Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test

Howard Root

Follow this and additional works at: https://scholarship.law.umn.edu/mlr

Part of the Law Commons

Recommended Citation
https://scholarship.law.umn.edu/mlr/1503

This Article is brought to you for free and open access by the University of Minnesota Law School. It has been accepted for inclusion in Minnesota Law Review collection by an authorized administrator of the Scholarship Repository. For more information, please contact lenzx009@umn.edu.
Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test

Since the development of the first electro-mechanical computing device less than forty years ago, the computer industry has grown and evolved with phenomenal speed. Microcomputers and mass-marketed software for personal computers are two of the most recent innovations in the computer field. Concurrent with the increased profitability of mass-marketed software, illicit copying ("pirating") of computer programs has become inexpensive, easy, and prevalent. As a result, software manufacturers are turning to the copyright laws for protection.

Though courts now generally agree that copyright protection extends to most forms of software, they lack a workable standard for determining when infringement of copyrights in computer programs has occurred. The judicial standard used

---

1. Software is easiest to define as everything that is not hardware. Hardware is defined as the "[p]hysical equipment used in data processing." SCIENCE RESEARCH ASSOC. INC., DATA PROCESSING GLOSSARY 127 (1979). Software generally comprises three classes of subject matter: computer programs, data bases, and documentation. Bender, Licensing and Protecting Computer Software via Patents and Trade Secrets, in 1 SOFTWARE PROTECTION AND MARKETING: COMPUTER PROGRAMS AND DATA BASES; VIDEO GAMES AND MOTION PICTURES 619, 621 (1983). All forms of computer programs are software. See F. SEIPEL, COMPUTING LAW 351 app. (1977). For the purposes of this Note, the terms software and computer program are used interchangeably.


4. Piracy of personal computer programs has been estimated to drain $500 million in annual sales from software companies. Id. at 90.

to infer illicit copying in all media other than computer software, the substantial similarity test, focuses on aesthetic similarities as viewed by an ordinary observer. The few courts that have had to decide whether a program constituted an illicit copy of an original computer program, however, have either ignored the substantial similarity test altogether, or stated that they were applying the test and then ignored its standards. These courts have at least implicitly recognized that the substantial similarity test in its present form is of limited utility when applied to computer programs.

This Note argues for an "iterative" approach to the substantial similarity test in the computer software copyright field. Because recent developments in copyright protection of computer software center on mass-marketed software, this Note emphasizes such programs, designed for personal use. To aid in understanding the issues involved in copyright protection of computer programs, Part I describes the rudiments of software and computer technology. Part II then examines the scope of copyright protection for computer software and the definition of an infringing copy. Part III traces the development of the substantial similarity test from its initial application by courts to theatrical plays, literature, and other aesthetic media to its current use (and nonuse) for computer programs. Part IV analyzes the reluctance of courts to apply the substantial similarity test to copyrighted computer programs as well as the problems inherent in its application to computer programs. Finally, Part V proposes a modification of the substantial similarity test that will take account of the differences between software and the more traditional media, and also promote the policies of copyright law.

6. In this Note, an "iterative" reproduction is defined as a restatement in substantially the same form (i.e., a literal copy or translation) of a substantial portion of the copyrighted work. The term "iterative" reproduction has been applied in terms of fair use analysis to second order technology. See Note, Toward a Unified Theory of Copyright Infringement for an Advanced Technological Era, 96 Harv. L. Rev. 450 (1982). In that Note, "iterative" reproduction was defined as copying "for the purpose of simply reasserting (in substantially the same form) all or part of the contents of the original." Id. at 462. The definition used in the present Note differs slightly from that definition, although the basic idea is the same.

7. All of the copyright cases involving computer programs cited in this Note dealt with mass-marketed software. Further, software that is not sold to the general public often receives protection under trade secret law, thus making copyright protection essentially superfluous. See generally Root, Protecting Computer Software in the '80s: Practical Guidelines for Evolving Needs, 8 Rev. Comput. & Tech. L.J. 205, 228-27 (1981).
I. TECHNOLOGY OF COMPUTER SOFTWARE

Many of the problems courts encounter in resolving claims of infringement of computer software result from a misunderstanding of the technology. In order to analyze the substantial similarity test, a basic knowledge of the different versions and functions of software as well as the storage devices is necessary.

First, computer programs may be expressed in either source code or object code. Computer programmers usually will first write a program in one of the standard higher-level computer languages, such as FORTRAN or COBOL. These semi-English forms of a program are types of source code. Because computers cannot process commands expressed in

8. See Keplinger, Computer Software—Its Nature and Its Protection, 30 EMORY L.J. 483, 485 (1981). For example, one court has held that a computer program communicates with the computer, not the user, and therefore copyright protection was denied. Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812, 824-25 (E.D. Pa. 1982), rev’d, 714 F.2d 1240 (3d Cir. 1983), cert. dismissed, 104 S. Ct. 690 (1984). The computer, however, only deals in voltages, currents, and magnetic domains, with the program being a symbolic representation of the path of that current. Thus, any computer program is designed for human intelligibility, not to instruct the computer. See Brooks, Agreements with Consultants and Employees and Registering Copyrights in Computer Software, in COMPUTER LAW 1982: ACQUIRING COMPUTER GOODS AND SERVICES 150-51 (1982).


10. The Copyright Act defines a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1982). This definition implies that a computer program must be written as a statement or instruction in order to qualify for copyright protection. A better definition of a computer program, which avoids this implication, is a "specification of the sequence of computational steps" used in a computer. A. RALSTON, ENCYCLOPEDIA OF COMPUTER SCIENCE AND ENGINEERING 1219 (2d ed. 1983).

11. For a more detailed, but still basic, description of the various codes and their use in the computer, see J. O’BRIEN, COMPUTERS IN BUSINESS MANAGEMENT 246-54 (3d ed. 1982).

12. FORTRAN (FORmula TRANslation), developed in 1957, is the oldest source code language and is used primarily by scientists and mathematicians. Id. at 254.

