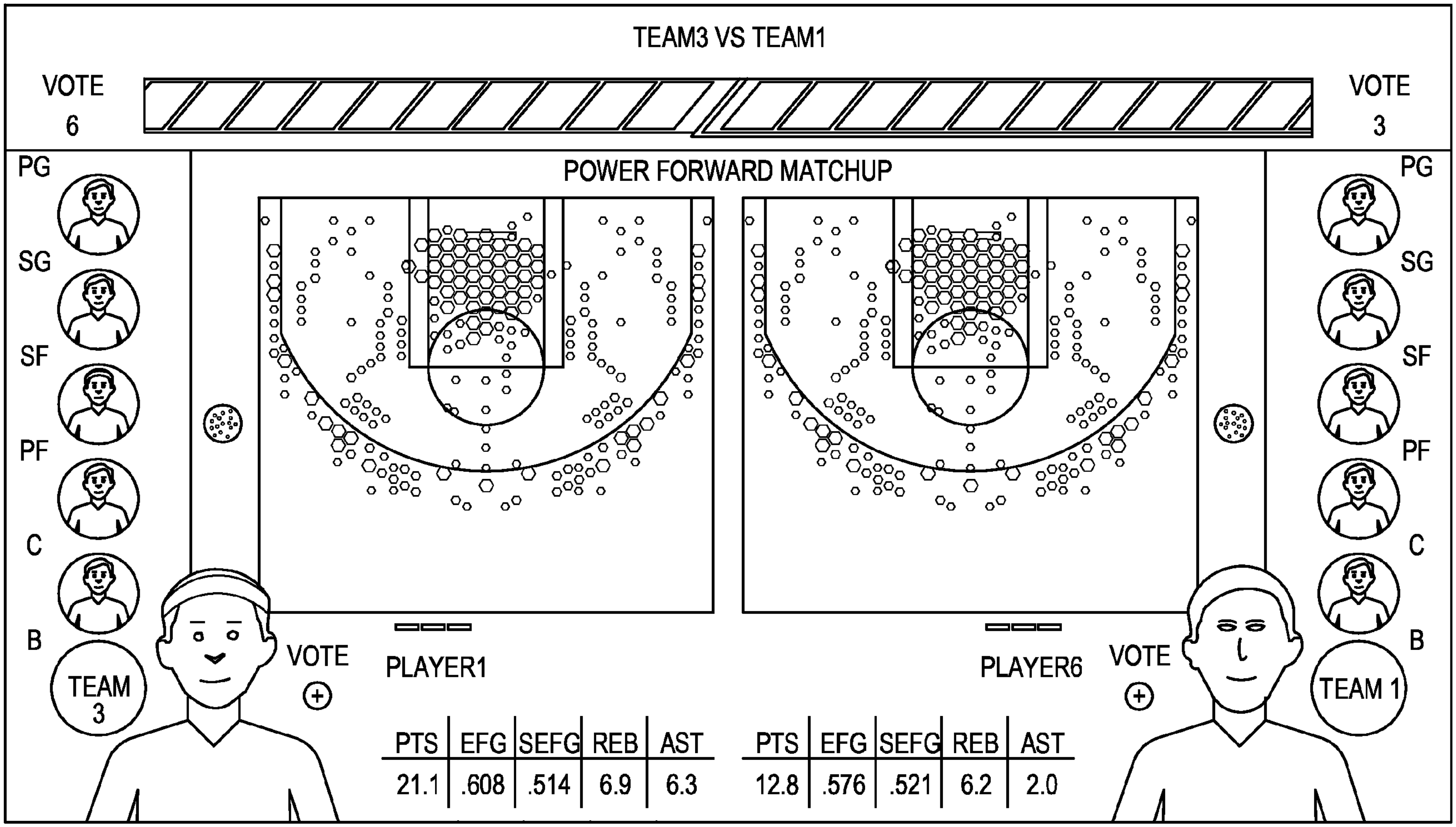


FIG. 42C



PTS	EFG	SEFG	REB	AST
21.1	.608	.514	6.9	6.3

PTS	EFG	SEFG	REB	AST
12.8	.576	.521	6.2	2.0

FIG. 43

SYSTEM AND METHOD FOR PERFORMING SPATIO-TEMPORAL ANALYSIS OF SPORTING EVENTS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to the following provisional U.S. patent applications, which are hereby incorporated by reference in their entirety: provisional U.S. patent application 62/072,308 filed Oct. 29, 2014 and provisional U.S. patent application 61/945,899 filed Feb. 28, 2014.

BACKGROUND

[0002] 1. Field of the Invention

[0003] The present application generally relates to a system and method for performing analysis of events that appear in live and recorded video feeds, such as sporting events. In particular, the present application relates to a system and methods for enabling spatio-temporal analysis of component attributes and elements that make up events within a video feed, such as of a sporting event, systems for discovering, learning, extracting and analyzing such events, metrics and analytic results relating to such events, and methods and systems for display, visualization and interaction with outputs from such methods and systems.

[0004] 2. Description of the Related Art

[0005] Live events, such as sports, especially at the college and professional levels, continue to grow in popularity and revenue as individual colleges and franchises reap billions in revenue each year. To provide valuable insights and gain a competitive advantage in such endeavors, quantitative methodologies, such as Sabermetrics, have grown in importance and ubiquity as a valuable augmentation to traditional scouting methods. However, as no one person can evaluate and accurately store all of the information available from the vast volumes of sporting information generated on a daily basis, there seldom exists a storehouse of properly coded and stored information reflecting such large volumes of sports information and, even were such information available, there is lacking the provision of tools capable of mining and analyzing such information.

[0006] Systems are now available for capturing and encoding event information, such as sporting event information, such as "X, Y, Z" motion data captured by imaging cameras deployed in National Basketball Association (NBA) arenas. However, there are many challenges with such systems, including difficulty handling the data, difficulty transforming X, Y, Z data into meaningful and existing sports terminology, difficulty identifying meaningful insights from the data, difficulty visualizing results, and others. Also, there are opportunities to identify and extract novel insights from the data. Accordingly, a need exists for methods and systems that can take event data captured in video feeds and enable discovery and presentation of relevant events, metrics, analytic results, and insights.

SUMMARY

[0007] In accordance with various exemplary and non-limiting embodiments, methods and systems disclosed herein enable the exploration of event data captured from video feeds, the discovery of relevant events (such as within a video feed of a sporting event), and the presentation of novel insights, analytic results, and visual displays that enhance decision-making, provide improved entertainment and provide other benefits.

[0008] Embodiments include taking data from a video feed and enabling an automated machine understanding of a game, aligning video sources to the understanding and utilizing the video sources to automatically deliver highlights to an end-user.

[0009] In accordance with another exemplary and non-limiting embodiments, a method comprises receiving a sport playing field configuration and at least one image and determining a camera pose based, at least in part, upon the sport playing field configuration and at least one image.

[0010] In accordance with another exemplary and non-limiting embodiments, a method comprises performing automatic recognition of a camera pose based, at least in part, on video input comprising a scene and augmenting the video input with at least one of additional imagery and graphics rendered within the reconstructed 3D space of the scene.

[0011] Methods and systems described herein may include taking a video feed of an event; using machine learning to develop an understanding of the event; automatically, under computer control, aligning the video feed with the understanding; and producing a transformed video feed that includes at least one highlight that may be extracted from the machine learning of the event. In embodiments, the event may be a sporting event. In embodiments, the event may be an entertainment event. In embodiments, the event may be at least one of a television event and a movie event. In embodiments, the event may be a playground pickup game or other amateur sports game. In embodiments, the event may be any human activity or motion in a home or commercial establishment. In embodiments, the transformed video feed creates a highlight video feed of video for a defined set of players. In embodiments, the defined set of players may be a set of players from a fantasy team. Embodiments may include delivering the video feed to at least one of an inbox, a mobile device, a table, an application, a scoreboard, a Jumbotron board, a video board, and a television network.

[0012] Methods and systems described herein may include taking a source data feed relating to an event; using machine learning to develop an understanding of the event; automatically, under computer control, aligning the source feed with the understanding; and producing a transformed feed that includes at least one highlight that may be extracted from the machine learning of the event. In embodiments, the event may be a sporting event. In embodiments, the event may be an entertainment event. In embodiments, the event may be at least one of a television event and a movie event. In embodiments, the source feed may be at least one of an audio feed, a text feed, a statistics feed, and a speech feed.

[0013] Methods and systems described herein may include: taking a data set associated with a video feed of a live event; taking spatiotemporal features of the live event; applying machine learning to determine at least one spatiotemporal pattern of the event; and using a human validation process to at least one of validate and teach the machine learning of the spatiotemporal pattern. In embodiments, the event may be a sporting event.

[0014] Methods and systems described herein may include taking at least one of a video feed and an image feed; taking data relating to a known configuration of a venue; and automatically, under computer control, recognizing a camera pose based on the video feed and the known configuration. In embodiments, the venue may be a sporting event venue.

[0015] Methods and systems described herein may include taking at least one feed, selected from the group consisting of a video feed and an image feed of a scene; taking data relating to a known configuration of a venue; automatically, under computer control, recognizing a camera pose based on the video feed and the known configuration; and automatically, under computer control, augmenting the at least one feed with at least one of an image and a graphic within the space of the scene. The methods and systems may include using human input to at least one of validate and assisting the automatic recognition of the camera pose. The methods and system may include presenting at least one metric in the augmented feed. The methods and systems may include enabling a user to interact with at least one of the video feed and a frame of the video feed in a 3D user interface. The methods and systems may include augmenting the at least one feed to create a transformed feed. In embodiments, the transformed video feed may create a highlight video feed of video for a defined set of players.

[0016] Methods and systems described herein may include taking a data set associated with a video feed of a live event; taking spatiotemporal features of the live event; applying machine learning to determine at least one spatiotemporal pattern of the event; and calculating a metric based on the determined pattern. In embodiments, the metric may be at least one of a shot quality (SEFG) metric, an EFG+ metric, a rebound positioning metric, a rebounding attack metric, a rebounding conversion metric, an event-count per playing time metric, and an efficiency per event-count metric.

[0017] Methods and systems described herein may include providing an interactive, graphical user interface for exploration of data extracted by machine learning from the video capture of live events. In embodiments, the graphical user interface enables exploration and analysis of events. In embodiments, the graphical user interface is at least one of a mobile device interface, a laptop interface, a tablet interface, a large-format touchscreen interface, and a personal computer interface. In embodiments, the data may be organized to present at least one of a breakdown, a ranking, a field-based comparison and a statistical comparison. In embodiments, the exploration enables at least one of a touch interaction, a gesture interaction, a voice interaction and a motion-based interaction.

[0018] Methods and systems described herein may include taking a data set associated with a video feed of a live event; automatically, under computer control, recognizing a camera pose for the video; tracking at least one of a player and an object in the video feed; and placing the tracked items in a spatial location corresponding to spatial coordinates.

[0019] Methods and systems described herein may include taking a data set associated with a video feed of a live event; taking spatiotemporal features of the live event; applying machine learning to determine at least one spatiotemporal pattern of the event; and delivering contextualized information during the event. In embodiments, the contextualized information includes at least one of a statistic, a replay, a visualization, a highlight, a compilation of highlights, and a replay. In embodiments, the information may be delivered to at least one of a mobile device, a laptop, a tablet, and a broadcast video feed. The methods and systems may include providing a touch screen interaction with a visual representation of at least one item of the contextualized information.

BRIEF DESCRIPTION OF THE FIGURES

[0020] The following detailed description of certain embodiments may be understood by reference to the following figures:

[0021] FIG. **1** illustrates a technology stack according to an exemplary and non-limiting embodiment.

[0022] FIG. **2** illustrates a stack flow according to an exemplary and non-limiting embodiment.

[0023] FIG. **3** illustrates an exploration loop according to an exemplary and non-limiting embodiment.

[0024] FIG. **4** illustrates ranking user interface according to an exemplary and non-limiting embodiment.

[0025] FIGS. **5**A-**5**B illustrate a ranking user interface according to an exemplary and non-limiting embodiment.

[0026] FIGS. **6**A-**6**B illustrate a filters user interface according to an exemplary and non-limiting embodiment.

[0027] FIG. **7** illustrates a breakdown user interface according to an exemplary and non-limiting embodiment.

[0028] FIG. **8** illustrates a breakdown user interface according to an exemplary and non-limiting embodiment.

[0029] FIG. **9** illustrates a personalized user interface according to an exemplary and non-limiting embodiment.

[0030] FIG. **10** illustrates an alternative video user interface according to an exemplary and non-limiting embodiment.

[0031] FIG. **11** illustrates an alternative report according to an exemplary and non-limiting embodiment.

