

The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.

Paper No. 32

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ROBERT J. DEVINS and PAUL M. SCHANELY

Appeal No. 2004-1558
Application No. 09/283,386

HEARD: February 22, 2005

Before KRASS, GROSS and NAPPI, Administrative Patent Judges.

KRASS, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal from the final rejection of claims 1-3, 5-7, 9-11, and 13-25. Claims 4, 8, and 12 have been indicated by the examiner as being directed to allowable subject matter and are not on appeal before us.

Appeal No. 2004-1558
Application No. 09/283,386

The invention is directed to graphics rendering by using captured graphics hardware instructions.

Representative independent claim 1 is reproduced as follows:

1. A method for performing graphics rendering, comprising:

capturing, in a memory as an executable program, hardware-level instructions generated by a device driver in response to basic rendering functions called by a graphics application program running in a host operating system; and

defining said captured hardware-level instructions as an executable program to the host operating system wherein said hardware-level instructions are captured in the memory separate from the host operating system.

The examiner relies on the following references:

Shaw et al. (Shaw)	5,657,479	Aug. 12, 1997
Devic	5,675,773	Oct . 07, 1997

Neider, et al. "OpenGL™ Programming Guide: The Official Guide to Learning OpenGL" Release 1, Addison-Wesley Publishing Company (1993), (OGL)

Claims 1-3, 5-7, 9-11, and 13-25 stand rejected under 35 U.S.C. §103. As evidence of obviousness, the examiner cites Devic and OGL with regard to claims 1-3, 5-7, 9-11, 13-18, and 21-23, adding Shaw with regard to claims 19, 20, 24, and 25.

Reference is made to the briefs and answer for the respective positions of appellants and the examiner.

OPINION

In rejecting claims under 35 U.S.C. §103, it is incumbent upon the examiner to establish a factual basis to support the legal conclusion of obviousness. See In re Fine, 837 F.2d 1071, 1073, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). In so doing, the examiner is expected to make the factual determinations set forth in Graham v. John Deere Co., 383 U.S. 1, 17, 148 USPQ 459, 467 (1966), and to provide a reason why one having ordinary skill in the pertinent art would have been led to modify the prior art or to combine prior art references to arrive at the claimed invention. Such reason must stem from some teachings, suggestions or implications in the prior art as a whole or knowledge generally available to one having ordinary skill in the art. Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 1051, 5 USPQ2d 1434, 1438 (Fed. Cir.), cert. denied, 488 U.S. 825 (1988); Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281, 293, 227 USPQ 657, 664 (Fed. Cir. 1985), cert. denied, 475 U.S. 1017 (1986); ACS Hosp. Sys., Inc. v. Montefiore Hosp., 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed. Cir. 1984). These showings by the examiner are an essential part of complying with the burden of presenting a prima facie case of obviousness. Note In re Oetiker, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). If that burden is met, the burden then shifts to the applicant to overcome the prima facie case with

Appeal No. 2004-1558
Application No. 09/283,386

argument and/or evidence. Obviousness is then determined on the basis of the evidence as a whole and the relative persuasiveness of the arguments. See Id.; In re Hedges, 783 F.2d 1038, 1039, 228 USPQ 685, 686 (Fed. Cir. 1986); In re Piasecki, 745 F.2d 1468, 1472, 223 USPQ 785, 788 (Fed. Cir. 1984); and In re Rinehart, 531 F.2d 1048, 1052, 189 USPQ 143, 147 (CCPA 1976). Only those arguments actually made by appellants have been considered in this decision. Arguments which appellants could have made but chose not to make in the brief have not been considered and are deemed to be waived.

With regard to independent claims 1, 5, 9, and 13 the examiner applies *Devic* to the claims as follows:

The “capturing...hardware-level instructions” and the storing of these instructions in memory is said to be taught by *Devic* at column 8, line 44, through column 9, line 4.