13. COBOL (Common Business Oriented Language) is used primarily for business data processing. Id. at 257.

14. The term source code also encompasses assembly languages, which are alphanumeric forms of expression and are not as easily understandable to humans as the higher-level languages. See A. RALSTON, supra note 10, at 1383. The relationship between computer languages and codes may be better illustrated by a noncomputer analogy. Source code is the term used to describe a form of communication, as alphabetical representations describe a manner of communicating for humans. A high-level computer language, such as FORTRAN, is a type of source code in the same manner as English is a type of al-
these semi-English languages, a translating program, known as
a compiler or an assembler, must convert the source code ver-
sion of a program into object code,15 the binary form16 of a pro-
gram. Each binary numeral's location,17 or bit,18 represents
either an open or closed electrical circuit in the computer, with
each grouping of eight bits commonly known as a byte.19 A
computer program, whether expressed in source code or object
code, can be copied quickly and inexpensively.20

In addition to being expressed in different codes, computer
programs can be stored in different media, usually either floppy
disks or silicon chips.21 A "floppy" is a small magnetic disk, re-
sembling a phonograph record, that may be written on, erased,
reprogrammed, and removed.22 Programs stored on floppies
can be expressed in source code and then be compiled into ob-
ject code by the computer immediately before performing the
program's functions. Silicon chips, on the other hand, are built
into the central processor23 of the computer and consist of very

15. An assembly program, or assembler, converts source code programs
into object code either inside the processing computer, or in the case of most
microcomputers, in a larger computer before mass-marketing of the program.
See id. at 124-27. A compiler serves basically the same function as an assem-
blers except that it can call in other routines stored in a computer and incorpo-
rate them into the program, whereas an assembler merely translates a program
into object code. See A. CHANDOR, THE FACTS ON FILE DICTIONARY
OF MICROCOMPUTERS 17, 44 (1981).

16. Binary form is simply a base 2 numerical system. All of the digits of a
binary number are either 0's or 1's, corresponding in a computer to either an
open or a closed circuit. See generally J. BOYCE, MICROPROCESSOR AND

17. A "location" is the memory cell or circuit where an individual piece of
information in object code is stored within the silicon chip. P. BURTON, A DI-
CTIONARY OF MINICOMPUTING AND MICROCOMPUTING 153 (1982).

18. A bit, since it is in binary form, can only have the value of a 0 or a 1.
Id. at 27.

19. Although computer words and bytes are not synonymous, the size of a
microcomputer word generally is the same as a byte, or eight bits. See L.
HORENSTEIN, COMPUTER PERIPHERALS FOR MINICOMPUTERS, MICROPROCESSORS,
AND PERSONAL COMPUTERS 17 (1980). Amusingly, a four bit computer word,
sometimes referred to as a byte, is now commonly referred to as a "nibble." A.
CHANDOR, supra note 15, at 128; P. BURTON, supra note 17, at 40.

20. See Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775, 783
(C.D. Cal. 1983), aff'd, 721 F.2d 521 (9th Cir. 1984).

21. Computer programs can also be represented on other media, such as
coding sheets or keypunch cards. Bender, supra note 1, at 627. The most com-
mon storage devices for microcomputer programs, however, are floppy disks
and silicon chips.

22. See J. O'BRIEN, supra note 11, at 126.

23. A central processor, or CPU, contains the circuits that control the exe-
cution of a computer program. Id. at 654.
small rectangular chips that store programs in integrated circuits. A chip stores a computer program permanently and cannot be reprogrammed without extensive work. In addition, programs stored in silicon chips are always expressed in object code inside the chip and thus need not be compiled by the operating computer.

Finally, computer programs function either as operating programs or as application programs. Operating programs direct the basic processes of the computer, such as the sign-on procedure or the translation of source code programs. These programs are usually stored in silicon chips inside the computer (and thus are only fixed in object code) since the same basic processes directed by operating programs are essential to any operation of the computer. The majority of programs, however, are application programs, which perform special functions such as word processing, budget planning, or balance sheet bookkeeping. These most often are stored on floppy disks and therefore can be expressed in source code. Virtually all mass-marketed microcomputer programs, however, whether operating programs or application programs, are marketed in object code.

II. SCOPE OF COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS

Copyright protection has emerged as the primary tool to safeguard a programmer's proprietary interest in mass-marketed computer software. Because copyright laws have only

24. Id. at 82. The chips contained in a microcomputer are generally “smaller than an Oreo cookie.” J. WILLS & M. MILLER, COMPUTERS FOR EVERYBODY 78 (1981).
25. See J. O'BRIEN, supra note 11, at 160.
26. Id. at 82.
27. See generally M. DANKLE, MICROCOMPUTER OPERATING SYSTEMS 9-13 (1982).
28. The operating program’s purpose is to act as an intermediary between the user of the program and the hardware. Id. at I; see also A. RALSTON, supra note 10, at 1134.
30. Id. at 158.
31. Occasionally a program written in BASIC will be marketed in its high-level language code; however, this is an extremely rare occurrence. See D. REMER, LEGAL CARE FOR YOUR SOFTWARE: A STEP-BY-STEP GUIDE FOR COMPUTER SOFTWARE WRITERS 39 (1982). One of the reasons that programmers will market only their object code is to maintain trade secret protection for their source code. Id.
32. There are three other potential forms of proprietary protection for com-
recently been interpreted to extend to computer software, courts are still struggling in their efforts to apply copyright principles to the industry's complex technological advances. With the recent proliferation of computer copyright litigation, computer software: patent, trade secret, and contract law. All three are of little utility in the mass-marketed microcomputer field.

Patents are available only for new innovations that are novel and unobvious to the average person in the field. Most microcomputer programs do not involve new processes or inventions, but merely apply the existing technology in a new manner to a particular problem. Thus, computer programs face difficulty in satisfying the "novelty" requirement for a patent. See, e.g., Parker v. Flook, 437 U.S. 584, 591-94 (1978) (improved method of calculation is unpatentable); Gottschalk v. Benson, 409 U.S. 63, 67 (1972) (computer program that converted base ten numbers to binary numbers held unpatentable). But see Diamond v. Diehr, 450 U.S. 175, 185-87 (1981) (holding a process for molding rubber patentable even though embodied in a computer program). Copyright law, on the other hand, only requires a "modicum of creativity" in the work to be protected. See Universal Athletic Sales Co. v. Salkeld, 511 F.2d 904, 908 (3d Cir. 1974), cert. denied, 423 U.S. 863 (1975); see also Gemignani, Legal Protection for Computer Software: The View From '79, RUT. J. COMPUTER L. & TECH. 269, 293-94 (1980) (practical problems in administering patents for software further reduce their attractiveness).

Trade secret protection, although it has been the primary method of protecting nonmass-marketed software, is useless in protecting the object code of a mass-marketed computer program. Protection as a trade secret requires confidentiality in every dealing. Mass-marketing of software prevents the required confidentiality. See Davidson, Protecting Computer Software: A Comprehensive Analysis, 23 JURIMETRICS J. 339, 396 (1983).

Finally, contract law's primary usefulness also lies in protecting software that is not mass-marketed. Since it would be impractical for administrative reasons to create an individually signed contract agreement with each purchaser of mass-marketed software, package inserts are the only practical manner of attempting to create a contract. Package inserts, however, which resemble unilateral contracts, may be characterized as contracts of adhesion and held unenforceable. Greguras, Protecting Computer Software by Contract, SOFTWARE PROTECTION, Oct. 1983, at 1, 10. See also Root, supra note 7, at 228.