[0032] FIG. **12** illustrates a court comparison view according to an exemplary and non-limiting embodiment.

[0033] FIG. **13** illustrates a court view according to an exemplary and non-limiting embodiment.

[0034] FIG. **14** illustrates a report according to an exemplary and non-limiting embodiment.

[0035] FIG. **15** illustrates a detailed depiction of a game according to an exemplary and non-limiting embodiment.

[0036] FIG. **16** illustrates querying and aggregation according to an exemplary and non-limiting embodiment.

[0037] FIG. **17** illustrates a hybrid classification process flow according to an exemplary and non-limiting embodiment.

[0038] FIG. **18** illustrates test inputs according to an exemplary and non-limiting embodiment.

[0039] FIG. **19** illustrates test inputs according to an exemplary and non-limiting embodiment.

[0040] FIG. **20** illustrates player detection according to an exemplary and non-limiting embodiment.

[0041] FIG. **21** illustrates a process flow according to an exemplary and non-limiting embodiment.

[0042] FIG. **22** illustrates rebounding according to an exemplary and non-limiting embodiment.

[0043] FIG. **23** illustrates scatter rank according to an exemplary and non-limiting embodiment.

[0044] FIGS. **24**A-**24**B illustrate reports according to an exemplary and non-limiting embodiment.

[0045] FIG. **25** illustrates a quality assurance user interface according to an exemplary and non-limiting embodiment.

[0046] FIG. **26** illustrates a quality assurance user interface according to an exemplary and non-limiting embodiment.

[0047] FIG. **27** illustrates camera pose detection according to an exemplary and non-limiting embodiment.

[0048] FIG. **28** illustrates camera pose detection according to an exemplary and non-limiting embodiment.

[0049] FIG. **29** illustrates auto-rotoscoping according to an exemplary and non-limiting embodiment.

[0050] FIGS. **30**A-**30**C illustrate scripted storytelling with assets according to an exemplary and non-limiting embodiment.

[0051] FIG. **31** illustrates an example according to an exemplary and non-limiting embodiment.

[0052] FIG. **32** illustrates an example according to an exemplary and non-limiting embodiment.

[0053] FIG. **33** illustrates an example according to an exemplary and non-limiting embodiment.

[0054] FIG. **34** illustrates an example according to an exemplary and non-limiting embodiment.

[0055] FIG. **35** illustrates an example according to an exemplary and non-limiting embodiment.

[0056] FIG. **36** illustrates an example according to an exemplary and non-limiting embodiment.

[0057] FIG. **37** illustrates an example according to an exemplary and non-limiting embodiment.

[0058] FIG. **38** illustrates a screen shot according to an exemplary and non-limiting embodiment.

[0059] FIGS. **39**A-**39**E illustrate a screen shot according to an exemplary and non-limiting embodiment.

[0060] FIG. **40** illustrates a screen shot according to an exemplary and non-limiting embodiment.

[0061] FIGS. **41**A-**41**B illustrate a screen shot according to an exemplary and non-limiting embodiment.

[0062] FIGS. **42**A-**42**C illustrate a screen shot according to an exemplary and non-limiting embodiment.

[0063] FIG. **43** illustrates a screen shot according to an exemplary and non-limiting embodiment.

DETAILED DESCRIPTION

[0064] FIG. **1** illustrates a technology stack **100** indicative of technology layers configured to execute a set of capabilities, in accordance with an embodiment of the present invention. The technology stack **100** may include a customization layer **102**, an interaction layer **104**, a visualizations layer **108**, an analytics layer **110**, a patterns layer **112**, an events layer **114**, and a data layer **118**, without limitations. The different technology layers or the technology stack **100** may be referred to as an "Eagle" Stack **100**, which should be understood to encompass the various layers allow precise monitoring, analytics, and understanding of spatio-temporal data associated with an event, such as a sports event and the like. For example, the technology stack may provide an analytic platform that may take spatio-temporal data (e.g., 3D motion capture "XYZ" data) from National Basketball Association (NBA) arenas or other sports arenas and, after cleansing, may perform spatio-temporal pattern recognition to extract certain "events". The extracted events may be for example (among many other possibilities) events that correspond to particular understandings of events within the overall sporting event, such as "pick and roll" or "blitz." Such events may correspond to real events in a game, and may in turn be subject to various metrics, analytic tools, and visualizations around the events. Event recognition may be based on pattern recognition by machine learning, such as spatio-temporal pattern recognition, and in some cases may be augmented, confirmed, or aided by human feedback.

[0065] The customization layer **102** may allow performing custom analytics and interpretation using analytics, visualization, and other tools, as well as optional crowd-sourced feedback for developing team-specific analytics, models, exports and related insights. For example, among many other possibilities, the customization layer **102** may facilitate in generating visualizations for different spatio-temporal movements of a football player, or group of players and counter movements associated with other players or groups of players during a football event.

[0066] The interaction layer **104** may facilitate generating real-time interactive tasks, visual representations, interfaces, videos clips, images, screens, and other such vehicles for allowing viewing of an event with enhanced features or allowing interaction of a user with a virtual event derived from an actual real-time event. For example, the interaction layer **104** may allow a user to access features or metrics such as a shot matrix, a screens breakdown, possession detection, and many others using real-time interactive tools that may slice, dice and analyze data obtained from the real-time event such as a sports event.

[0067] The visualizations layer **108** may allow dynamic visualizations of patterns and analytics developed from the data obtained from the real-time event. The visualizations may be presented in the form of a scatter rank, shot comparisons, a clip view and many others. The visualizations layer **108** may use various types of visualizations and graphical tools for creating visual depictions. The visuals may include various types of interactive charts, graphs, diagrams, comparative analytical graphs and the like. The visualizations layer **108** may be linked with the interaction layer so that the visual depictions may be presented in an interactive fashion for a user interaction with real-time events produced on a virtual platform such as analytic platform of the present invention.

[0068] The analytics layer **110** may involve various analytics and Artificial Intelligence (AI) tools to perform analysis and interpretation of data retrieved from the real-time event such as a sports event so that the analyzed data results in insights that make sense out of the pulled big data from the real-time event. The analytics and AI tools may comprise such as search and optimization tools, inference rules engines, algorithms, learning algorithms, logic modules, probabilistic tools and methods, decision analytics tools, machine learning algorithms, semantic tools, expert systems and the like without limitations.

[0069] Output from the analytics **110** and patterns layers **112** is exportable by the user as a database that enables the customer to configure their own machines to read and access the events and metrics stored in the system. In accordance with various exemplary and non-limiting embodiments, patterns and metrics are structured and stored in an intuitive way. In general, the database utilized for storing the events and metric data is designed to facilitate easy export and to enable integration with a team's internal workflow. In one embodiment, there is a unique file corresponding to each individual game. Within each file, individual data structures may be configured in accordance with included structure definitions for each data type indicative of a type of event for which data may be identified and stored. For example, types of events that may be recorded for a basketball game include, but are not limited to, isos, handoffs, posts, screens, transitions, shots, closeouts and chances. With reference to, for example, the data type "screens", table **1** is an exemplary listing of the data structure for storing information related to each occurrence of a screen. As illustrated, each data type is comprised of a plurality of component variable definitions each comprised of a data type and a description of the variable.

TABLE 1

screens
id
INT
Internal ID of this screen.
possession_id
STRING
Internal ID of the possession in which this event took place.
frame
INT
Frame ID, denoting frame number from the start of the current period. Currently, this marks the frame at which the screener and ballhandler are closest.
frame_time
INT
Time stamp provided in SportVU data for a frame, measured in milliseconds in the current epoch (i.e. from 00:00:00 UTC on Jan. 1, 1970).
game_code
INT
Game code provided in SportVU data.
period
INT
Regulation periods 1-4, overtime periods 5 and up.
game_clock
NUMBER
Number of seconds remaining in period, from 720.00 to 0.00.
location_x
NUMBER
Location along length of court, from 0 to 94.
location _y
NUMBER
Location along baseline of court, from 0 to 50.
screener
INT
ID of screener, matches SportVU ID.
ballhandler
INT
ID of the ball handler, matches SportVU ID.
screener_defender
INT
ID of the screener's defender, matches SportVU ID.
ballhandler_defender
INT
ID of the ball handler's defender, matches SportVU ID.
oteam
INT
ID of team on offense, matches IDs in SportVU data.
dteam
INT
ID of team on defense, matches IDs in SportVU data.
rdef
STRING
String representing the observed actions of the ballhandler's defender.
sdef
STRING
String representing the observed actions of the screener's defender.
scr_type
STRING
Classification of the screen into take, reject, or slip.
outcomes_bhr
ARRAY
Actions by the ballhandler, taken from the outcomes described at the end of the document, such as FGX or FGM.
outcomes_scr
ARRAY
Actions by the screener, taken from the outcomes described at the end of the document, such as FGX or FGM.

[0070] These exported files, one for each game, enable other machines to read the stored understanding of the game and build further upon that knowledge. In accordance with various embodiments, the data extraction and/or export is optionally accomplished via a JSON schema.

[0071] The patterns layer **112** may provide a technology infrastructure for rapid discovery of new patterns arising out of the retrieved data from the real-time event such as a sports event. The patterns may comprise many different patterns that corresponding to an understanding of the event, such as a defensive pattern (e.g., blitz, switch, over, under, up to touch, contain-trap, zone, man-to-man, or face-up pattern), various offensive patterns (e.g., pick-and-roll, pick-and-pop, horns, dribble-drive, off-ball screens, cuts, post-up, and the like), patterns reflecting plays (scoring plays, three-point plays, "red zone" plays, pass plays, running plays, fast break plays, etc.) and various other patterns associated with a player in the game or sports, in each case corresponding to distinct spatio-temporal events.

[0072] The events layer **114** may allow creating new events or editing or correcting current events. For example, the events layer may allow analyzing accuracy of markings or other game definitions and may comment on whether they meet standards and sports guidelines. For example, specific boundary markings in an actual real-time event may not be compliant with the guidelines and there may exist some errors, which may be identified by the events layers through analysis and virtual interactions possible with the platform of the present invention. Events may corresponding to various understandings of a game, including offensive and defensive plays, matchups among players or groups of players, scoring events, penalty or foul events, and many others.

[0073] The data layer **118** facilitates management of the big data retrieved from the real-time event such as a sports event. The data layer **118** may allow creating libraries that may store raw data, catalogues, corrected data, analyzed data, insights and the like. The data layer **118** may manage online warehousing in a cloud storage setup or in any other manner in various embodiments.

[0074] FIG. **2** illustrates a process flow diagram **200**, in accordance with an embodiment of the present invention. The process **200** may include retrieving spatio-temporal data associated with a sports or game and storing in a data library at step **202**. The spatio-temporal data may relate to a video feed that was captured by a 3D camera, such as one positioned in a sports arena or other venue, or it may come from another source.

[0075] The process **200** may further include cleaning of the rough spatio-temporal data at step **204** through analytical and machine learning tools and utilizing various technology layers as discussed in conjunction with FIG. **1** so as to generate meaningful insights from the cleansed data.

[0076] The process **200** may further include recognizing spatio-temporal patterns through analysis of the cleansed data at step **208**. Spatio-temporal patterns may comprise a wide range of patterns that are associated with types of events. For example, a particular pattern in space, such as the ball bouncing off the rim, then falling below it, may contribute toward recognizing a "rebound" event in basketball. Patterns in space and time may lead to recognition of single events, or multiple events that comprise a defined sequence of recognized events (such as in types of plays that have multiple steps).

[0077] The recognized patterns may define a series of events associated with the sports that may be stored in an event datastore at step **210**. These events may be organized according to the recognized spatio-temporal patterns; for example, a series of events may have been recognized as "pick," "rebound," "shot," or like events in basketball, and they may be stored as such in the event datastore **210**. The event datastore **210** may store a wide range of such events, including individual patterns recognized by spatiotemporal pattern recognitions and aggregated patterns, such as when one pattern follows another in an extended, multi-step event (such as in plays where one event occurs and then another occurs, such as "pick and roll" or "pick and pop" events in basketball, football events that involve setting an initial block, then springing out for a pass, and many others).

[0078] The process **200** may further include querying or aggregation or pattern detection at step **212**. The querying of data or aggregation may be performed with the use of search tools that may be operably and communicatively connected with the data library or the events datastore for analyzing, searching, aggregating the rough data, cleansed or analyzed data, or events data or the events patterns.

[0079] At step **214**, metrics and actionable intelligence may be used for developing insights from the searched or aggregated data through artificial intelligence and machine learning tools.

