UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/696,647	04/04/2007	John Harper	P3352USC1 (119-0033USC)	4857

61947 7590 03/04/2013
(Apple) WONG, CABELLO, LUTSCH, RUTHERFORD
& BRUCCULERI, LLP
20333 Tomball Parkway
SUITE 600
HOUSTON, TX 77070

EXAMINER
RICHER, JONI

ART UNIT	PAPER NUMBER
2677	

NOTIFICATION DATE	DELIVERY MODE
03/04/2013	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

WCPATENTS@COUNSELIP.COM
cmiles@counselip.com
sconroy@counselip.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte JOHN HARPER, RALPH BRUNNER,
PETER GRAFFAGNINO, and MARK ZIMMER

Appeal 2010-003412
Application 11/696,647
Technology Center 2600

Before CARLA M. KRIVAK, THU A. DANG, and
ELENI MANTIS MERCADER, *Administrative Patent Judges*.

MANTIS MERCADER, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from a final rejection of claims 1-21. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

THE INVENTION

Appellants claimed invention is directed to producing images including an application program interface. Abstract.

Independent claim 1, reproduced below, is representative of the subject matter on appeal.

1. A system for command-based interface to graphics resources comprising:

a first process using a microprocessor resource to request a graphics service in a request at runtime, said request comprising an indication of a function and an association between said function and a graphics object; and

a second process receiving said request and, based upon said request, evaluating a plurality of specifically interrelated graphics program modules by running an optimization routine, said optimization routine examining and optimizing said graphics program modules and their inter-relationships to produce an optimized and compiled version of said graphics program modules, said graphics program modules being executable on a graphics processor resource to implement filter functions on graphics objects,

wherein said optimized and compiled version of said graphics program modules is executable on said graphics processor resource to generate a result graphics object at runtime.

REFERENCES and REJECTIONS

1. The Examiner rejected claims 1-3, 6-9, and 12 under 35 U.S.C. § 103(a) as being unpatentable over Grantham (US 6,215,495 B1; issued Apr. 10, 2001) and McCrossin (US 6,660,840 B1; issued Jul. 29, 2003).

2. The Examiner rejected claims 4 and 10 under 35 U.S.C. § 103(a) as being unpatentable over Grantham, McCrossin, and further in view of Kilgard (US 6,982,718 B2; issued Jan. 3, 2006).
3. The Examiner rejected claims 5, 11, 13-16, and 18-21 under 35 U.S.C. § 103(a) as being unpatentable over Grantham, McCrossin, and further in view of Boudier (US 6.995,765 B2; issued Feb. 7, 2006).
4. The Examiner rejected claim 17 under 35 U.S.C. § 103(a) as being unpatentable over Grantham, McCrossin, Boudier, and further in view of Parikh (US 6,424,348 B2; issued Jul. 23, 2002).

ISSUES

The issues are whether the Examiner erred in finding that:

1. Grantham teaches a first process to "request a graphics service" with the request "comprising an indication of a function and an association between said function and a graphics object" as recited in claim 1;
2. Boudier teaches (a) determining whether an output product of a first of said graphics program modules will comprise data that is similar to an input product of a second of said graphics program modules, (b) examining each program line in said second graphics program module to determine if it negates the possibility of combining said first and second graphics program modules, (c) identifying references to said input product in the program code of said second graphics program module, and (d) altering or replacing

said identified references to said input product, as recited in claim 13.

ANALYSIS

Claims 1-3, 6-9, and 12

Appellants argue that Grantham does not teach a first process to "request a graphics service" with the request "comprising an indication of a function and an association between said function and a graphics object" as recited in the claim 1.

We do not agree with Appellants' arguments. Grantham teaches that the VRML file that is requested instructs API (Application Programming Interface) 112 to make a number of function calls, wherein the API is structured as collection of class hierarchies (col. 4, l. 63-col. 5, l. 23), including Graphics State classes that includes Context, Appearance, Material, Texture, and TexTransform classes (col. 28, ll. 14-16). The context class maintains the graphics state for a particular graphics context (col. 3, ll. 2-4). The other Graphics State classes define how the resulting image is to be created (col. 8, ll. 38-46). Thus, the Graphics State classes indicate parameters associated with the creation of the resulting image. The DrawAction class is used to draw a scene (col. 7, ll. 31-33).

We agree with the Examiner that since the first process requests this VRML file, and the request for this VRML file results in a request for a graphics service, then the first process is considered to request a graphics service with the request comprising an indication of a function (i.e., creation of context, creation of result image) (Ans. 13). The Examiner finds (Ans. 13) and we agree that Grantham teaches a processor 108 of computer system 116 initiating a request (col. 4, ll. 63-67). Therefore, a microprocessor

resource (108) is used to request the graphics service, as it is recited in the claims. Further, we find no error in the Examiner's application of McCrossin to teach that the request comprises an association between said function (filter) and a graphics object (col. 6, l. 46-col. 7, 1.9).

Appellants further argue that Grantham does not teach a second process to "produce an optimized and compiled version of said graphics program modules" as recited in claims 1 and 7.