33. The Register of Copyrights announced on May 19, 1964 that computer software would be accepted for copyright deposit and registration. See Copyright Regulations for Computer Programs, 11 BULL. COPYRIGHT SOC'Y U.S.A. 361 (1964). The first computer program copyright was registered in 1964 to John F. Banzhaf III for a law research program. In Copyright: INFORMATION TECHNOLOGY PUBLIC POLICY '74 (1975). The first case to imply that copyright protection extended to a computer program was Harcourt, Brace & World, Inc. v. Graphic Controls Corp., 329 F. Supp. 517 (S.D.N.Y. 1971). Although the issue before the court was whether machine-readable answer sheets could be copyrighted, the court analogized to computer programs and held that the answer sheets could be copyrighted. Id. at 525. For a thorough discussion of the process of acquiring copyright protection, see Root, supra note 7, at 205.

34. Congress, as well as the judiciary, is hard pressed to keep current on computer developments. See Holder, Have Your Computer Call My Computer, STUDENT LAW., Nov. 1983, at 28, 30.

35. The number of cases dealing with copyright infringement of computer programs has grown enormously since the first such case, Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1063 (N.D. Tex. 1978), was decided in 1978. See cases cited infra notes 46-51 and accompanying text.
however, a judicial consensus is beginning to form on the existence of copyright protection for computer programs.

A. EXISTENCE OF COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS

The Copyright Act of 1976 provides protection for “original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.” Although courts recognized that computer programs written in source code fit within these requirements, they were somewhat reluctant to place other types of computer programs within the ambit of the copyright laws. By proposing explicit amendments to the Copyright Act, the National Commission on New Technological Uses of Copyrighted Works (CONTU), attempted to clarify that copyright protection extended to all forms of computer programs. Congress enacted CONTU’s


38. Although the courts never decided before the 1980 amendments whether source code programs were protected, such protection was, prior to that time, described as “beyond question.” See Maggs, Some Problems of Legal Protection of Programs for Microcomputer Control Systems, 1979 U. ILL. L.F. 453, 459-60. Even before the amendments, a programmer could register a FORTRAN program for a copyright as a literary work. See M. Nimmer, supra note 36, at § 2.04[C].


40. Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU) in 1975 to study the problems of photocopying and computer programs as they relate to copyright protection. See NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 4 (1979) [hereinafter cited as CONTU REPORT].

41. CONTU proposed two changes to the Copyright Act of 1976 relating to computer programs. First, it proposed adding a definition of “computer program” to § 101 of the Act. See supra note 10. Second, it proposed replacing the then existing § 117 with a new section allowing a rightful owner of a copy of a copyrighted program to make a copy of the program provided that the production of the copy is essential to operation of the program (i.e., reading a program off a floppy into the internal storage of a computer) or the new copy is solely
recommendations, verbatim, in 1980.42

Even after the 1980 amendments, however, three forms of software have faced difficulties in meeting the statutory requirements for copyright protection. Specifically, courts have questioned whether the object code form of a program is an engineering process and thus not a work of authorship;43 whether a program stored in a silicon chip is computer hardware and thus not a fixed literary work44 and whether an operating program is a utilitarian engineering process and thus neither a work of authorship nor a medium of expression.45

More recently, however, courts have generally interpreted the statutory guidelines and CONTU’s expression of intent surrounding the 1980 amendments46 to provide copyright protection to all forms of computer programs.47 The Third Circuit, in Apple Computer, Inc. v. Franklin Computer Corp.,48 recently

used for archival purposes (i.e., storing a back-up copy of a program). CONTU Report, supra note 40, at 12.


43. See Apple Computer, Inc. v. Franklin Computer Corp. 545 F. Supp. 812, 821-22 (E.D. Pa. 1982), rev’d, 714 F.2d 1240 (3d Cir. 1983), cert. dismissed, 104 S. Ct. 690 (1984); see also CONTU Report, supra note 40, at 28-30 (dissent of Commissioner Hersey arguing that expression must be directed to a human audience, and object code only communicates with the computer).

44. See Data Cash Systems, Inc. v. JS&A Group, Inc., 480 F. Supp. 1063, 1069 (N.D. Ill. 1979) (holding that a Read Only Memory Chip (ROM) is not a copy of a copyrighted computer program since it is not fixed), aff’d on other grounds, 628 F.2d 1038 (7th Cir. 1980).


46. Because Congress passed CONTU’s recommendations without change, the CONTU report is viewed as part of the legislative history of copyright protection of computer programs. See H.R. Rep. No. 1307, pt. 1, 96th Cong., 2d Sess. 23, reprinted in 1980 U.S. CODE CONG. & AD. NEWS 6460, 6462.


Only two courts have specifically stated that a form of a computer program is not copyrightable. Neither case, however, has been upheld on that issue by the circuit court reviewing the case. See Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812, 824-25 (E.D. Pa. 1982), rev’d, 714 F.2d 1240 (3d Cir. 1983), cert. dismissed, 104 S. Ct. 690 (1984); Data Cash Systems, Inc. v. JS&A Group, Inc., 480 F. Supp. 1063, 1066-67 n.4 (N.D. Ill. 1979) (dictum), aff’d on other grounds, 628 F.2d 1038 (7th Cir. 1980).

addressed the aforementioned barriers to copyright protection of an operating program fixed in object code in a silicon chip. The court rejected all three of the arguments, implying that a computer program in any form, written on any recording device and serving any computer purpose is copyrightable. Thus, the Third Circuit joined a growing number of courts in expanding copyright protection for computer programs.

B. INFRINGEMENT OF A COPYRIGHT IN COMPUTER PROGRAMS

Although the Copyright Act grants the owner of a copyrighted work five exclusive rights in the work, the right to reproduce copies has provided virtually the sole protection for the computer programmer’s interests. The owner of a valid copyright in a computer program seeking to prove that the exclusive right of reproduction has been infringed must prove (1) that the defendant has produced a “fixed” work within the meaning of section 101 of the Act, and (2) that the alleged copy has appropriated the “expression” of the copyrighted program.