[0080] At step **218**, for example, the metrics and actionable intelligence may convert the data into interactive visualization portals or interfaces for use by a user in an interactive manner.

[0081] Raw input XYZ data obtained from various data sources is frequently noisy, missing, or wrong. XYZ data is sometimes delivered with attached basic events already identified in it, such as possession, pass, dribble, and shot events; however, these associations are frequently incorrect. This is important because event identification further down the process (in Spatiotemporal Pattern Recognition) sometimes depends on the correctness of these basic events. For example, if two players' XY positions are switched, then "over" vs "under" defense would be incorrectly characterized, since the players' relative positioning is used as a critical feature for the classification. Even player-by-player data sources are occasionally incorrect, such as associating identified events with the wrong player.

[0082] First, validation algorithms are used to detect all events, including the basic events such as possession, pass, dribble, shot, and rebound that are provided with the XYZ data. Possession/Non-possession models may use a Hidden Markov Model to best fit the data to these states. Shots and rebounds may use the possession model outputs, combined with 1) projected destination of the ball, and 2) player by player information (PBP) information. Dribbles may be identified using a trained ML algorithm and also using the output of the possession model. These algorithms may decrease the basic event labeling error rate by approximately 50% or more.

[0083] Second, the system has a library of anomaly detection algorithms to identify potential problems in the data including, but not limited to, temporal discontinuities (intervals of missing data are flagged), spatial discontinuities (objects traveling is a non-smooth motion, "jumping") and interpolation detection (data that is too smooth, indicating that post-processing was done by the data supplier to interpolate between known data points in order to fill in missing data). This problem data is flagged for human review, so that events detected during these periods are subject to further scrutiny.

[0084] Spatiotemporal Pattern Recognition

[0085] Spatiotemporal pattern recognition **208** is used to automatically identify relationships between physical and temporal patterns and various types of events. In the example of basketball, one challenge is how to turn x, y, z positions of ten players and one ball at twenty-five frames/sec into usable input for machine learning and pattern recognition algorithms. For patterns one is trying to detect (e.g. pick & rolls), the raw inputs may not suffice. The instances within each pattern category can look very different from each other. One therefore may benefit from a layer of abstraction and generality. Features that relate multiple actors in time are key components to the input. Examples include, but are not limited to, the motion of player one (P1) towards player two (P2), for at least T seconds, a rate of motion of at least V m/s for at least T seconds and at the projected point of intersection of paths A and B, and a separation distance less than D.

[0086] In embodiments of the present disclosure, there is provided a library of such features involving multiple actors over space and time. In the past machine learning (ML) literature, there has been relatively little need for such a library of spatiotemporal features, because there were few datasets with these characteristics on which learning could have been considered as an option. The library may include relationships between actors (e.g., players one through ten in basketball), relationships between the actors and other objects such as the ball, and relationships to other markers, such as designated points and lines on the court or field, and to projected locations based on predicted motion.

[0087] Another key challenge is there have not been a labeled dataset for training the ML algorithms. Such a labeled dataset may be used in connection with various embodiments disclosed herein. For example, there has previously been no XYZ player-tracking dataset that already has higher level events, such as pick and roll (P&R) events) labeled at each time frame they occur. Labeling such events, for many different types of events and sub-types, is a laborious process. Also, the number of training examples required to adequately train the classifier may be unknown. One may use a variation of active learning to solve this challenge. Instead of using a set of labeled data as training input for a classifier trying to distinguish A and B, the machine finds an unlabeled example that is closest to the boundary between As and Bs in the feature space. The machine then queries a human operator/labeler for the label for this example. It uses this labeled example to refine its classifier, and then repeats.

[0088] In one exemplary embodiment of active learning, the system also incorporates human input in the form of new features. These features are either completely devised by the human operator (and inputted as code snippets in the active learning framework), or they are suggested in template form by the framework. The templates use the spatiotemporal pattern library to suggest types of features that may be fruitful to test. The operator can choose a pattern, and test a particular instantiation of it, or request that the machine test a range of instantiations of that pattern.

Multi-Loop Iterative Process

[0089] Some features are based on outputs of the machine learning process itself. Thus, multiple iterations of training are used to capture this feedback and allow the process to converge. For example, a first iteration of the ML process may suggest that the Bulls tend to ice the P&R. This fact is then fed into the next iteration of ML training as a feature, which biases the algorithm to label Bulls' P&R defense as ices. The process converges after multiple iterations. In practice, two iterations has typically been sufficient to yield good results.

[0090] In accordance with exemplary embodiments, a canonical event datastore **210** may contain a definitive list of events that the system knows occurred during a game. This includes events extracted from the XYZ data, as well as those specified by third-party sources, such as PBP data from various vendors. The events in the canonical event datastore **210** may have game clock times specified for each event. The datastore **210** may be fairly large. To maintain efficient processing, it is shared and stored in-memory across many machines in the cloud. This is similar in principle to other methods such as Hadoop™; however, it is much more efficient, because in embodiments involving events, such as sporting events, where there is some predetermined structure that is likely to be present (e.g., the 24-second shot clock, or quarters or halves in a basketball game), it makes key structural assumptions about the data. Because the data is from sports games, for example, in embodiments one may enforce that no queries will run across multiple quarters/periods. Aggregation steps can occur across quarters/periods, but query results will not. This is one instantiation of this assumption. Any other domain in which locality of data can be enforced will also fall into this category.

[0091] Such a design allows rapid and complex querying across all of the data, allowing arbitrary filters, rather than relying on either 1) long-running processes, or 2) summary data, or 3) pre-computed results on pre-determined filters.

[0092] In accordance with exemplary and non-limiting embodiments, data is divided into small enough shards that each worker shard has a low latency response time. Each distributed machine may have multiple workers corresponding to the number of processes the machine can support concurrently. Query results never rely on more than one shard, since we enforce that events never cross quarter/period boundaries. Aggregation functions all run incrementally rather than in batch process, so that as workers return results, these are incorporated into the final answer immediately. To handle results such as rankings pages, where many rows must be returned, the aggregator uses hashes to keep track of the separate rows and incrementally updates them.

[0093] Referring to FIG. **3**, an exploration loop may be enabled by the methods and systems disclosed herein, where questioning and exploration can occur, such as using visualizations (e.g., data effects, referred to as DataFX in this disclosure), processing can occur, such as to identify new events and metrics, and understanding emerges, leading to additional questions, processing and understanding.

[0094] Referring to FIG. **4**, the present disclosure provides an instant player rankings feature as depicted in the illustrated user interface. A user can select among various types of available rankings **402**, as indicated in the drop down list **410**, such as rankings relating to shooting, rebounding, rebound ratings, isolations (Isos), picks, postups, handoffs, linups, matchups, possessions (including metrics and actions), transitions, plays and chances. Rankings can be selected in a menu element **404** for players, teams or other entities. Rankings can be selected for different types of play in the menu element **408**, such as for offense, defense, transition, special situations, and the like. The ranking interface allows a user to quickly query the system to answer a particular question instead of thumbing through pages of reports. The user interface lets a user locate essential factors and evaluate talent of a player to make more informed decisions.

[0095] FIGS. **5**A-**5**B shows certain basic, yet quite in-depth, pages in the systems described herein, referred to in some cases as the "Eagle system." This user interface may allow the user to rank players and teams by a wide variety of metrics. This may include identified actions, metrics derived from these actions, and other continuous metrics. Metrics may relate to different kinds of events, different entities (players and teams), different situations (offense and defense) and any other patterns identified in the spatiotemporal pattern recognition system. Examples of items on which various entities can be ranked in the case of basketball include chances, charges, closeouts, drives, frequencies, handoffs, isolations, lineups, matches, picks, plays, possessions, postups, primary defenders, rebounding (main and raw), off ball screens, shooting, speed/load and transitions.

[0096] The Rankings UI makes it easy for a user to understand relative quality of one row item versus other row items, along any metric. Each metric may be displayed in a column, and that row's ranking within the distribution of values for that metrics may be displayed for the user. Color coding makes it easy for the user to understand relative goodness.

[0097] FIGS. **6**A-**6**B show a set of filters in the UI, which can be used to filter particular items to obtain greater levels of detail or selected sets of results. Filters may exist for seasons, games, home teams, away teams, earliest and latest date, postseason/regular season, wins/losses, offense home/away, offensive team, defensive team, layers on the court for offense/defense, players off court for offense/defense, locations, offensive or defensive statistics, score differential, periods, time remaining, after timeout play start, transition/no transition, and various other features. The filters **602** for offense may include selections for the ballhandler, the ballhandler position, the screener, the screener position, the ballhandler outcome, the screener outcome, the direction, the type of pick, the type of pop/roll, the direction of the pop/roll, and presence of the play (e.g., on the wing or in the middle). Many other examples of filters are possible, as a filter can exist for any type of parameter that is tracked with respect to an event that is extracted by the system or that is in the spatiotemporal data set used to extract events. The present disclosure also allows situational comparisons. The user interface allows a user to search for a specific player that may fit into offense. The highly accurate dataset and easy to use interface allows the user to compare similar players in similar situations. The user interface may allow the user to explore player tendencies. The user interface may allow locating shot locations and also may provide advanced search capabilities.

[0098] Filters enable users to subset the data in a large number of ways, and immediately receive metrics calculated on the subset. Using multiple loops for convergence in machine learning enables the system to return the newly filtered data and metrics in real-time, whereas existing methods would require minutes to re-compute the metrics given the filters, leading to inefficient exploration loops (FIG. **3**). Given that the data exploration and investigation process often requires many loops, these inefficiencies can otherwise add up quickly.

[0099] As illustrated with reference to FIGS. **6**A-**6**B, there are many filters that may enable a user to select specific situations of interest to analyze. These filters may be categorized in logical groups, including, but not limited to, Game, Team, Location, Offense, Defense, and Other. The possible filters may automatically change depending on the type of event being analyzed, for example, Shooting, Rebounding, Picks, Handoffs, Isolations, Postups, Transitions, Closeouts, Charges, Drives, Lineups, Matchups, Play Types, Possessions.

[0100] For all event types, under the Game category, filters may include Season, specific Games, Earliest Date, Latest Date, Home Team, Away Team, where the game is being played Home/Away, whether the outcome was Wins/Losses, whether the game was a Playoff game, and recency of the game.

[0101] For all event types, under the Team category, filters may include Offensive Team, Defensive Team, Offensive Players on Court, Defenders Players on Court, Offensive Players Off Court, Defenders Off Court.

[0102] For all event types, under the Location category, the user may be given a clickable court map that is segmented into logical partitions of the court. The user may then select any number of these partitions in order to filter only events that occurred in those partitions.

[0103] For all event types, under the Other category, the filters may include Score Differential, Play Start Type (Multi-Select: Field Goal ORB, Field Goal DRB, Free Throw ORB, Free Throw DRB, Jump Ball, Live Ball Turnover, Defensive Out of Bounds, Sideline Out of Bounds), Periods, Seconds Remaining, Chance After Timeout (T/F/ALL), Transition (T/F/ALL).

[0104] For Shooting, under the Offense category, the filters may include Shooter, Position, Outcome (Made/Missed/All), Shot Value, Catch and Shoot (T/F/ALL), Shot Distance, Simple Shot Type (Multi-Select: Heave, Angle Layup, Driving Layup, Jumper, Post), Complex Shot Type (Multi-Select: Heave, Lob, Tip, Standstill Layup, Cut Layup, Driving Layup, Floater, Catch and Shoot), Assisted (T/F/ALL), Pass From (Player), Blocked (T/F/ALL), Dunk (T/F/ALL), Bank (T/F/ALL), Goaltending (T/F/ALL), Shot Attempt Type (Multi-select: FGA No Foul, FGM Foul, FGX Foul), Shot SEFG (Value Range), Shot Clock (Range), Previous Event (Multi-Select: Transition, Pick, Isolation, Handoff, Post, None).

[0105] For Shooting, under the Defense category, the filters may include Defender Position (Multi-Select: PG, SG, SF, PF, CTR), Closest Defender, Closest Defender Distance, Blocked By, Shooter Height Advantage.

[0106] For Picks, under the Offense category, the filters may include Ballhandler, Ballhandler Position, Screener, Screener Position, Ballhandler Outcome (Pass, Shot, Foul, Turnover), Screener Outcome (Pass, Shot, Foul, Turnover), Direct or Indirect Outcome, Pick Type (Reject, Slip, Pick), Pop/Roll, Direction, Wing/Middle, Middle/Wing/Step-Up.