The Examiner finds, and we agree, that Grantham teaches a scene graph rendered by a graphics subsystem 109 (col. 5, ll. 9-11). Grantham discloses "portions of a scene graph are called subgraphs" (col. 4, ll. 24-26). Thus, the scene graph has defined functions that are performed by engines at a computer (graphics subsystem 109). Grantham teaches that the VRML file that is requested instructs API 112, and API is structured as a collection of class hierarchies (col. 4, l. 63-col. 5, l. 23), including CompileAction class 403 (col. 7, ll. 50-52). Therefore, a second process receives the request (i.e., receives the VRML file) and, based upon the request, executes the CompileAction class 403 (VRML file that is received instructs API 112, and executes the CompileAction class 403 included in API) (col. 4, l. 63-col. 5, l 23; col. 7, ll. 50-52). Grantham teaches that the "CompileAction class 403 compiles a specified subgraph into a data structure which is more efficient for traversals" (col. 7, ll. 50-52). Classes represent data objects (col. 5, l. 29). Grantham teaches that the "data objects ... allow many general and platform-specific optimizations to be advantageously exploited" (col. 4, ll. 55-58). So, the defined functions (i.e., scene graph) performed by the engines at the computer (graphics subsystem 109) (col. 5, ll. 9-11) can be examined and optimized by the CompileAction class 403 to produce an

optimized and compiled version of such calls or functions (i.e., compiles a specified subgraph into a data structure which is more efficient for traversals) (col. 7, ll. 50-52; col. 5, l. 29; col. 4, ll. 55-58). Thus, we agree with the Examiner that the second process receives the request (i.e., receives a VRML file), and based upon the request, produces an optimized and compiled version of the graphics program modules (i.e., the VRML file that is received instructs API 112, executes the CompileAction class 403 included in API, and the CompileAction class 403 produces an optimized and compiled version of said graphics program modules) (col. 4, 1. 63-col. 5, l. 23; col. 7, ll. 50-52; col. 7, ll. 50-52).

Accordingly, we affirm the Examiner's rejections of claims 1 and 7. We also affirm the Examiner's rejections of claims 2-3, 6-9, and 12 which were not separately argued.

Claims 5, 11, 13-16, and 18-21

Appellants argue that Boudier is silent as to:

> (a) determining whether an output product of a first of said graphics program modules will comprise data that is similar to an input product of a second of said graphics program modules, (b) examining each program line in said second graphics program module to determine if it negates the possibility of combining said first and second graphics program modules, (c) identifying references to said input product in the program code of said second graphics program module, and (d) altering or replacing said identified references to said input product

as recited in claim 13 (Br. 18).

The Examiner finds, and we agree, Boudier describes that the optimization gathers geometries that are in the same state to create an aggregate node (Ans. 15). In particular, the input scene graph is traversed

and, for each node, a determination is made as to whether the subtree consists of geometry (Ans. 15). The geometries of the subtree are then combined (col. 8, ll. 4-23). Boudier describes "Scene graphs are often represented in object oriented languages such as C++" (col. 1, ll. 36-37). The Examiner reasons that in order to have the input scene graph, a first of the graphics program modules must produce this input scene graph. Since the scene graph is an "input scene graph," this means that the scene graph is an input product of a second of said graphics program modules (Ans. 15). Therefore, Boudier teaches (a) determining whether an output product of a first of said graphics program modules will comprise data that is similar to an input product of a second of said graphics program modules.

Since the input scene graph is traversed and each node is examined to determine if it is possible to combine the first and second graphics program modules (col. 8, ll. 4-23), and the scene graphs are represented in C++ (col. 1, ll. 36-37), we conclude that it would have been obvious to one of ordinary skill in the art that examining each node is equivalent to examining each program line. Therefore, we find no error in the Examiner's finding that Boudier teaches (b) examining each program line in said second graphics program module to determine if it negates the possibility of combining said first and second program modules, (c) identifying references to said input product in the program code of second graphics program module (i.e., identifying geometry), and (d) altering or replacing said identified references to said input product (combining geometries of the subtree in the input scene graph) (col. 8, ll. 4-23; col. 1, ll. 36-37).

Appellants further argue that Grantham and McCrossin are not analogous art (Br. 20-21).

It has been held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned, in order to be relied upon as a basis for rejection of the claimed invention. *See In re Oetiker*, 977 F.2d 1443, 1447 (Fed. Cir. 1992). In this case, Grantham is pertinent to the problem of optimizing graphics program modules (col. 2, 1. 52-col. 3, l. 1; col. 4, ll. 55-58; col. 7, l. 50-52). McCrossin is also pertinent to the problem of optimizing graphics program modules (col. 1, ll. 18-21; col. 5, ll. 46-58). The particular problem with which the Appellants are concerned is also optimizing graphics program modules. Therefore, we conclude that Grantham and McCrossin are reasonably pertinent to the particular problem with which the Appellants are concerned, and therefore, can be relied upon as a basis for rejection of the claimed invention.

Accordingly, we affirm the Examiner's rejection of claim 13 and for the same reasons the rejections of claims 5, 11, 14-16, and 18-21 which were not separately argued.

CONCLUSIONS

The Examiner did not err in finding that:

1. Grantham teaches a first process to "request a graphics service" with the request "comprising an indication of a function and an association between said function and a graphics object" as recited in claim 1;
2. Boudier teaches (a) determining whether an output product of a first of said graphics program modules will comprise data that is similar to an input product of a second of said graphics program

modules, (b) examining each program line in said second graphics program module to determine if it negates the possibility of combining said first and second graphics program modules, (c) identifying references to said input product in the program code of said second graphics program module, and (d) altering or replacing said identified references to said input product, as recited in claim 13.

DECISION

The Examiner's decision rejecting claims 1-21 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rwk