49. Id. at 1246-54.
50. Id. at 1246-48. The Third Circuit in Apple, although not stating a broad holding, agreed with the opinion in Apple Computer, Inc. v. Formula Int’l, Inc., 562 F. Supp. 775 (C.D. Cal. 1983), aff’d, 725 F.2d 521 (9th Cir. 1984), which stated: “Any doubt as to whether the Copyright Act as presently worded protects computer programs of all types, however fixed, is removed by examining the legislative history of the 1980 Act and particularly the work of CONTU.” Id. at 781.
51. See cases cited supra note 47 and accompanying text.
52. Copyright protection grants to the owner the rights to reproduce the copyrighted work, prepare derivative works, distribute copies, publicly perform the work, and display the work publicly. See 17 U.S.C. § 106 (1982).
53. The derivative works protection is the only other doctrine that has been applied in a computer copyright context. In Midway Mfg. Co. v. Artic Int’l, Inc., 704 F.2d 1009 (7th Cir.), cert. denied, 104 S. Ct. 90 (1983), the court held that silicon chips that were to be substituted for the original chips in a video game constituted a derivative work of the copyrighted audiovisual program. Id. at 1013-14. The Midway decision is not directly applicable to computer programs, however, because the holding is confined to infringement of the audiovisual copyright of the video game, not the underlying computer program. Id. at 1012.
54. One commentator has suggested that the exclusive right to perform a copyrighted work also adds protection for the owner of a copyright in a computer program. See Prasinos, World-Wide Protection of Computer Programs by Copyright, 4 Rutgers J. Comp. & Law 42, 67 (1974).
55. The word “copies” is defined in 17 U.S.C. § 101 (1982) which provides the “fixation” requirement.
56. 17 U.S.C. § 102(b) (1982). The idea/expression dichotomy was placed in the copyright laws by the 1976 Copyright Act. The inclusion was intended “to restate . . . that the basic dichotomy between expression and idea remains unchanged.” H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670.
The requirement that an alleged copy be “fixed” is the same as the “fixed” requirement for attaining copyright protection. The alleged copy must be fixed in a material object “from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.” Since the 1980 Amendments, it has become relatively clear that a program in the internal memory of a computer is “fixed”, even though the work is merely temporarily stored and will be erased when the power is turned off.

Whereas the “fixed” requirement lends itself to an objective test, the decision whether an alleged copy appropriates the expression of a copyrighted work requires a more subjective, unprincipled determination. The copyright laws, unlike the patent laws, do not protect the idea behind the work but merely protect the manner of expressing the idea. Thus, a copyright on a blueprint of a house does not prevent another

56. Just as an author’s work must be fixed to qualify for copyright protection, 17 U.S.C. § 102(a) (1982), so must a work be fixed to qualify as a “copy” of the copyrighted program, 17 U.S.C. § 101 (1982).


58. Hubco Data Prod. v. Management Assistance, Inc., COPYRIGHT L. REP. (CCH) ¶ 25,529, at 18,105 (D. Idaho Feb. 3, 1983). Further, the duplication of a silicon chip containing a program results in the creation of a fixed copy. See Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171, 174-75 (N.D. Cal. 1981); see also 2 M. Nimmer, supra note 36, at § 8.08 (arguing that the 1980 amendments, by including works that may be perceived “with the aid of a machine” in the definition of copy, clarified that unauthorized input of a computer program into any storage device constitutes an infringing act); CONTU REPORT, supra note 40, at 12 (“[T]he placement of any copyrighted work into a computer is the preparation of a copy and, therefore, a potential infringement of copyright.”).


The legislative history regarding 17 U.S.C. § 102(b) indicates that copyright protection of computer programs covers “the expression adopted by the programmer . . . and that the actual process or methods embodied in the program are not within the scope of copyright law.” H.R. REP. NO. 1476, supra note 55, at 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS, at 5670. In addition, if the idea of the copyright is so narrow that the subject requires a very limited range of expressions, the idea and expression are said to merge and literal copying of the expression does not infringe the copyright. Morrissey v. Proctor & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967). Thus, a copyright in a lawyer’s form is not infringed by literal copying when the language it contains may be expressed in a very limited number of ways to accomplish the same purpose. See Continental Casualty Co. v. Beardsley, 253 F.2d 702, 705-06 (2d Cir. 1958). CONTU indicated that the Morrissey doctrine should be applied to computer programs, CONTU REPORT, supra note 40, at 20, and the Third Circuit held that the doctrine was applicable to computer programs. See Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), cert. denied, 104 S. Ct. 690 (1984).
builder from constructing the dwelling pictured, the copyright only protects against reproducing the actual blueprint. The distinction between the expression of a copyrighted work and the idea behind the work is incapable of being drawn with a precise line in many cases. For example, a play that duplicated Romeo and Juliet's famous dialogue has obviously appropriated the expression of the play; however, if the "pirating" play consisted of essentially the same plot set in a different time period with some specific similarities to Romeo and Juliet, the determination of whether it appropriated Shakespeare's expression or his ideas would become more difficult. Instead of viewing an alleged copy as simply appropriating either the expression or the idea, the distinction is better envisioned as a continuum, progressing gradually from appropriating pure ideas to appropriating the expression word for word. The middle "gray" area is partly expression and partly idea, making it difficult to draw the line between infringing and noninfringing uses of the copyrighted work.

The copyright laws, by distinguishing between an idea and its expression, attempt to strike a balance between the competing goals of protection of works and free dissemination of ideas. This balance must be struck to further the stated purpose of the copyright laws: "To Promote the Progress of Science and useful Arts." The copyright laws were designed to give authors a limited monopoly, the primary goal of which is

60. Imperial Homes Corp. v. Lamont, 458 F.2d 895, 899 (5th Cir. 1972).
61. Judge Learned Hand, in one of his earliest opinions on the subject, stated that the line between expression and idea, "wherever it is drawn, will seem arbitrary." Nichols v. Universal Pictures Corp., 45 F.2d 119, 122 (2d Cir. 1930).
62. The characteristics ascribed to a "pirating" play are found in West Side Story, which bears a resemblance to Shakespeare's Romeo and Juliet. Nimmer argues that West Side Story could be classified as infringing the expression of Romeo and Juliet, see 3 M. NImmer, supra note 36, at § 13.03[A][1], while other commentators, using a different test argue that it merely appropriates Shakespeare's ideas, see Knowles & Palmieri, Dissecting Krofft: An Expression of New Ideas in Copyright?, 8 SAN FERN. V.L. REV. 109, 152-66 (1980).
63. See Knowles & Palmieri, supra note 62, at 126.
64. "The guiding consideration in drawing the line [between idea and expression] is the preservation of the balance between competition and protection reflected in the patent and copyright laws." Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971).
66. CONTU REPORT, supra note 40, at 16. See generally R. Posner, ECONOMIC ANALYSIS OF LAW § 13.6 (2d ed. 1977) (comparing copyright protection with monopoly and arguing that the protection should be limited.)
wide dissemination of authors' ideas. Without copyright protection, it is argued, authors would have little to gain by marketing their works; a pirate could immediately copy the work and sell it for less than the original. Thus, some proprietary protection is necessary to achieve an economic incentive to create new works. The policy in favor of dissemination requires that protection be limited merely to that which is necessary to achieve this incentive to create. Prohibiting the further use of the idea in a copyrighted work would not only narrow the range of artistic endeavors, but would also prevent most of the cumulative innovation that results from building on knowledge. Therefore, the copyright laws prohibit the unproductive use of a protected work (copying the expression), but allow the economically and artistically beneficial use (copying the idea).