[0107] For Picks, under the Defense category, the filters may include Ballhandler Defender, Ballhandler Defender Position, Screener Defender, Screener Defender Position, Ballhandler Defense Type (Over, Under, Blitz, Switch, Ice), Screener Defense Type (Soft, Show, Ice, Blitz, Switch), Ballhandler Defense (Complex) (Over, Under, Blitz, Switch, Ice, Contain Trap, Weak), Screener Defense (Complex) (Over, Under, Blitz, Switch, Ice, Contain Trap, Weak, Up to Touch).

[0108] For Drives, under the Offense category, the filters may include Ballhandler, Ballhandler Position, Ballhandler Outcome, Direct or Indirect, Drive Category (Handoff, Iso, Pick, Closeout, Misc.), Drive End (Shot Near Basket, Pullup, Interior Pass, Kickout, Pullout, Turnover, Stoppage, Other), Direction, Blowby (T/F).

[0109] For Drives, under the Defense category, the filters may include Ballhandler Defender, Ballhandler Defender Position, Help Defender Present (T/F), Help Defenders.

[0110] For most other events, under the Offense category, the filters may include Ballhandler, Ballhandler Position, Ballhandler Outcome, Direct or Indirect.

[0111] For most other events, under the Defense category, the filters may include Ballhandler Defender, Ballhandler Defender Position.

[0112] For Postups, under the Offense category, the filters may additionally include Area (Left, Right, Middle).

[0113] For Postups, under the Defense category, the filters may additionally include Double Team (T/F).

[0114] The present disclosure provides detailed analysis capabilities, such as through the depicted user interface embodiment of FIG. **7**. In an example depicted in the FIG. **7**, the user interface may be used to know if a player should try and ice the pick and roll or not between two players. Filters can go from all picks, to picks involving a selected player as ballhandler, to picks involving that ballhandler with a certain screener, to the type of defense played by that screener. By filtering down to particular matchups (by player combinations and actions taken), the system allows rapid exploration of the different options for coaches and players, and selection of preferred actions that had the best outcomes in the past. Among other things, the system may give detailed breakdown of a player's opponent and a better idea of what to expect during a game. The user interface may be used to know and highlight opponent capabilities. A breakdowns UI may make it easy for a user to drill down to a specific situation, all while gaining insight regarding frequency and efficacy of relevant slices through the data.

[0115] The events captured by the present system may be capable of being manipulated using the UI. FIG. **8** shows a visualization, where a dropdown feature **802** allows a user to select various parameters related to the ballhandler, such as to break down to particular types of situations involving that ballhandler. These types of "breakdowns" facilitate improved interactivity with video data, including enhanced video data created with the methods and systems disclosed herein. Most standard visualizations are static images. For large and complex datasets, especially in cases where the questions to be answered are unknown beforehand, interactivity enables the user to explore the data, ask new questions, get new answers. Visualizations may be color coded good (e.g., orange) to bad (e.g., blue) based on outcomes in particular situations for easy understanding without reading the detailed numbers. Elements like the sizes of partitions can be used, such as to denote frequency. Again, a user can comprehend significance from a glance. In embodiments, each column represents a variable for partitioning the dataset. It is easy for a user to add, remove, and re-arrange columns by clicking and dragging. This makes it easy to experiment with different visualizations. Furthermore, the user can drill into a particular scenario by clicking on the partition of interest, which zooms into that partition, and redraws the partitions in the columns to the right so that they are re-scaled appropriately. This enables the user to view the relative sample sizes of the partitions in columns to the right, even when they are small relative to all possible scenarios represented in columns further to the left. In embodiments, a video icon takes a user to video clips of the set of plays that correspond to a given partition. Watching the video gives the user ideas for other variables to use for partitioning.

[0116] Various interactive visualizations may be created to allow users to better understand insights that arise from the classification and filtering of events, such as ones that emphasize color coding for easy visual inspection and detection of anomalies (e.g. a generally good player with lots of orange but is bad/blue in one specific dimension). Conventionally, most standard visualizations are static images. However, for large and complex datasets, especially in cases where the questions to be answered are unknown beforehand, interactivity enables the user to explore the data, ask new questions, get new answers. For example, a breakdown view may be color coded good (orange) to bad (blue) for easy understanding without reading the numbers. Sizes of partitions may denote frequency of events. Again, one can comprehend from a glance the events that occur most frequently. Each column of a visualization may represent a variable for partitioning the dataset. It may be easy to add, remove, and re-arrange columns by clicking and dragging. This makes it easy to experiment with possible visualizations. In embodiments, a video icon may take a user to video clips, such as of the set of plays that correspond to that partition. Watching the video gives the user ideas for other variables to use for partitioning.

[0117] In embodiments, a ranking view is provided. Upon mousing over each row of a ranking view, histograms above each column may give the user a clear contextual understanding that row's performance for each column variable. The shape of a distribution is often informative. Color-coded bars within each cell may also provide a view of each cell's performance that is always available, without mousing over. Alternatively, the cells themselves may be color-coded.

[0118] The system may provide a personalized video in embodiments of the methods and systems described herein. For example, with little time to scout the opposition, the system can provide a user relevant information to quickly prepare team. The team may rapidly retrieve the most meaningful plays, cut and compiled to specific needs of players. The system may provide immediate video cut-ups. In embodiments, the present disclosure provides a video that is synchronized with identified actions. For example, if spatiotemporal machine learning identifies a segment of video as showing a pick and roll involving two players, then that video segment may be tagged, so that when that event is found (either by browsing or by filtering to that situation), the video can be displayed. Because the machine understands the precise moment that an event occurs in the video, a user-customizable segment of video can be created. For example, the user can retrieve video corresponding to x seconds before, and y seconds after, each event occurrence. Thus, video may be tagged and associated with events. The present disclosure may provide a video that may allow customization by numerous filters of the type disclosed above, relating to finding video that satisfies various parameters, that displays various events, or combinations thereof. For example, in embodiments, an interactive interface provided by the present disclosure allows watching videos clips for specific game situations or actions.

[0119] Reports may provide a user with easy access to printable pages summarizing pre-game information about an opponent, scouting report for a particular player, or a post-game summary. For example, the reports may collect actionable useful information in one to two easy-to-digest pages. These pages may be automatically scheduled to be sent to other staff members, e.g. post-game reports sent to coaches after each game. Referring to FIG. **11**, a report may include statistics for a given player, as well as visual representations, such as of locations **1102** where shots were taken, including shots of a particular type (such as catch and shoot shots).

[0120] The UI as illustrated in FIG. **12** provides a court comparison view **1202** among several parts of a sports court (and can be provided among different courts as well). For example, filters **1204** may be used to select the type of statistic to show for a court. Then statistics can be filtered to show results filtered by left side **1208** or right side **1214**. Where the statistics indicate an advantage, the advantages can be shown, such as of left side advantages **1210** and right side advantages **1212**.

[0121] In sports, the field of play is an important domain constant or elements. Many aspects of the game are best represented for comparison on a field of play. In embodiments, a four court comparison view **1202** is a novel way to compare two players, two teams, or other entities, to gain an overview view of each player/team (Leftmost and Rightmost figures) **1208**, **1214** and understand each one's strengths/weaknesses (Left and Right Center figures **1210**, **1212**).

[0122] The court view UI **1302** as illustrated in FIG. **13** provides a court view **1304** of a sport arena **1304**, in accordance with an embodiment of the present disclosure. Statistics for very specific court locations can be presented on a portion **1308** of the court view. The UI may provide a view of custom markings, in accordance with an embodiment of the present invention.

[0123] Referring to FIG. **14**, filters may enable users to subset the data in a large number of ways, and immediately receive metrics calculated on the subset. Descriptions of particular events may be captured and made available to users.

[0124] Various events may be labeled in a game, as reflected in FIG. **15**, which provides a detailed view of a timeline **1502** of a game, broken down by possession **1504**, by chances **1508**, and by specific events **1510** that occurred along the timeline **1502**, such as determined by spatiotemporal pattern recognition, by human analysis, or by a combination of the two. Filter categories available by a user interface of the present disclosure may include ones based on seasons, games, home teams, away teams, earliest date, latest date, postseason/regular season, wins/losses, offense home/away, offensive team, defensive team, players on the court for offense/defense, players off court for offense/defense, location, score differential, periods, time remaining, play type (e.g., after timeout play) and transition/no transition. Events may include ones based on primitive markings, such as shots, shots with a corrected shot clock, rebounds, passes, possessions, dribbles, and steals, and various novel event types, such as SEFG (shot quality), EFG+, player adjusted SEFG, and various rebounding metrics, such as positioning, opportunity percentage, attack, conversion percentage, rebounding above position (RAP), attack+, conversion+ and RAP+. Offensive markings may include simple shot types (e.g., angled layup, driving layup, heave, post shot, jumper), complex shot types (e.g., post shot, heave, cut layup, standstill layup, lob, tip, floater, driving layup, catch and shoot stationary, catch and shoot on the move, shake & raise, over screen, pullup and stepback), and other information relating to shots (e.g., catch and shoot, shot clock, 2/3S, assisted shots, shooting foul/not shooting foul, made/missed, blocked/not blocked, shooter/defender, position/defender position, defender distance and shot distance). Other events that may be recognized, such as through the spatiotemporal learning system, may include ones related to picks (ballhandler/screener, ballhandler/screener defender, pop/roll, wing/middle, step-up screens, reject/slip/take, direction (right/left/none), double screen types (e.g., double, horns, L, and handoffs into pick), and defense types (ice, blitz, switch, show, soft, over, under, weak, contain trap, and up to touch), ones related to handoffs (e.g., receive/setter, receiver/setter defender, handoff defense (ice, blitz, switch, show, soft, over, or under), handback/dribble handoff, and wing/step-up/middle), ones related to isolations (e.g., ballhandler/defender and double team), and ones related to post-ups (e.g., ballhandler/defender, right/middle/left and double teams).

[0125] Defensive markings are also available, such as ones relating to closeouts (e.g. ballhandler/defender), rebounds (e.g., players going for rebounds (defense/offense)), pick/handoff defense, post double teams, drive blow-bys and help defender on drives), ones relating to off ball screens (e.g., screener/cutter and screener/cutter defender), ones relating to transitions (e.g. when transitions/fast breaks occur, players involved on offense and defense, and putback/no putback), ones relating to how plays start (e.g., after timeout/not after timeout, sideline out of bounds, baseline out of bounds, field goal offensive rebound/defensive rebound, free throw offensive rebound/defensive rebound and live ball turnovers), and ones relating to drives, such as ballhandler/defender, right/left, blowby/no blowby, help defender presence, identity of help defender, drive starts (e.g., handoff, pick, isolation or closeout) and drive ends (e.g., shot near basket, interior pass, kickout, pullup, pullout, stoppage, and turnover). These examples and many others from basketball and other sports may be defined, based on any understanding of what constitutes a type of event during a game. Markings may relate to off ball screens (screener/cutter), screener/cutter defender, screen types (down, pro cut, UCLA, wedge, wide pin, back, flex, clip, zipper, flare, cross, and pin in).

[0126] FIG. **16** shows a system **1602** for querying and aggregation. In embodiments, data is divided into small enough shards that each worker has low latency response time. Each distributed machine may have multiple workers corresponding to the number of processes the machine can support concurrently. Query results never rely on more than one shard, since we enforce that events never cross quarter/period boundaries. Aggregation functions all run incrementally rather than in batch process, so that as workers return results, these are incorporated into the final answer immediately. To handle results such as rankings pages, where many rows must be returned, the aggregator uses hashes to keep track of the separate rows and incrementally updates them.

[0127] FIG. **17** shows a process flow for a hybrid classification process that uses human labelers together with machine learning algorithms to achieve high accuracy. This is similar to the flow described above in connection with FIG. **2**, except with the explicit inclusion of the human-machine validation process. By taking advantage of aligned video as described herein, one may provide an optimized process for human validation of machine labeled data. Most of the components are similar to those described in connection with FIG. **2** and in connection with the description of aligned video, such as the XYZ data source **1702**, cleaning process **1704**, spatiotemporal pattern recognition module **1712**, event processing system **1714**, video source **1708**, alignment facility **1710** and video snippets facility **1718**. Additional components include a validation and quality assurance process **1720** and an event-labeling component **1722**. Machine learning algorithms are designed to output a measure of confidence. For the most part, this corresponds to the distance from a separating hyperplane in the feature space. In embodiments, one may define a threshold for confidence. If an example is labeled by the machine and has confidence above the threshold, the event goes into the canonical event datastore **210** and nothing further is done. If an example has a confidence score below the threshold, then the system may retrieve the video corresponding to this candidate event, and ask a human operator to provide a judgment. The system asks two separate human operators for labels. If the given labels agree, the event goes into the canonical event datastore **210**. If they do not, a third person, known as the supervisor, is contacted for final opinion. The supervisor's decision may be final. The canonical event datastore **210** may contain both human marked and completely automated markings. The system may use both types of marking to further train the pattern recognition algorithms. Event labeling is similar to the canonical event datastore **210**, except that sometimes one may either 1) develop the initial gold standard set entirely by hand, potentially with outside experts, or 2) limit the gold standard to events in the canonical event datastore **210** that were labeled by hand, since biases may exist in the machine labeled data.