The examiner recognizes that *Devic* “does not expressly teach defining the captured hardware-level instructions to the host operating system” (answer-page 4), but the examiner turns to OGL for a disclosure “that the start of a display list is specified by *glNewList* (*Gluint list*, *Glenum mode*); with the parameter *list* being a unique integer identifying the display list and the parameter *mode* identifying whether the display list is to be compiled for later execution or compiled for later execution as well as immediate execution at pg. 126” (answer-page 4).

The examiner concludes that it would have been obvious to combine *Devic* and

Appeal No. 2004-1558
Application No. 09/283,386

OGL “such that the captured microinstructions placed in the display list of Devic’s invention are identified with the display list’s unique identifier as taught by [OGL] so that Devic’s display list can be easily called for execution, as many times as desired” (answer-page 4).

Appellants point out that Devic describes that the hardware independent output from the high level graphics library 220 or 230 can “comprise an array of batch cells, each batch cell representing a separate graphic operation to be performed, and wherein the array of batch cells is passed to the low level hardware dependent graphics library to be processed in sequence to generate the microinstructions” (column 4, lines 4-8), and argue that a “microinstruction is defined as a computer instruction that activates the circuits necessary to perform a **single machine operation**, usually as part of the execution of a machine-language instruction.” (brief- pages 15-16) (Emphasis added).

Moreover, appellants point out that the hardware-independent “array of batch cells” of Devic contain one or more graphics primitives to be generated such as points, lines, polygons, shaded polygons, blended polygons, etc. or a graphics rendering

command to be performed, and that these graphics primitives or graphics rendering commands are passed to the low level hardware dependent graphics library 240 for subsequent generation of microinstructions (brief-page 16) (pointing to Figure 3 of Devic).

It is appellants' view that "a microinstruction is not fairly characterized as an executable program, and that an array of graphics primitives or graphics rendering commands is also submitted as not being an executable program. Thus, these graphics primitives or graphics rendering commands of Devic are not believed by appellants to fairly represent an **executable program** which is "captured in [a] memory. . .," and which is "defin[ed]. . .**as an executable program to the host operating system,**" as recited in claim 1 (see page 4 of the reply brief).

Appellants have offered a reasonable dictionary definition of "microinstruction," citing <http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=microinstruction>, and the examiner offers no reasonable alternative definition, arguing only that the hardware dependent microinstructions of Devic "read on Appellant's [sic, appellants'] executable program" (answer-page 8).

Our independent research of the term, "microinstruction" finds the following definition:

Small, simple, basic instructions that a computer is capable of executing. Sets of microinstructions might make up macroinstructions or microprograms. Microinstructions, even sets of them, might be permanently wired or built in and so are executed automatically. In variable logic computers, different combinations of microinstructions or microoperations, can be programmed. In most modern computers, microinstructions are not used and the macroinstructions are wired in.¹

Thus, since microinstructions may make up microprograms, it is not unreasonable to find that microinstructions may make up an “executable program,” as broadly recited in the instant claims, and as determined by the examiner. However, that being said, the claims call for more than a mere “executable program.” For example, claim 1 must “capture, in a memory as an executable program, hardware-level instructions generated. . .in response to basic rendering functions called by a graphics application program running in a host operating system,” and then define those captured hardware-level instructions as an executable program to the host operating system, wherein the hardware-level instructions are captured in the memory separate from the host operating system.

The examiner admits that Devic lacks a teaching of the “defining” step of the independent claims and relies on OGL to supply this teaching. We agree that OGL teaches that once a display list is created, it can be executed by calling a specified

name and the display list can be executed “many times” (OGL-page 127). However,

¹Martin H. Weik; Standard Dictionary of Computers and Information Processing; Hayden Book Company, Inc., New York; copyright 1969; 3rd Printing, 1970; page 188.

merely because the display list in OGL may be displayed “many times,” and the instant invention is concerned with storing instructions as an executable program which can be nested and executed from within another program, more efficiently rendering and re-rendering scenes, thus overcoming the prior art problem of having to regenerate the hardware instructions each time, this does not necessarily provide a reason why the skilled artisan would have taken OGL’s teaching of executing a display list “many times” and applied it to Devic in order to provide, in Devic, the “defining [of] hardware-level instructions as an executable program to the host operating system. . .,” as claimed, in differing language, in independent claims 1, 5, 9, 13, and 16.