III. USE (AND NONUSE) OF THE SUBSTANTIAL SIMILARITY TEST

The most obvious method of attempting to prove copying in any medium is to show that the defendant transcribed the work verbatim. Plaintiffs, however, would face tremendous difficulty in winning a copyright infringement action if they were required to prove literal copying. Because independent creation of the copyrighted work does not infringe the copyright, the

68. This is especially true for computer programs because the cost of a literal reproduction of a computer program is far smaller than the cost of developing a computer program. See J. Soma, THE COMPUTER INDUSTRY 120 (1976).
69. See CONTU REPORT, supra note 40, at 16.
70. Lord Mansfield stated the basic conflict in 1785 when he wrote: [W]e must take care to guard against two extremes equally prejudicial, the one that men of ability, who have employed their time for the service of the community may not be deprived of their just merits, and the reward of their ingenuity and labour; the other, that the world may not be deprived of improvements, nor the progress of the arts be retarded. Sayre v. Moore, 102 Eng. Rep. 139, 140 (1785).
71. See, e.g., Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124, 131 (E.D. Mich. 1979) ("[W]ith respect to the useful arts, there is a societal interest in having many offer the art in the marketplace. Our economy functions best under competition. And, if many can present variants on the copyrighted material, we hope that advances in its teaching will result."); see also Chafee, Reflections on the Law of Copyright, 45 COLUM. L. REV. 503, 511 (1945) ("Progress would be stifled if the author had a complete monopoly of everything in his book. . . .")
72. Since the copyright protects the expression and not the idea, anyone may create an identical work as long as he or she does not use the copyrighted work in preparing the identical work. See Fred Fisher, Inc. v. Dillingham, 298 F. 145, 147 (S.D.N.Y. 1924).

Independent creation in this context refers to producing a work identical
plaintiff, to prove literal copying, would have to show not only that the alleged copy is the same as the copyrighted work, but also that the defendant literally transcribed the copyrighted work in preparing the alleged copy. Because pirates are unlikely to be obvious about their copying, proof of the direct use of the copyrighted work in preparing a copy is virtually impossible.

This problem of proof, as well as the difficulty in determining whether the expression rather than the idea of a work has been appropriated, led courts to the development of the substantial similarity test for all copyright actions. The test accomplishes two interrelated functions: (1) it shifts the burden of persuasion on the issue of use of the copyrighted work to the defendant after the plaintiff has shown similarities between the works, and (2) it provides a guide to determine whether the expression or the idea of the copyrighted work has been appropriated.

A. Development of the Substantial Similarity Test

The principal beginnings of the present substantial similarity test are found in Learned Hand’s opinion in Nichols v. Universal Pictures Co. To illustrate how to determine whether the play “The Cohens and the Kelleys” infringed the expression substantially similar to a copyrighted work without basing the work on the copyrighted work. See Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49, 54 (2d Cir.), cert. denied, 298 U.S. 669 (1936).

73. Especially with mass-marketed works, a potential pirate would have numerous opportunities to view a work and copy it without the copyright owner's knowledge. The actual copying may occur in private.


75. For a description of the difficulty in proving copying, see id. at 907. For an analysis of the idea/expression problem and its relationship to the substantial similarity test, see Nichols v. Universal Pictures Corp. 45 F.2d 119, 121-23 (2d Cir. 1930).

76. See Granite Music Corp. v. United Artists Corp., 532 F.2d 718, 721 (9th Cir. 1976). Courts have held that it is clearly erroneous to find for the defendant when the plaintiff has established a prima facie case of substantial similarity and no countervailing evidence of independent creation has been presented by the defendant. See Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110-11 (9th Cir. 1970); R. Dakin & Co. v. Charles Offset Co., 441 F. Supp. 434, 438-39 (S.D.N.Y. 1977).

77. See Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930).

78. 45 F.2d 119 (2d Cir. 1930). Tests resembling the substantial similarity test were used prior to Judge Hand’s decision. See Marks v. Leo Feist, Inc., 290 F. 959, 960 (2d Cir. 1923) (“To constitute an infringement of the appellant's composition, it would be necessary to find a substantial copying of a substantial and material part of it.”); Emerson v. Davies, 8 F. Cas. 615, 625 (D. Mass. 1845) (No. 4438).
sion of the play "Abie's Irish Rose," Judge Hand described a series of abstractions of the copyrighted work. He began with the general theme of the play, and then successively added detail from the play so that the abstraction more closely resembled a literal duplication of the script. Drawing an exact line between abstractions that appropriate the expression and ones that merely use the ideas was impossible, according to Judge Hand. Instead, Judge Hand stated that the work should be viewed as a whole, without expert testimony on the issue of similarities, and the factfinder must draw the inevitably arbitrary line based "upon the firmer, if more naive, ground of its considered impressions upon its own perusal."

Although Judge Hand's "abstractions" test has since undergone modification, the basic test for substantial similarity remains. The Ninth Circuit, in *Harold Lloyd Corp. v. Witwer,* emphasized that the substantial similarity determination is to be made from the vantage point of an "ordinary observer." Unless an ordinary observer would consider the alleged copy to be substantially similar to the protected work, the "pirate" has not infringed the copyright. Other courts have referred to the standard as an "audience" test, in which the factfinder should view the work as the actual audience of the work would view it, not just as any ordinary observer would view it. In either case, expert testimony and detailed dissection of the work to show similarities are useless because the reaction of the ordinary observer or audience is to be "spontaneous and immedi-

---

79. 45 F.2d at 120-21.
80. Id. at 121.
81. In *Nichols,* Judge Hand chastised the attorneys for their lengthy presentation of expert evidence regarding specific similarities between the two plays. Judge Hand expressed the hope that "such evidence may in the future be entirely excluded." Id. at 123.
82. Id. Judge Hand concluded that the two plays were not substantially similar and therefore the infringement action was dismissed. Id.
83. 65 F.2d 1 (9th Cir. 1933).
84. Id. at 10. In subsequent cases, it is interesting to note the different ways courts refer to this standard. It has been termed the ordinary observer test, the ordinary reasonable person test, and the ordinary lay hearer test. There appear to be no differences among these tests beyond their labels.
85. Id.
86. *See, e.g.,* Atari, Inc. v. North American Philips Consumer Elecs. Corp., 672 F.2d 607, 619 (7th Cir.), cert. denied, 103 S. Ct. 176 (1982) (applying the audience test). At least one commentator has suggested that the audience test provides a better standard because it views the similarities from the point of view of the actual observers, thus better estimating the probable economic effect of the "pirate" on the copyrighted work's market. *See* Note, *Copyright Infringement Actions: The Proper Role for Audience Reactions in Determining Substantial Similarity,* 54 S. CAL. L. REV. 395, 396 (1981).
Simply put, the trier of fact should view the two works as any observer or audience would and decide intuitively whether the expression was appropriated.