[0128] FIG. **18** shows test video input for use in the methods and systems disclosed herein, including views of a basketball court from simulated cameras, both simulated broadcast camera views **1802** as well as purpose-mounted camera views **1804**.

[0129] FIG. **19** shows additional test video input for use in the methods and systems disclosed herein, including input from broadcast video **1902** and from purpose-mounted cameras **1904** in a venue. Referring to FIG. **20**, probability maps **2004** may be computed based on likelihood there is a person standing at each x, y location.

[0130] FIG. **21** shows a process flow of an embodiment of the methods and systems described herein. Initially, in an OCR process **2118**, machine vision techniques are used to automatically locate the "score bug" and determine the location of the game clock, score, and quarter information. This information is read and recognized by OCR algorithms. Post-processing algorithms using various filtering techniques are used to resolve issues in the OCR. Kalman filtering/HMMs used to detect errors and correct them. Probabilistic outputs (which measure degree of confidence) assist in this error detection/correction. Next, in a refinement process **2120**, sometimes, a score bug is non-existent or cannot be detected automatically (e.g. sometimes during PIP or split screens). In these cases, remaining inconsistencies or missing data is resolved with the assistance of human input. Human input is designed to be sparse so that labelers do not have to provide input at every frame. Interpolation and other heuristics are used to fill in the gaps. Consistency checking is done to verify game clock. Next, in an alignment process, **2112** the Canonical Datastore **2110** (referred to elsewhere in this disclosure alternatively as the event datastore) contains a definitive list of events that the system knows occurred during a game. This includes events extracted from the XYZ data **2102**, such as after cleansing **2104** and spatiotemporal pattern recognition **2108**, as well as those specified by third-party sources such as player-by-player data sets **2106**, such as available from various vendors. Differences among the data sources can be resolved, such as by a resolver process. The events in the canonical datastore **2110** may have game clock times specified for each event. Depending on the type of event, the system knows that the user will be most likely to be interested in a certain interval of game play tape before and after that game clock. The system can thus retrieve the appropriate interval of video for the user to watch.

[0131] One challenge pertains to the handling of dead ball situations and other game clock stoppages. The methods and systems disclosed herein include numerous novel heuristics to enable computation of the correct video frame that shows the desired event, which has a specified game clock, and which could be before or after the dead ball, since those frames have the same game clock. The game clock is typically specified only at the one-second level of granularity, except in the final minute of each quarter.

[0132] Another advance is to use machine vision techniques to verify some of the events. For example: video of a made shot will typically show the score being increased, or will show a ball going through a hoop. Either kind of automatic observation serves to help the alignment process result in the correct video frames being shown to the end user.

[0133] Next, in a query UI component **2130**, the UI enables a user to quickly and intuitively request all video clips associated with a set of characteristics: player, team, play type, ballhandler, ballhandler velocity, time remaining, quarter, defender, etc. In addition, when a user is watching a video clip, the user can request all events that are similar to whatever just occurred in the video. The system uses a series of cartoon-like illustration to depict possible patterns that represent "all events that are similar." This enables the user to choose the intended pattern, and quickly search for other results that match that pattern.

[0134] Next, the methods and systems may enable delivery of enhanced video, or video snips **2124**, which may include rapid transmission of clips from stored data in the cloud. The system may store video as chunks (e.g., one minute chunks), such as in AWS S3, with each subsequent file overlapping with a previous file, such as by 30 seconds. Thus, each video frame may be stored twice. Other instantiations of the system may store the video as different sized segments, with different amounts of overlap, depending on the domain of use. In embodiments, each video file is thus kept at a small size. The 30-second duration of overlap may be important because most basketball possessions (or chances in our terminology) do not last more than 24 seconds. Thus, each chance can be found fully contained in one video file, and in order to deliver that chance, the system does not need to merge content from multiple video files. Rather, the system simply finds the appropriate file that contains the entire chance (which in turn contains the event that is in the query result), and returns that entire file, which is small. With the previously computed alignment index, the system is also able to inform the UI to skip ahead to the appropriate frame of the video file in order to show the user the query result as it occurs in that video file. This delivery may occur using AWS S3 as the file system, the Internet as transport, and a browser-based interface as the UI. It may find other instantiations with other storage, transport, and UI components.

[0135] FIG. **22** shows certain metrics that can be extracted using the methods and systems described herein, relating to rebounding in basketball. These metrics include positioning metrics, attack metrics, and conversion metrics. For positioning, the methods and systems described herein first address how to value the initial position of the players when the shot is taken. This is a difficult metric to establish. The methods and systems disclosed herein may give a value to the real estate that each player owns at the time of the shot. This breaks down into two questions: (1) what is the real estate for each player? (2) what is it worth? To address the first question, one may apply the technique of using Voronoi (or Dirichlet) tessellations. Voronoi tessellations are often applied to problems involving spatial allocation. These tessellations partition a space into Voronoi cells given a number of points in that space. For any point, it is the intersection of the self-containing halfspaces defined by hyper-planes equidistant from that point to all other points. That is, a player's cell is all the points on the court that are closer to the player than any other player. If all players were equally capable they should be able to control any rebound that occurred in this cell. One understands that players are not equally capable however this establishment of real estate is to set a baseline for performance. Over performance or under performance of this baseline will be indicative of their ability. To address the second question, one may condition based on where the shot was taken and calculate a spatial probability distribution of where all rebounds for similar shots were obtained. For each shot attempt, one may choose a collection of shots closest to the shot location that provides enough samples to construct a distribution. This distribution captures the value of the real estate across the court for a given shot. To assign each player a value for initial positioning, i.e., the value of the real estate at the time of the shot, one may integrate the spatial distribution over the Voronoi cell for that player. This yields the likelihood of that player getting the rebound if no one moved when the shot was taken and they controlled their cell. We note that because we use the distribution of location of the rebound conditioned on the shot, it is not a matter of controlling more area or even necessarily area close to the basket, but the most valuable area for that shot. While the most valuable areas are almost always close to the basket, there are some directional effects.

[0136] For an attack or hustle metric, one may look at phases following a shot, such as an initial crash phase. To analyze this, one may look at the trajectory of the ball and calculate the time that it gets closest to the center of the rim. At this point, one may reapply the Voronoi-based analysis and calculate the rebound percentages of each player, i.e., the value of the real the estate that each player has at the time the ball hits the rim. The change in this percentage from the time the shot is taken to the time it hits the rim is the value or likelihood the player had added during the phase. Players can add value by crashing the boards, i.e., moving closer to the basket towards places where the rebound is likely to go, or by blocking out, i.e., preventing other players by taking valuable real estate that is already established. A useful, novel metric for the crash phase is generated by subtracting the rebound probability at the shot from the rebound probability at the rim. The issue is that the ability to add probability is not independent from the probability at the shot. Consider a case of a defensive player who plays close to the basket. The player is occupying high value real estate, and once the shot is taken, other players are going to start coming into this real estate. It is difficult for players with high initial positioning value to have positive crash deltas. Now consider a player out by the three-point line. Their initial value is very low and moving any significant distance toward the rim will give them a positive crash delta. Thus, it is not fair to compare these players on the same scale. To address this, one may look at the relationship of the raw crash deltas (the difference between the probability at rim and probability at shot) compared to the probability at shot. In order to normalize for this effect, one may subtract the value of the regression at the player's initial positioning value from the raw crash delta to form the players Crash value. Intuitively, the value indicates how much more probability is added by this player beyond what a player with similar initial positioning would add. One may apply this normalization methodology to all the metrics the initial positioning affects the other dimensions and it can be beneficial to control for it.

[0137] A player has an opportunity to rebound the ball if they are the closest player to the ball once the ball gets below ten feet (or if they possess the ball while it is above ten feet). The player with the first opportunity may not get the rebound so multiple opportunities could be created after a single field goal miss. One may tally the number of field goal misses for which a player generated an opportunity for themselves and divided by the number of field goals to create an opportunity percentage metric. This indicates the percentage of field goal misses for which that player ended up being closest to the ball at some point. The ability for a player to generate opportunities beyond his initial position is the second dimension of rebounding: Hustle. Again, one may then apply the same normalization process as described earlier for Crash.

[0138] The reason that there are often multiple opportunities for rebounds for every missed shot is that being closest to the ball does not mean that a player will convert it into a rebound. Thus, the third dimension of rebounding, conversion. The raw conversion metric for players is calculated simply by dividing the number of rebounds obtained by the number of opportunities generated.

[0139] Formally, given a shot s described by its 2D coordinates on the court, s_x and s_y, which is followed by a rebound r, also described by its coordinates on the court of r_x and r_y, one may estimate P(r_y, r_x|s_x, s_y), the probability density of the rebound occurring at each position on the court given its shot location.

[0140] This may be accomplished by first discretizing the court into, for example, 156 bins, created by separating the court into 13 equally spaced columns, and 12 equally spaced rows. Then, given some set S of shots from a particular bin, the rebounds from S will be distributed in the bins of the court according to a multinomial distribution. One may then apply maximum likelihood estimation to determine the probability of a rebound in each of the bins of the court, given the training set S. This process may be performed for bins that shots may fall in, giving 156 distributions for the court.

[0141] Using these distributions one may determine P(r_y, r_x|s_x, s_y). First the shot s is mapped to an appropriate bin. The probability distribution determined in the previous step is then utilized to determine the probability of the shot being rebounded in every bin of the court. One assumes that within a particular bin, the rebound is uniformly likely to occur in any coordinate. Thus a probability density of the probability of the rebound falling in the bin is assigned to all points in the bin.

[0142] Using the probability density P(r_y, r_x|s_x, s_y), one may determine the probability that each particular player grabs the rebound given their location and the position of the other players of the court.

[0143] To accomplish this, one may first create a Voronoi diagram of the court, where the set of points is the location (p_x, p_y) for each player on the court. In such a diagram, each player is given a set of points that they control. Formally one may characterize the set of points that player P_k controls in the following manner, where X is all points on the court, and d denotes the Cartesian distance between 2 points.

$$R_k=\{x\epsilon X|d(x,P_k)\leq d(x,P_j)\text{ for all }j\neq k\}$$

[0144] Now there exists the 2 components for determining the probability that each player gets the rebound given their location, specifically, the shot's location, and the location of all the other players on the court. One may determine this value by assuming that if a ball is rebounded, it will always be rebounded by the closest available player. Therefore, by integrating the probability of a rebound over each location in the player's Voronoi cell, we determine their rebound probability:

$$\int_R P(r_x,r_y|s_x,s_y)dxdy$$

[0145] The preceding section describes a method for determining the players rebounding probability, assuming that the players are stationary. However, players often move in order to get into better positions for the rebound, especially when they begin in poor positions. One may account for this phenomena. Let the player's raw rebound probability be denoted r_p and let d be an indicator variable denoting whether the player is on defense.

[0146] On may then attempt to estimate the player's probability of getting a rebound, which we express in the following manner:

$$P(r|rp,d)$$

[0147] One does this by performing two linear regressions, one for the offensive side of the ball and one for the defensive. One may attempt to estimate p(r|r_p, d) in the following manner:

$$P(r|r_p,d=0)=A_o*r_p+B_o$$

$$P(r|r_p,d=1)=A_d*r_p+B_d$$

[0148] This results in four quantities to estimate. One may do this by performing an ordinary least squares regression for offensive and defensive players over all rebounds in the test set. One may use 1 as a target variable when the player rebounds the ball, and 0 when he does not. This regression is performed for offense to determine A_o and B_o and for defense to determine A_d and B_d. One can then use the values to determine the final probability of each player getting the rebound given the shots location and the other players on the court.

[0149] Novel shooting metrics can also be created using this system. One is able to determine the probability of a shot being made given various features of the shot s, denoted as F. Formally each shot can be characterized by a feature vector of the following form:

$$[\mathrm{dist}(\mathrm{hoop},\mathrm{shooter}),\mathrm{dist}(\mathrm{shooter},\mathrm{defender}_0),|\mathrm{angle}(\mathrm{hoop},\mathrm{shooter},\mathrm{defender}_0)|,|\mathrm{angle}(\mathrm{shooter},\mathrm{hoop},\mathrm{hoop}_{other}),I(\mathrm{shot}=\mathrm{catchAndShoot}),\mathrm{dist}(\mathrm{shooter},\mathrm{defender}_1)]$$

[0150] Here, the hoop represents the basket the shooter is shooting at, $\mathrm{defender}_0$ refers to the closest defender to the shooter, defender' refers to the second closest defender, and hoop_{other} refers to the hoop on the other end of the court. The angle function refers to the angle between three points, with the middle point serving as the vertex. I(shot=catchAndShoot) is an indicator variable, set to 1 if the shooter took no dribbles in the individual possession before shooting the shot, otherwise set to 0.