Since we do not find that the examiner’s rationale (viz., “. . .so that Devic’s display list can be easily called for execution, as many times as desired) (answer-page 4) establishes sufficient motivation for making the proposed combination of Devic and OGL, we will not sustain the rejection of claims 1-3, 5-7, 9-11, 13-18, and 21-23 under 35 U.S.C. §103.

Turning to independent claims 19 and 20, these claims do not recite “defining. . . captured hardware-level instructions as an executable program to the host operating

system,” but they do recite that the hardware-level instructions are captured in a memory as an executable program in response to a sequence of basic rendering functions called by a graphics application program running in a host operating system, and that the sequence defines a subscene. The examiner relies on Shaw for a teaching of the claimed specification of a location within a primary scene for the subscene to be rendered, but it is the combination of Devic and OGL which the examiner presumably relies on for the capturing of instructions in response to a sequence of basic rendering functions, and wherein the sequence defines a subscene.

The trouble with the examiner’s rationale for the rejection (see pages 6-7 of the answer) is that the rationale never addresses the specific claim limitations of the hardware-level instructions generated by a device driver in response to a sequence of basic rendering functions called by a graphics application program running in a host operating system, said sequence defining a subscene.” The rationale, at pages 6-7 of the answer, never addresses a “sequence” or a “subscene” at all. Rather, the examiner appears to be restating the same rationale for rejecting claims 1-3, 5-7, 9-11, 13-18, and 21-23. Accordingly, the examiner has never addressed key limitations in the claims and, for this deficiency alone, the rejection of claims 19, 20, 24, and 25 under 35 U.S.C. §103 cannot be sustained.

We have not sustained any of the rejections under 35 U.S.C. §103 because the examiner has failed to establish the requisite motivation for combining the proposed references and/or has failed to address each and every claim limitation. But, we note that some of appellants' arguments were not persuasive. Whereas appellants argue that appellants' invention executes hardware instructions as if "for real" (principal brief-page 13), but stores the instructions as an executable program, without the need for any special binding libraries at a low level, and appellants argue that their program can be "nested and executed from within another program, to make scene rendering and re-rendering more efficient" (principal brief-page 13), we note that these argued limitations do not appear in the claims and, so, are unpersuasive. Similarly, appellants point out, at page 15 of the principal brief, that Devic's function calls are "only temporarily" stored in a memory until the last high-level hardware instruction is received, and then the high-level hardware instructions are executed and discarded, making it necessary for the host processor to regenerate the high-level hardware instructions required for drawing a repetitive shape, for example. Appellants make this point to distinguish Devic's system from the instant invention which repeatedly calls on the executable program, so that the instructions need not be regenerated as they are already nested within another program. However, we do not find any such distinguishing language in the instant claims.

In any event, the examiner's decision rejecting claims 1-3, 5-7, 9-11, and 13-25

Appeal No. 2004-1558
Application No. 09/283,386

under 35 U.S.C. §103 is reversed.

REVERSED

ERROL A. KRASS)	
Administrative Patent Judge)	
)	
)	
)	
)	BOARD OF PATENT
ANITA PELLMAN GROSS)	APPEALS
Administrative Patent Judge)	AND
)	INTERFERENCES
)	
)	
)	
ROBERT E. NAPPI)	
Administrative Patent Judge)	

EAK/vsh

Appeal No. 2004-1558
Application No. 09/283,386

CONNOLLY, BOVE, LODGE & HUTZ, LLP
SUITE 800
1990 M STREET NW
WASHINGTON DC 20036-3425