More recently, courts have added a preliminary level of inquiry to Learned Hand’s “abstractions” test. The courts first inquire whether the defendant had access to the copyrighted work, or, in other words, whether the defendant had an opportunity to view the work. If so, courts next use the ordinary observer test to determine whether substantial similarity exists. If, on the other hand, there is no reasonable probability of access, courts use the more stringent “striking similarity” test. Under that test, copying will be inferred only if the similarities are so striking that they “preclude the possibility of independent creation.” Both the substantial similarity test and the striking similarity test rely on the ordinary observer to view the similarities without the use of expert testimony.

In *Arnstein v. Porter* the Second Circuit enunciated a different two-step substantial similarity test. Under the *Arnstein* approach the factfinder first determines whether the defendant used the plaintiff's work in preparing the alleged copy by

---

87. *Harold Lloyd Corp.*, 65 F.2d at 18. As explained in *Frankel v. Irwin*, 34 F.2d 142 (S.D.N.Y. 1918), the introduction of expert testimony leads to “the classic difficulty of not being able to see the forest for the trees.” *Id.* at 144.

88. Access has been considered by some courts to be synonymous with actual viewing and knowledge of the copyrighted work. See *Bradbury v. Columbia Broadcasting Sys., Inc.*, 287 F.2d 478, 479 (9th Cir. 1961); *Christie v. Harris*, 47 F. Supp. 39, 40 (S.D.N.Y. 1942). The more generally accepted definition of access, however, is the opportunity to view the copyrighted work. See *Smith v. Little, Brown & Co.*, 245 F. Supp. 451, 458 (S.D.N.Y. 1965), aff’d, 360 F.2d 928 (2d Cir. 1966). *See also* 3 M. *Nimmer*, *supra* note 36, at § 13.02[A] (arguing for defining access as having the opportunity to view the copyrighted work).

89. The plaintiff cannot satisfy the access requirement by merely showing a “bare possibility” of access, there must be a “reasonable possibility” that the defendant had the opportunity to view the copyrighted work. *Testa v. Janssen*, 492 F. Supp. 198, 202-03 (W.D. Pa. 1980).

90. *See Ferguson v. National Broadcasting Co.*, 584 F.2d 111, 113 (5th Cir. 1978); *see also* 3 M. *Nimmer*, *supra* note 36, at § 13.01[A].

91. The striking similarity test increases the burden on the plaintiff to show more similarities, but does not fundamentally change the underlying test. *See Ferguson*, 584 F.2d at 113.

92. 154 F.2d 464 (2d Cir. 1946).

93. The court in *Arnstein* phrased the first issue as whether the defendant “copied” the copyrighted work. *Id.* at 468-69. The court applied the word “copied” not in its statutory sense, but rather to refer to the defendant's use of the copyrighted work in preparing the alleged infringing work. Thus, one has “copied” a work if he or she has viewed the work and applied anything from the work to the alleged infringing work. *Id.* In order to prevent confusion, this Note will refer to *Arnstein's* first step as defendant's use of the copyrighted work.
showing a combination of access and similarities. Expert evidence and detailed dissection of the work are relevant to this determination because the issue is merely whether the copyrighted work was used in creating the alleged copy, not whether the expression of the copyrighted work was appropriated. If use of the copyrighted work is shown, the second step consists of determining whether the defendant appropriated the expression or the idea. Under this step, the Arnstein test resembles the abstractions test in its use of the ordinary observer standard to judge the similarities between the works. Detailed dissection of the work and expert testimony become irrelevant to this determination—the intuition of the ordinary observer controls.

The Ninth Circuit, in Sid & Marty Krofft Television Productions v. McDonald's Corp., interpreted Arnstein as creating a "bifurcated" test for showing substantial similarity. The Ninth Circuit changed the first step of the Arnstein test into an "extrinsic" determination of whether the ideas of the alleged copy resemble the ideas of the copyrighted work. The subject matter of the works, the materials used, and the type of artwork created, among other things, must be examined to determine whether the ideas are similar. Expert testimony and

94. The court in Arnstein emphasized that if the requisite similarities did not exist between the works, then even the highest showing of access would not result in a finding of infringement. Id. at 463.
95. Id.
96. Id.
97. Id. In Arnstein, the copyright issue concerned copying a musical composition. On a summary judgment motion by the defendant, the court held that, although the plaintiff’s story was improbable, summary judgment would not be granted unless there was not the slightest doubt as to the facts. Id. at 469-70. This aspect of the Arnstein holding is no longer good law. See, e.g., First Nat’l Bank v. Cities Service Co., 391 U.S. 253, 288-90 (1968). The two-step copyright infringement test of Arnstein, however, is still followed by some courts. See, e.g., Ferguson, 584 F.2d at 1141.
98. 562 F.2d 1157 (9th Cir. 1977). For a detailed discussion of the Krofft case, see Knowles & Palmieri, supra note 62, at 103.
99. Id. at 1164-65. Although the Krofft court stated that they were following the Arnstein approach, they also noted that the precedential value of the “bifurcated” test did not rely on whether their reading of Arnstein was correct. Id. at 1165-66 n.7. It is clear that Krofft modified the Arnstein approach to some extent.
100. The first step is labelled “extrinsic” because it depends not on the responses of the trier of fact, but on specific criteria that can be listed and analyzed. Thus, the relevant factors are extrinsic to the trier of fact. Id. at 1164.
101. Nimmer has characterized this “similarity in ideas” standard as loosening the first step of the Arnstein approach, since similar ideas will almost always be present, whereas “copying” requires an inference of use of the work from the similarities. See 3 M. Nimmer, supra note 36, at § 13.03[B](3).
102. Krofft, 562 F.2d at 1164.
detailed dissection of the works are appropriate to show these similarities and the factfinder examines the specific criteria, not as an ordinary lay observer, but as a trier of fact examining all the expert and detailed evidence. Assuming the plaintiff establishes general similarity between the ideas of the two works, the court turns to the "intrinsic" aspect of the test, which duplicates the Arnstein test by using the ordinary observer standard and disallowing the use of expert testimony to show similarities between the expressions of the two works.

It is tempting to overemphasize the contrasts between the different versions of the substantial similarity test and ignore their fundamental likenesses. Basically, courts apply either of two approaches to determine substantial similarity: the "abstractions" test, which first focuses on whether access to the copyrighted work existed and then uses the standard of the ordinary observer to measure the extent of similarities in expression; or the Arnstein or "bifurcated" approach which first determines whether the "pirate" used the copyrighted work (or, in the Ninth Circuit under Krofft, whether the two works contain the same ideas) and then, as in the "abstractions" test, focuses on the ordinary observer to detect similarities in expression. Under either approach, expert evidence and detailed dissection of the work are not relevant to show appropriation of the expression because the ordinary observer is the focal point of that inquiry.