[0151] Given these features one seeks to estimate P(s=make). To do this, one may first split the shots into 2 categories, one for where dist(hoop, shooter) is less than 10, and the other for the remaining shots. Within each category one may find coefficients $\beta_0, \beta_1, \ldots, \beta_5$ for the following equation:

$$1/(1+e\hat{}(-t))$$

where

$$t=F0*\beta 0+F1*\beta 1+\ldots+F5*\beta 5$$

[0152] Here, F_0 through F_5 denote the feature values for the particular shot. One may find the coefficient values $\beta_0, \beta_1, \ldots, \beta_5$ using logistic regression on the training set of shots S. The target for the regression is 0 when the shot is missed and 1 when the shot is made. By performing two regressions, one is able to find appropriate values for the coefficients, for both shots within 10 feet, and longer shots outside 10 feet.

[0153] As depicted in FIG. **23**, three or four dimensions can be dynamically displayed on a 2-D graph scatter rank view **2302**, including the x, y, size of the icon, and changes over time. Each dimension may be selected by the user to represent a variable of the user's choice. Also, on mouse-over, related icons may highlight, e.g. mousing over one player may highlight all players on the same team.

[0154] As depicted in FIG. **40**, reports **2402** can be customized by the user, so that a team can create a report that is specifically tailored to that team's process and workflow. Another feature is that the report may visually display not only the advantages and disadvantages for each category shown, but also the size of that advantage or disadvantage, along with the value and rank of each side being compared. This visual language enables a user to quickly scan the report and understanding the most important points.

[0155] Referring to FIG. **25**, an embodiment of a quality assurance UI **2502** is provided. The QA UI **2502** presents the human operator with both an animated 2D overhead view **2510** of the play, as well as a video clip **2508** of the play. A key feature is that only the few seconds relevant to that play are shown to the operator, instead of an entire possession, which might be over 20 seconds long, or even worse, requiring the human operator to fast forward in the game tape to find the event herself. Keyboard shortcuts are used for all operations, to maximize efficiency. Referring to FIG. **26**, the operator's task is simplified to its core, so that we lighten the cognitive load as much as possible: if the operator is verifying a category of plays X, the operator has to simply choose, in an interface element **2604** of the embodiment of the QA UI **2602** whether the play shown in the view **2608** is valid (Yes or No), or (Maybe). She can also deem the play to be a (Duplicate), a (Compound) play that means it is just one type-X action in a consecutive sequence of type-X actions, or choose to (Flag) the play for supervisor review for any reason. Features of the UI **2602** include the ability to fast word, rewind, submit and the like, as reflected in the menu element **2612**. A table **2610** can allow a user to indicate validity of plays occurring at designated times.

[0156] FIG. **27** shows a method of camera pose detection, also known as "court solving." The figure shows result of automatic detection of the "paint", and use of the boundary lines to solve for the camera pose. The court lines and hoop location, given the solved camera pose, are then shown projected back onto the original image **2702**. This projection is from the first iteration of the solving process, and one can see that the projected court and the actual court do not yet align perfectly. One may use machine vision techniques to find the hoop and to find the court lines (e.g. paint boundaries), then use found lines to solve for the camera pose. Multiple techniques may be used to determine court lines, including detecting the paint area. Paint area detection can be done automatically. One method involves automatically removing the non-paint area of the court by automatically executing a series of "flood fill" type actions across the image, selecting for court-colored pixels. This leaves the paint area in the image, and it is then straightforward to find the lines/points. One may also detect all lines on the court that are visible, e.g. background or 3-poin arc. In either case, intersections provide points for camera solving. A human interface **2702** may be provided for providing points or lines to assist algorithms, to fine-tune the automatic solver. Once all inputs are provided, the camera pose solver is essentially a randomized hill climber that uses the mathematical models as a guide (since it may be under-constrained). It may use multiple random initializations. It may advance a solution if it is one of the best in that round.

When an iteration is done, it may repeat until the error is small. FIG. **46** shows the result of automatic detection of the "paint", and use of the boundary lines to solve for the camera pose. The court lines and hoop location, given the solved camera pose, are then shown projected back onto the original image. This projection is from the first iteration of the solving process, and one can see that the projected court and the actual court do not yet align perfectly.

[0157] FIG. **28** relates to camera pose detection. The second step **2802** shown in the Figure shows how the human can use this GUI to manually refine camera solutions that remain slightly off.

[0158] FIG. **29** relates to auto-rotoscoping. Rotoscoping **2902** is required in order to paint graphics around players without overlapping the players' bodies. Rotoscoping is partially automated by selecting out the parts of the image with similar color as the court. Masses of color left in the image can be detected to be human silhouettes. The patch of color can be "vectorized" by finding a small number of vectors that surround the patch, but without capturing too many pixels that might not represent a player's body.

[0159] FIGS. **30**A-**30**C relate to scripted storytelling with an asset library **3002**. To produce the graphics-augmented clips, a company may either learn heavily on a team of artists, or a company may determine how best to handle scripting based on a library of assets. For example, instead of manually tracing a player's trajectory and increasing the shot probability in each frame as the player gets closer to the ball, a scripting language allows the methods and systems described herein to specify this augmentation in a few lines of code. In another example, for rebound clips, the Voronoi partition and the associated rebound positioning percentages can be difficult to compute for every frame. A library of story element effects may list each of these current and future effects. Certain combinations of scripted story element effects may be best suited for certain types of clips. For example, a rebound and put-back will likely make use of the original shot probability, the rebound probabilities including Voronoi partitioning, and then go back to the shot probability of the player going for the rebound. This entire script can be learned as being well-associated with the event type in the video. Over time, the system can automatically infer the best, or at least retrieve an appropriate, story line to match up with a selected video clip containing certain events. This enables augmented video clips, referred to herein as DataFX clips, to be autogenerated and delivered throughout a game.

[0160] FIGS. **31**-**38** show examples of DataFX visualizations. The visualization of FIG. **31** requires court position to be solved in order to lay down grid, player "puddles". Shot arc also requires backboard/hoop solution. In FIG. **32**, Voronoi tessellation, heat map, shot and rebound arcs all require the camera pose solution. The highlight of the player uses rotoscoping. In FIG. **33**, in addition to the above, players are rotoscoped for highlighting. FIGS. **34**-**38** show additional visualizations that are based on use of the methods and systems disclosed herein.

[0161] In embodiments, DataFX (video augmented with data-driven special effects) may be provided for pre-, during, or post-game viewing, for analytic and entertainment purposes. DataFX may combine advanced data with Hollywood-style special effects. Pure numbers can be boring, while pure special effects can be silly, but combination of the two and the results can be very powerful. Example features used alone or in combination in DataFX can include use of a Voronoi overlay on court, a Grid overlay on court, a Heat map overlay on court, a Waterfall effect showing likely trajectories of the ball after a missed field goal attempt, a Spray effect on a shot, showing likely trajectories of the shot to the hoop, Circles and glows around highlighted players, Statistics and visual cues over or around players, Arrows and other markings denoting play actions, Calculation overlays on court, and effects showing each variable taken into account.

[0162] FIGS. **39**-**41** show a product referred to as "Clippertron." Provided is a method and system whereby fans can use their distributed mobile devices to individually and/or collectively control what is shown on the Jumbotron or video board (s). An embodiment enables the fan to go through mobile application dialogs in order to choose the player, shot type, and shot location to be shown on the video board. The fan can also enter in his or her own name, so that it is displayed alongside the highlight clip. Clips are shown on the Video Board in real time, or queued up for display. Variations include getting information about the fan's seat number. This could be used to show a live video feed of the fan while their selected highlight is being shown on the video board. Referred to as "FanMix" is a web-based mobile app that enables in-stadium fans to control the Jumbotron and choose highlight clips to push to the Jumbotron. An embodiment of FanMix enables fans to choose their favorite player, shot type, and shot location using a mobile device web interface. Upon pressing the submit button, a highlight showing this particular shot is sent to the Jumbotron and displayed according to placement order in a queue. Enabling this capability is that video is lined up to each shot within a fraction of a second. This allows many clips to be shown in quick succession, each showing video from the moment of release to the ball going through the hoop. In some cases, video may start from the beginning of a play, instead of when a play begins.

[0163] FIG. **41** relates to an offering referred to as "inSight." This offering allows pushing of relevant stats to fans' mobile devices **4104**. For example, if player X just made a three-point shot from the wing, this would show statistics about how often he made those types of shots **4108**, versus other types of shots, and what types of play actions he typically made these shots off of inSight does for hardcore fans what Eagle (the system described above) does for team analysts and coaches. Information, insights, and intelligence may be delivered to fans' mobile devices while they are seated in the arena. This data is not only beautiful and entertaining, but is also tuned in to the action on the court. For example, after a seemingly improbable corner three by a power forward, the fan is immediately pushed information that shows the shot's frequency, difficulty, and likelihood of being made. In embodiments, the platform features described above as "Eagle," or a subset thereof, may be provided, such as in a mobile phone form factor for the fan. An embodiment may include a storyboard stripped down, such as from a format for an 82" touch screen to a small 4" screen. Content may be pushed to device that corresponds to the real time events happening in the game. Fans may be provided access to various effects (e.g., DataFX features described herein) and to the other features of the methods and systems disclosed herein.

[0164] FIGS. **42** and **43** show touchscreen product interface elements **4202**, **4204**, **4208**, **4302** and **4304**. These are essentially many different skins and designs on the same basic functionality described throughout this disclosure. Advanced stats are shown in an intuitive large-format touch screen interface. A touchscreen may act as a storyboard for showing various visualizations, metric and effects that conform to an understanding of a game or element thereof. Embodiments include a large format touch screen for commentators to use during a broadcast. While InSight serves up content to a fan, the Storyboard enables commentators on TV to access content in a way that helps them tell the most compelling story to audiences.

[0165] Features include providing a court view, a hexagonal Frequency+Efficiency View, a "City/Matrix" View with grids of events, a Face/Histogram View, Animated intro sequences that communicate to a viewer that each head's position means that player's relative ranking, an Animated face shuttle that shows re-ranking when metric is switched, a ScatterRank View, a ranking using two variables (one on each axis), a Trends View, integration if metrics with on-demand video and the ability to r-skin or simplify for varying levels of commentator ability.

[0166] In embodiments, new metrics can be used for other activities, such as driving new types of fantasy games, e.g. point scoring in fantasy leagues could be based on new metrics.

[0167] In embodiments, DataFX can show the player how his points were scored, e.g. overlay that runs a counter over a RB's head showing yards rushed while the video shows RB going down the field. In embodiments, one can deliver, for example, video clips (possibly enhanced by DataFX effects) corresponding to plays that scored points for a fantasy user's team for that night or week.

[0168] Using an inSight-like mobile interface, a social game can be made so that much of the game play occurs in real time while the fan is watching the game.

[0169] Using Insight-like mobile device features, a social game can be managed so that game play occurs in real time while a fan is watching the game, experiencing various DataFX effects and seeing fantasy scoring-relevant metrics on screen during the game. In embodiments, the methods and systems may include a fantasy advice or drafting tool for fans, presenting rankings and other metrics that aid in player selection.

[0170] Just as Eagle enables teams to get more wins by devising better tactics and strategy, we could provide an Eagle-like service for fantasy players that gives the players a winning edge. The service/tool would enable fans to research all the possible players, and help them execute a better draft or select a better lineup for an upcoming week/game.

[0171] DataFX can also be used for instant replays with DataFX optimized so that it can produce "instant replays" with DataFX overlays. This relies on a completely automated solution for court detection, camera pose solving, player tracking, and player roto-scoping.

[0172] Interactive DataFX may also be adapted for display on a second screen, such as a tablet, while a user watches a main screen. Real time or instant reply viewing and interaction may be used to enable such effects. On a second screen-type viewing experience, the fan could interactively toggle on and off various elements of DataFX. This enables the fan to customize the experience, and to explore many different metrics. Rather than only DataFX-enabled replays, the system could be further optimized so that DataFX is overlaid in true real time, enabling the user to toggle between a live video feed, and a live video feed that is overlaid with DataFX. The user would then also be able to choose the type of DataFX to overlay, or which player(s) to overlay it on.