B. USE AND NONUSE OF THE SUBSTANTIAL SIMILARITY TEST FOR COMPUTER PROGRAMS

The few courts presented with a question of illicit copying of a computer program have rarely mentioned the substantial similarity test and never applied the test in substance. In Synercom Technology, Inc. v. University Computing Company, University Computing Company (UCC), after viewing

---

103. Id.
104. Id. The second part of the test is described as "intrinsic" because it is governed by the general observations and intuition of the ordinary observer and not specific criteria. Id.
Synercom's copyrighted computer program, engineered a program containing Synercom's exact input formats.\textsuperscript{107} The court characterized UCC's procedure for producing the new program as involving substantial work and creativity.\textsuperscript{108} Synercom, however, argued that UCC had illicitly copied the input formats from the copyrighted program. Although the case clearly contained a "copying" problem controlled by the substantial similarity test,\textsuperscript{109} the court avoided any mention of the test or its standards, and instead merely analyzed whether the idea or the expression of the copyrighted formats had been appropriated.\textsuperscript{110} The court reasoned that the sequence and ordering of data in an input format is an idea, much like the figure "H" pattern of an automobile stick shift is an idea, not an expression.\textsuperscript{111} In the court's view, UCC had not copied Synercom's expression, but had created "original expressions of the copied

\textsuperscript{107} An input format is a part of a computer program that specifies the order and manner of inserting data into a program for execution. \textit{Id.} at 1005.

\textsuperscript{108} \textit{Id.} at 1013 n.5. As described by the court in \textit{Synercom}, the defendants' use of the copyrighted program resembled reverse engineering, which consists of viewing the functioning of a computer program and creating a similar program to accomplish the same results as the original. The defendants' purpose in duplicating the input formats was to achieve compatibility with Synercom's computer hardware. \textit{See id.} at 1012. Compatibility refers to the ability of a program to be run on a specific computer. A program is IBM compatible if it matches with the operating system of the IBM computer so that the program can be read by the computer. Obviously, some similarities are required to make a program compatible.

\textsuperscript{109} As stated by the court in \textit{Synercom}, the primary issue concerned whether the defendants appropriated the expression or the idea of the copyrighted program. \textit{Id.} at 1013. As noted by Learned Hand in \textit{Nichols}, the substantial similarity test is used to judge the idea/expression dichotomy. Nichols v. Universal Pictures Corp., 45 F.2d 119, 122 (2d Cir. 1930).

\textsuperscript{110} 462 F. Supp. at 1013.

\textsuperscript{111} \textit{Id.} at 1013. The \textit{Synercom} court held, alternatively, that the input formats of the program were not copyrightable subject matter. \textit{Id.} at 1014. According to the court, the test in determining whether the subject matter was copyrightable was "whether the material proffered for copyright undertakes to express." \textit{Id.} at 1011. The Copyright Act, however, does not contain such a requirement. \textit{See supra} note 37 and accompanying text. Instead, the Copyright Act requires the material be "fixed in any tangible medium of expression." 17 U.S.C. § 102(a) (1982). Certainly, the source code computer program is a medium of expression and the input formats were fixed in the computer program, thus satisfying the copyright requirements.

The \textit{Synercom} court confused the existence of copyright protection with the infringement question. Whether the input formats are an expression or an idea is properly considered in determining if the defendant "copied" the copyrighted work, but it is not properly considered in determining if copyright protection existed. \textit{See generally} Gorman, \textit{Copyright Protection for the Collection and Representation of Facts}, 76 HARV. L. REV. 1569, 1600-02 (1963) (discussing the copyright protection of analogous business forms as "easy-to-copyright/difficult-to-infringe.").
idea" and thus no infringement was found.\textsuperscript{112}

Courts in two more recent cases have superficially discussed the copying of a computer program after determining that copyright protection existed for the computer program. In \textit{Williams Electronics v. Artic International, Inc.},\textsuperscript{113} after deciding that copyright protection extended to a computer program stored in a silicon chip,\textsuperscript{114} the court considered whether Williams's program had been copied. The plaintiff produced overwhelming evidence of similarity, including a "buried" copyright notice in the copy,\textsuperscript{115} the same error in both the original and the copy,\textsuperscript{116} and identical locations\textsuperscript{117} of 85\% of the object code.\textsuperscript{118} Without discussing the application of the substantial similarity test, the court simply held that the expert evidence of similarities constituted overwhelming evidence of copying.\textsuperscript{119}

Similarly, in \textit{Hubco Data Products Corp. v. Management Assistance Inc.},\textsuperscript{120} after stating that an operating program expressed in object code constituted a work of authorship and thus was probably copyrightable,\textsuperscript{121} the court turned to the determination of whether Hubco copied the program. The court stated that the Ninth Circuit's "bifurcated" standard\textsuperscript{122} controlled, but failed to apply either step of the test to the infringement issue. Instead, the court studied the expert testimony concerning Hubco's creation of an exact duplicate of Management Assistance's program.\textsuperscript{123} Hubco had taken Management Assistance's copyrighted program and either made a written printout of the object code or stored the program in an internal computer memory. Then, Hubco located and removed the

\begin{itemize}
  \item \textsuperscript{112} 462 F. Supp. at 1013.
  \item \textsuperscript{113} 685 F.2d 870 (3d Cir. 1982).
  \item \textsuperscript{114} \textit{Id.} at 873-75.
  \item \textsuperscript{115} \textit{Id.} at 876 n.6. A "buried" copyright notice is stored inside the program and is never visible to the user when the program is run. When one makes a duplicate of the chip or floppy, however, the copyright notice is transferred to the copy with the rest of the program.
  \item \textsuperscript{116} \textit{Id.} The error found in both the copyrighted work and the alleged copy in the \textit{Williams} case consisted of displaying an incorrect score for destroying a particular alien in the video game program. \textit{Id.} n.6.
  \item \textsuperscript{117} \textit{See supra} note 17 and accompanying text.
  \item \textsuperscript{118} 685 F.2d at 876 n.6.
  \item \textsuperscript{119} \textit{Id.} at 876. Though the defendant on appeal did not dispute the copying claim in \textit{Williams}, the court specifically mentioned the similarities used to infer copying.
  \item \textsuperscript{120} 2 COPYRIGHT L. REP. (CCH) \$25,529 (D. Idaho Feb. 3, 1983).
  \item \textsuperscript{121} \textit{Id.} at 18,105. Because the court was ruling on motions for a temporary restraining order and a preliminary injunction it merely decided the probability that the program was copyrightable.
  \item \textsuperscript{122} \textit{See supra} notes 98-104 and accompanying text.
  \item \textsuperscript{123} 2 COPYRIGHT L. REP. (CCH) at 18,102.
\end{itemize}
"governors," which were designed to provide a pricing mechanism for Management Assistance's program. Hubco's purpose in removing the governors was to provide extra memory and peripheral capacity in the computer system for the user without compensating Management Assistance. Although never mentioning the ordinary observer standard or the exclusion of expert testimony on the expression question, the court determined that the duplicative program was probably a copy, and that Hubco therefore had probably infringed Management Assistance's copyright.