[0173] A touch screen UI may be established for interaction with DataFX.

[0174] Many of the above embodiments may be used for basketball, as well as for other sports and for other items that are captured in video, such as TV shows, movies, or live video (e.g., news feeds). For sports, we use the player tracking data layer to enable the computer to "understand" every second of every game. This enables the computer to deliver content that is extracting from portions of the game, and to augment that content with relevant story-telling elements. The computer thus delivers personalized interactive augmented experiences to the end user.

[0175] For non-sports domains, such as TV shows or movies, there is no player tracking data layer that assists the computer in understanding the event. Rather, in this case, the computer must derive, in some other way, an understanding of each scene in a TV show or movie. For example, the computer might use speech recognition to extract the dialogue throughout a show. Or it could use computer vision to recognize objects in each scene, such as robots in the Transformer movie. Or is could use a combination of these inputs and others to recognize things like explosions. The sound track could also provide clues.

[0176] The resulting system would use this understand to deliver the same kind of personalized interactive augmented experience as we have described for the sports domain. For example, a user could request to see the Transformer movie series, but only a compilation of the scenes where there are robots fighting and no human dialogue. This enables "short form binge watching", where users can watch content created by chopping up and re-combining bits of content from original video. The original video could be sporting events, other events TV shows, movies, and other sources. Users can thus gorge on video compilations that target their individual preferences. This also enables a summary form of watching, suitable for catching up with current events or currently trending video, without having to watch entire episodes or movies.

[0177] The methods and systems disclosed herein may also include one or more of the following features and capabilities: spatiotemporal pattern recognition (including active learning of complex patterns and learning of actions such as P&R, postups, play calls); hybrid methods for producing high quality labels, combining automated candidate generation from XY data, and manual refinement; indexing of video by automated recognition of game clock; presentation of aligned optical and video; new markings using combined display, both manual and automated (via pose detection etc); metrics: shot quality, rebounding, defense and the like; visualizations such as Voronoi, heatmap distribution, etc.; embodiment on various devices; video enhancement with metrics & visualizations; interactive display using both animations and video; gesture and touch interactions for sports coaching and commentator displays; and cleaning of XY data using HMM, PBP, video, hybrid validation.

[0178] Further details as to data cleaning **204** are provided herein. Raw input XYZ is frequently noisy, missing, or wrong. XYZ data is also delivered with attached basic events such as possession, pass, dribble, shot. These are frequently incorrect. This is important because event identification further down the process (Spatiotemporal Pattern Recognition) sometimes depends on the correctness of these basic events. As noted above, for example, if two players' XY positions are switched, then "over" vs. "under" defense would be incorrectly switches, since the players' relative positioning is used as a critical feature for the classification. Also, PBP data sources are occasionally incorrect. First, one may use validation algorithms to detect all events, including the basic events such as possession, pass, dribble, shot, and rebound that are provided with the XYZ data. Possession/Non-possession may use a Hidden Markov Model to best fit the data to these states. Shots and rebounds may use the possession model outputs, combined with 1) projected destination of the ball, and 2) PBP information. Dribbles may be identified using a trained ML algorithm, and also using the output of the possession model.

[0179] Specifically, once possessions are determined, dribbles may be identified with a hidden Markov model. The hidden Markov model consists of three states:

[0180] 1. Holding the ball while the player is still able to dribble.

[0181] 2. Dribbling the ball.

[0182] 3. Holding the ball after the player has already dribbled.

[0183] A player starts in State 1 when he gains possession of the ball. At all times players are allowed to transition to either their current state, or the state with a number one higher than their current state, if such a state exists.

[0184] The players likelihood of staying in their current state or transitioning to another state may be determined by the transition probabilities of the model as well as the observations. The transition probabilities may be learned empirically from the training data. The observations of the model consist of the player's speed, which is placed into two categories, one for fast movement, and one for slow movement, as well as the ball's height, which is placed into categories for low and high height. The cross product of these two observations represents the observation space for the model. Similar to the transition probabilities, the observation probabilities given a particular state may be learned empirically from the training data. Once these probabilities are known, the model is fully characterized, and may be used to classify when the player is dribbling on unknown data.

[0185] Once it is known that the player is dribbling, it remains to be determined when the actual dribbles occur. This may be done with a Support Vector Machine that uses domain specific information about the ball and player, such as the height of the ball as a feature to determine whether at that instant the player is dribbling. A filtering pass may also be applied to the resulting dribbles to ensure that they are sensibly separated, so that for instance, two dribbles do not occur within 0.04 seconds of each other.

[0186] Returning to the discussion of the algorithms, these algorithms decrease the basic event labeling error rate by a significant factor, such as about 50%. Second, the system has a library of anomaly detection algorithms to identify potential problems in the data. These include temporal discontinuities (intervals of missing data are flagged); spatial discontinuities (objects traveling is a non-smooth motion, "jumping"); interpolation detection (data that is too smooth, indicating that post-processing was done by the data supplier to interpolate between known data points in order to fill in missing data). This problem data is flagged for human review, so that events detected during these periods are subject to further scrutiny.

[0187] Spatio-player tracking may be undertaken in at least two types, as well as in a hybrid combined type. For tracking with broadcast video, the broadcast video is obtained from multiple broadcast video feeds. Typically, this will include a standard "from the stands view" from the center stands midway-up, a backboard view, a stands view from a lower angle from each corner, and potentially other views. Optionally, PTZ (pan tilt zoom) sensor information from each camera is also returned. An alternative is a Special Camera Setup method. Instead of broadcast feeds, this uses feeds from cameras that are mounted specifically for the purposes of player tracking. The cameras are typically fixed in terms of their location, pan, tilt, zoom. These cameras are typically mounted at high overhead angles; in the current instantiation, typically along the overhead catwalks above the court. A Hybrid/Combined System may be used. This system would use both broadcast feeds and feeds from the purpose-mounted cameras. By combining both input systems, accuracy is improved. Also, the outputs are ready to be passed on to the DataFX pipeline for immediate processing, since the DataFX will be painting graphics on top of the already-processed broadcast feeds. Where broadcast video is used, the camera pose must be solved in each frame, since the PTZ may change from frame to frame. Optionally, cameras that have PTZ sensors may return this info to the system, and the PTZ inputs are used as initial solutions for the camera pose solver. If this initialization is deemed correct by the algorithm, it will be used as the final result; otherwise refinement will occur until the system receives a useable solution. As described above, players may be identified by patches of color on the court. The corresponding positions are known since the camera pose is known, and we can perform the proper projections between 3D space and pixel space.

[0188] Where purpose mounted cameras are used, multiple levels of resolution may be involved. Certain areas of the court or field require more sensitivity, e.g. on some courts, the color of the "paint" area makes it difficult to track players when they are in the paint. Extra cameras with higher dynamic range and higher zoom are focused on these areas. The extra sensitivity enables the computer vision techniques to train separate algorithms for different portions of the court, tuning each algorithm to its type of inputs and the difficult of that task.

[0189] In a combination system, by combining the fixed and broadcast video feeds, the outputs of a player tracking system can feed directly into the DataFX production, enabling near-real-time DataFX. Broadcast video may also produce high-definition samples that can be used to increase accuracy.

[0190] Methods and systems disclosed herein may include tracklet stitching. Optical player tracking results in short to medium length tracklets, which typically end when the system loses track of a player or the player collides (or passes close to) with another player. Using team identification and other attributes, algorithms can stitch these tracklets together.

[0191] Where a human being is in the loop, systems may be designed for rapid interaction and for disambiguation and error handling. Such a system is designed to optimize human interaction with the system. Novel interfaces may be provided to specify the motion of multiple moving actors simultaneously, without having to match up movements frame by frame.

[0192] In embodiments, custom clipping is sued for content creation, such as involving OCR. Machine vision techniques may be used to automatically locate the "score bug" and determine the location of the game clock, score, and quarter information. This information is read and recognized by OCR algorithms. Post-processing algorithms using various filtering techniques are used to resolve issues in the OCR. Kalman filtering/HMMs may be used to detect errors and correct them. Probabilistic outputs (which measure degree of confidence) assist in this error detection/correction.

[0193] Sometimes, a score is non-existent or cannot be detected automatically (e.g. sometimes during PIP or split screens). In these cases, remaining inconsistencies or missing data is resolved with the assistance of human input. Human input is designed to be sparse so that labelers do not have to provide input at every frame. Interpolation and other heuristics are used to fill in the gaps. Consistency checking is done to verify game clock.

[0194] For alignment **2112**, as discussed in connection with FIG. **21**, another advance is to use machine vision techniques to verify some of the events. For example: video of a made shot will typically show the score being increased, or will show a ball going through a hoop. Either kind of automatic observation serves to help the alignment process result in the correct video frames being shown to the end user.

[0195] In accordance with an exemplary and non-limiting embodiment, augmented or enhanced video with extracted semantics-based experience is provided based, at least in part, on 3D position/motion data. [CV1A] In accordance with other exemplary embodiments there is provided embeddable app content for augmented video with an extracted semantics-based experience. [CV1B] In yet another exemplary embodiment, there is provided the ability to automatically detect the court/field, and relative positioning of the camera, in (near) real time using computer vision techniques. This may be combined with automatic rotoscoping of the players in order to produce dynamic augmented video content.

[0196] In accordance with an exemplary and non-limiting embodiment, there is described a method for the extraction of events and situations corresponding to semantically relevant concepts. In yet other embodiments, semantic events may be translated and catalogued into data and patterns.

[0197] In accordance with an exemplary and non-limiting embodiment, there is provided a touch screen or other gesture-based interface experience based, at least in part, on extracted semantic events.

[0198] In accordance with an exemplary and non-limiting embodiment, there is described a second screen interface unique to extracted semantic events and user selected augmentations. In yet other embodiments, the second screen may display real-time, or near real time, contextualized content.

[0199] In accordance with an exemplary and non-limiting embodiment, there is described a method for "painting" translated semantic data onto an interface.

[0200] In accordance with an exemplary and non-limiting embodiment, there is described spatio-temporal pattern recognition based, at least in part, on optical XYZ alignment for semantic events. In yet other embodiments, there is described the verification and refinement of spatiotemporal semantic pattern recognition based, at least in part, on hybrid validation from multiple sources.

[0201] In accordance with an exemplary and non-limiting embodiment, there is described human identified video alignment labels and markings for semantic events. In yet other embodiments, there is described machine learning algorithms for spatiotemporal pattern recognition based, at least in part, on human identified video alignment labels for semantic events.

[0202] In accordance with an exemplary and non-limiting embodiment, there is described automatic game clock indexing of video from sporting events using machine vision techniques, and cross-referencing this index with a semantic layer that indexes game events. The product is the ability to query for highly detailed events, and return corresponding video in near real-time.

[0203] In accordance with an exemplary and non-limiting embodiment, there is described unique metrics based, at least in part, on spatiotemporal patterns including, for example, shot quality, rebound ratings (positioning, attack, conversion) and the like.

[0204] In accordance with an exemplary and non-limiting embodiment, there is described player tracking using broadcast video feeds.

[0205] In accordance with an exemplary and non-limiting embodiment, there is described player tracking using multicamera system.

[0206] In accordance with an exemplary and non-limiting embodiment, there is described video cut-up based on extracted semantics. A video cut-up is a remix made up of small clips of video that are related to each other in some meaningful way. The semantic layer enables real-time discovery and delivery of custom cut-ups. The semantic layer may be produced in one of two ways: (1) Video combined with data produces semantic layer, or (2) video directly to a semantic layer. Extraction may be through ML or human tagging. In some exemplary embodiments, video cut-up may be based, at least in part, on extracted semantics, controlled by users in a stadium and displayed on a jumbotron. In other embodiments, video cut-up may be based, at least in part, on extracted semantics, controlled by users at home and displayed on broadcast TV. In yet other embodiments, video cut-up may be based, at least n part, on extracted semantics, controlled by individual users and displayed on web, tablet, or mobile for that user. In yet other embodiments, video cut-up may be based, at least in part, on extracted semantics, created by an individual user, and shared with others. Sharing could be through inter-tablet/inter-device communication, or via mobile sharing sites.

[0207] In accordance with an exemplary and non-limiting embodiment, X, Y and Z data may be collected for purposes of inferring player actions that have a vertical component.

[0208] The methods and systems described herein may be deployed in part or in whole through a machine that executes computer software, program codes, and/or instructions on a processor. The processor may be part of a server, client, network infrastructure, mobile computing platform, stationary computing platform, or other computing platform. A processor may be any kind of computational or processing device capable of executing program instructions, codes, binary instructions and the like. The processor may be or include a signal processor, digital processor, embedded processor, microprocessor or any variant such as a co-processor (math co-processor, graphic co-processor, communication co-processor and the like) and the like that may directly or indirectly facilitate execution of program code or program instructions stored thereon. In addition, the processor may enable execution of multiple programs, threads, and codes. The threads may be executed simultaneously to enhance the performance of the processor and to facilitate simultaneous operations of the application. By way of implementation, methods, program codes, program instructions and the like described herein may be implemented in one or more thread. The thread may spawn other threads that may have assigned priorities associated with them; the processor may execute these threads based on priority or any other order based on instructions provided in the program code. The processor may include memory that stores methods, codes, instructions and programs as described herein and elsewhere. The processor may access a storage medium through an interface that may store methods, codes, and instructions as described herein and elsewhere. The storage medium associated with the processor for storing methods, programs, codes, program instructions or other type of instructions capable of being executed by the computing or processing device may include but may not be limited to one or more of a CD-ROM, DVD, memory, hard disk, flash drive, RAM, ROM, cache and the like.

[0209] A processor may include one or more cores that may enhance speed and performance of a multiprocessor. In embodiments, the process may be a dual core processor, quad core processors, other chip-level multiprocessor and the like that combine two or more independent cores (called a die).

[0210] The methods and systems described herein may be deployed in part or in whole through a machine that executes computer software on a server, client, firewall, gateway, hub, router, or other such computer and/or networking hardware. The software program may be associated with a server that may include a file server, print server, domain server, Internet server, intranet server and other variants such as secondary server, host server, distributed server and the like. The server may include one or more of memories, processors, computer readable media, storage media, ports (physical and virtual), communication devices, and interfaces capable of accessing other servers, clients, machines, and devices through a wired or a wireless medium, and the like. The methods, programs or codes as described herein and elsewhere may be executed by the server. In addition, other devices required for execution of methods as described in this application may be considered as a part of the infrastructure associated with the server.

[0211] The server may provide an interface to other devices including, without limitation, clients, other servers, printers, database servers, print servers, file servers, communication servers, distributed servers and the like. Additionally, this coupling and/or connection may facilitate remote execution of program across the network. The networking of some or all of these devices may facilitate parallel processing of a program or method at one or more location without deviating from the scope. In addition, any of the devices attached to the server through an interface may include at least one storage medium capable of storing methods, programs, code and/or instructions. A central repository may provide program instructions to be executed on different devices. In this implementation, the remote repository may act as a storage medium for program code, instructions, and programs.

[0212] The software program may be associated with a client that may include a file client, print client, domain client, Internet client, intranet client and other variants such as secondary client, host client, distributed client and the like. The client may include one or more of memories, processors, computer readable media, storage media, ports (physical and virtual), communication devices, and interfaces capable of accessing other clients, servers, machines, and devices through a wired or a wireless medium, and the like. The methods, programs or codes as described herein and elsewhere may be executed by the client. In addition, other devices required for execution of methods as described in this application may be considered as a part of the infrastructure associated with the client.

[0213] The client may provide an interface to other devices including, without limitation, servers, other clients, printers, database servers, print servers, file servers, communication servers, distributed servers and the like. Additionally, this coupling and/or connection may facilitate remote execution of program across the network. The networking of some or all of these devices may facilitate parallel processing of a program or method at one or more location without deviating from the scope. In addition, any of the devices attached to the client through an interface may include at least one storage medium capable of storing methods, programs, applications, code and/or instructions. A central repository may provide program instructions to be executed on different devices. In this implementation, the remote repository may act as a storage medium for program code, instructions, and programs.

[0214] The methods and systems described herein may be deployed in part or in whole through network infrastructures. The network infrastructure may include elements such as computing devices, servers, routers, hubs, firewalls, clients, personal computers, communication devices, routing devices and other active and passive devices, modules and/or components as known in the art. The computing and/or non-computing device(s) associated with the network infrastructure may include, apart from other components, a storage medium such as flash memory, buffer, stack, RAM, ROM and the like. The processes, methods, program codes, instructions described herein and elsewhere may be executed by one or more of the network infrastructural elements.

[0215] The methods, program codes, and instructions described herein and elsewhere may be implemented on a cellular network having multiple cells. The cellular network may either be frequency division multiple access (FDMA) network or code division multiple access (CDMA) network. The cellular network may include mobile devices, cell sites, base stations, repeaters, antennas, towers, and the like. The cell network may be a GSM, GPRS, 3G, EVDO, mesh, or other networks types.

[0216] The methods, programs codes, and instructions described herein and elsewhere may be implemented on or through mobile devices. The mobile devices may include navigation devices, cell phones, mobile phones, mobile personal digital assistants, laptops, palmtops, netbooks, pagers, electronic books readers, music players and the like. These devices may include, apart from other components, a storage medium such as a flash memory, buffer, RAM, ROM and one or more computing devices. The computing devices associated with mobile devices may be enabled to execute program codes, methods, and instructions stored thereon. Alternatively, the mobile devices may be configured to execute instructions in collaboration with other devices. The mobile devices may communicate with base stations interfaced with servers and configured to execute program codes. The mobile devices may communicate on a peer to peer network, mesh network, or other communications network. The program code may be stored on the storage medium associated with the server and executed by a computing device embedded within the server. The base station may include a computing device and a storage medium. The storage device may store program codes and instructions executed by the computing devices associated with the base station.

[0217] The computer software, program codes, and/or instructions may be stored and/or accessed on machine readable media that may include: computer components, devices, and recording media that retain digital data used for computing for some interval of time; semiconductor storage known as random access memory (RAM); mass storage typically for more permanent storage, such as optical discs, forms of magnetic storage like hard disks, tapes, drums, cards and other types; processor registers, cache memory, volatile memory, non-volatile memory; optical storage such as CD, DVD; removable media such as flash memory (e.g. USB sticks or keys), floppy disks, magnetic tape, paper tape, punch cards, standalone RAM disks, Zip drives, removable mass storage, off-line, and the like; other computer memory such as dynamic memory, static memory, read/write storage, mutable storage, read only, random access, sequential access, location addressable, file addressable, content addressable, network attached storage, storage area network, bar codes, magnetic ink, and the like.

[0218] The methods and systems described herein may transform physical and/or or intangible items from one state to another. The methods and systems described herein may also transform data representing physical and/or intangible items from one state to another.

[0219] The elements described and depicted herein, including in flow charts and block diagrams throughout the figures, imply logical boundaries between the elements. However, according to software or hardware engineering practices, the depicted elements and the functions thereof may be implemented on machines through computer executable media having a processor capable of executing program instructions stored thereon as a monolithic software structure, as standalone software modules, or as modules that employ external routines, code, services, and so forth, or any combination of these, and all such implementations may be within the scope of the present disclosure. Examples of such machines may include, but may not be limited to, personal digital assistants, laptops, personal computers, mobile phones, other handheld computing devices, medical equipment, wired or wireless communication devices, transducers, chips, calculators, satellites, tablet PCs, electronic books, gadgets, electronic devices, devices having artificial intelligence, computing devices, networking equipment, servers, routers and the like. Furthermore, the elements depicted in the flow chart and block diagrams or any other logical component may be implemented on a machine capable of executing program instructions. Thus, while the foregoing drawings and descriptions set forth functional aspects of the disclosed systems, no particular arrangement of software for implementing these functional aspects should be inferred from these descriptions unless explicitly stated or otherwise clear from the context. Similarly, it may be appreciated that the various steps identified and described above may be varied, and that the order of steps may be adapted to particular applications of the techniques disclosed herein. All such variations and modifications are intended to fall within the scope of this disclosure. As such, the depiction and/or description of an order for various steps should not be understood to require a particular order of execution for those steps, unless required by a particular application, or explicitly stated or otherwise clear from the context.

[0220] The methods and/or processes described above, and steps thereof, may be realized in hardware, software or any combination of hardware and software suitable for a particular application. The hardware may include a general purpose computer and/or dedicated computing device or specific computing device or particular aspect or component of a specific computing device. The processes may be realized in one or more microprocessors, microcontrollers, embedded microcontrollers, programmable digital signal processors or other programmable device, along with internal and/or external memory. The processes may also, or instead, be embodied in an application specific integrated circuit, a programmable gate array, programmable array logic, or any other device or combination of devices that may be configured to process electronic signals. It may further be appreciated that one or more of the processes may be realized as a computer executable code capable of being executed on a machine readable medium.

[0221] The computer executable code may be created using a structured programming language such as C, an object oriented programming language such as C++, or any other high-level or low-level programming language (including assembly languages, hardware description languages, and database programming languages and technologies) that may be stored, compiled or interpreted to run on one of the above devices, as well as heterogeneous combinations of processors, processor architectures, or combinations of different hardware and software, or any other machine capable of executing program instructions.

[0222] Thus, in one aspect, each method described above and combinations thereof may be embodied in computer executable code that, when executing on one or more computing devices, performs the steps thereof. In another aspect, the methods may be embodied in systems that perform the steps thereof, and may be distributed across devices in a number of ways, or all of the functionality may be integrated into a dedicated, standalone device or other hardware. In another aspect, the means for performing the steps associated with the processes described above may include any of the hardware and/or software described above. All such permutations and combinations are intended to fall within the scope of the present disclosure.

[0223] While the methods and systems described herein have been disclosed in connection with certain preferred embodiments shown and described in detail, various modifications and improvements thereon may become readily apparent to those skilled in the art. Accordingly, the spirit and scope of the methods and systems described herein is not to be limited by the foregoing examples, but is to be understood in the broadest sense allowable by law.

[0224] All documents referenced herein are hereby incorporated by reference in their entirety.

What is claimed is:

1. A method, comprising:

Taking a video feed of an event;

Using machine learning to develop an understanding of the event based on the video feed;

Automatically, under computer control, aligning the video feed with the understanding; and

Producing a transformed video feed that includes at least one highlight that is extracted from the machine learning of the event.

2. A method of claim **1**, wherein the event is a sporting event.

3. A method of claim **1**, further comprising:

performing automatic recognition of a camera position based, at least in part, on a scene in the video feed; and

augmenting the video feed with at least one of additional imagery and graphics rendered within a 3D space of the scene.

4. A method of claim **1**, wherein the transformed video feed creates a highlight video feed of video for a defined set of players.

5. A method of claim **1**, further comprising delivering the transformed video feed to at least one of an inbox, a mobile device, a table, an application, a scoreboard, a Jumbotron board, a video board, and a television network.

6. A method of claim **1**, wherein developing an understanding comprises applying machine learning to determine at least one spatiotemporal pattern of the event.

7. A method of claim **6**, further comprising using a human validation process to at least one of validate and teach the machine learning of the spatiotemporal pattern.

8. A method of claim **6**, further comprising:

taking data relating to a known configuration of a venue where the event takes place; and

automatically, under computer control, recognizing a camera pose based on the video feed and the known configuration.

9. A method of claim **8**, wherein the venue is a sporting event venue.

10. A method of claim **6**, further comprising presenting at least one metric in the augmented feed based on the determined spatiotemporal pattern.

11. A method of claim **10**, further comprising enabling a user to interact with at least one of the video feed and a frame of the video feed in a 3D user interface.

12. A method, comprising:

taking a data set associated with a video feed of a live event;

taking spatiotemporal features of the live event;

applying machine learning to determine at least one spatiotemporal pattern of the event; and

calculating a metric based on the determined pattern.

13. A method of claim **12**, wherein the metric is at least one of a shot quality metric (SEFG), an EFG+ metric, a rebound positioning metric, a rebounding attack metric, a rebounding conversion metric, a event-count per playing time metric, and an efficiency per event-count metric.

14. A method of claim **12**, further comprising:

providing an interactive, graphical user interface for exploration of data extracted by the machine learning, wherein the graphical user interface enables exploration and analysis of events.

15. A method of claim **14**, wherein the graphical user interface is at least one of a mobile device interface, a laptop interface, a tablet interface, a large-format touchscreen, and a personal computer interface.

16. A method of claim **14**, wherein the exploration enables at least one of a touch interaction, a gesture interaction, a voice interaction and a motion-based interaction.

17. A method, comprising:

taking a data set associated with a video feed of a live event;

automatically, under computer control, recognizing a camera pose for the video;

tracking at least one of a player and an object in the video feed; and

placing the tracked items in a spatial location corresponding to spatial coordinates.

18. A method of claim **17**, further comprising delivering contextualized information during the event to a viewer.

19. A method of claim **18**, wherein the contextualized information includes at least one of a statistic, a replay, a visualization, a highlight, a compilation of highlights, and a replay.

20. A method of claim **18**, further comprising providing a touch screen interaction with a visual representation of at least one item of the contextualized information.

* * * * *