Whereas both Williams and Hubco only superficially discussed the copying issue, the court in Midway Manufacturing Co. v. Strohon considered the question at greater length. After holding that copyright protection existed for a program fixed in object code and stored in a silicon chip, the court determined whether the defendant had copied the chips. The Court, stating that it was applying the modified "abstractions" test, first considered whether the defendants had access to the work, and then whether the expressions of the two works were substantially similar. The defendants could not dispute that they had access; they contended, however, that the similarities between the two programs did not establish copying. As presented by the plaintiff, the similarities consisted exclusively of object code contained in silicon chips in the alleged copy that duplicated the object code contained in the copyrighted program. Experts testified that, by printing out the object code from Strohon's silicon chips, they determined

---

124. A "governor" is a device that restricts the operating capacity of the program by restricting its ability to compute large quantities of numbers or by reducing its memory capacity. Id.

125. Management Assistance had placed these "governors" on the program in order to differentiate between purchasers who required different quantities of storage and operational capacity, even though they purchased the same program. Purchasers who required very little storage would purchase programs with very restrictive governors at relatively low prices. If an extensive amount of storage were required, purchasers would buy programs with less restrictive governors. Thus, Management Assistance was able to differentiate among heavy and light users by price. Hubco thwarted this price discrimination by removing the governors on the program, thus increasing the storage without compensating Management Assistance. Id. at 18,101.

126. Id. at 18,106-07.


128. Id. at 752.

129. See supra notes 88-91 and accompanying text.

130. 564 F. Supp. at 752.

131. Id. at 752-53. The court did not discuss any similarities other than the percentage of duplication of object code on the infringement issue. Id.

132. A Microcomputer Development Lab is used to decipher the bits of ob-
that 89% of the bytes\textsuperscript{133} were identical to the plaintiff's bytes.\textsuperscript{134} From this evidence, the court characterized the degree of similarity as substantial, although not absolute.\textsuperscript{135} Again, however, the court failed to adhere to either the ordinary observer standard or the exclusion of expert testimony,\textsuperscript{136} both of which characterize the abstractions test. Thus, beyond merely stating the access and similarity of expression standards, the Midway court failed to apply the abstractions test's standards.

IV. ANALYSIS OF THE APPLICATION OF SUBSTANTIAL SIMILARITY TO COMPUTER PROGRAMS

A. EXPLANATIONS FOR THE COURTS' NONAPPLICATION OF SUBSTANTIAL SIMILARITY

As the four cases discussed above demonstrate, courts are reluctant to apply the current forms of the substantial similarity test to determine copying of a copyrighted computer program. At least two possible explanations account for this judicial reluctance: either the courts have found the infringing nature of the copy so obvious that they view a discussion of substantial similarity as unnecessary, or the substantial similarity test as presently constructed is simply unworkable in its application to computer programs.\textsuperscript{137} The first explanation could account for the court's opinion in Williams, since copying was essentially admitted by the defendant.\textsuperscript{138} In Hubco, Midway, and Synercom, however, the defendants contested the copying question, and the determination of an infringing copy in each use was not obvious enough to avoid analysis of the issue.\textsuperscript{139} Thus, at least to some degree, the courts must view the

\footnotesize{ject code contained in a silicon chip and print the information out as 0's and 1's. \textit{Id.} at 752.

\textsuperscript{133} See supra notes 17-19 and accompanying text.

\textsuperscript{134} Since a byte contains eight locations or bits of object code, an identical byte would consist of matching 0's and 1's in each of the eight locations. See \textit{id.} at 752.

\textsuperscript{135} \textit{Id.} at 753.

\textsuperscript{136} Not only did the court fail to disregard the expert testimony on the copying issue, it considered the defendant's failure to present expert evidence as "glaring and unfortunate." \textit{Id.} at 749 n.5.

\textsuperscript{137} A third possibility might be that the courts are simply misapplying the substantial similarity test. In light of the well-developed case history of the substantial similarity test and its simple standards, however, this possibility seems remote.

\textsuperscript{138} Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 876 (3d Cir. 1982).

\textsuperscript{139} In Hubco the defendant's claim of not copying was extremely tenuous because the "pirating" work essentially duplicated the copyrighted program; but the argument apparently was made. Further, the court in Hubco bothered to state the abstractions test for substantial similarity, thus indicating that it}
substantial similarity test in its present form as unsuitable for determining whether copyrighted computer programs have been copied.

B. Defects in Applying the Present Substantial Similarity Test to Computer Programs

Were the courts to strictly apply the substantial similarity test in one of its current forms, two general problems would arise. First, application of an analysis centered on the "ordinary observer" or "audience" to a computer program, which ordinarily is not "observed," would result in an improperly focused test. Second, application of judicial precedent from the use of the substantial similarity test on other copyrighted works might result in overprotection of computer programs, thereby causing lessened innovation in the software field.

1. The Ordinary Observer Standard is Inapplicable to Computer Programs.

The ordinary observer analysis that is used in finding substantial similarity under both the abstractions140 and the *Arnstein*141 test would incorrectly focus the determination of an infringement action if it were applied to computer programs. The ordinary observer inquiry was designed to determine general aesthetic similarities between the copyrighted work and the alleged copy.142 For example, the ordinary observer would view two plays in their entirety to determine whether they are substantially similar, rather than comparing each line of the

---

140. See supra notes 88-91 and accompanying text.

141. See supra notes 92-97 and accompanying text.

142. See *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 122-23 (2d Cir. 1930); *supra* notes 91-97 and accompanying text.
plays for identical words, phrases, or letters. Ordinary observers can make the substantial similarity judgment based on their background of knowledge as theatergoers. Computer programs, however, have neither ordinary observers nor an easily perceived aesthetic appeal. The actual computer program stored in a silicon chip or floppy is rarely, if ever, seen by the user. Instead, the computer operator views only the functioning and the output of programs, neither of which is protected by the copyright laws. Since there are no ordinary observers of computer programs, the application of the ordinary observer standard to judge similarities in computer programs would be an entirely fictitious process. Compounding this problem is the absence of an easily perceived general “aura” or “feeling” that an ordinary observer could judge, especially if the program is fixed in the binary form of object code. A series of 0’s and 1’s arranged in a particular order in a computer program simply fails to convey a general feeling to an ordinary person. The only way to obtain a general feeling from a computer program is to observe the unprotected output and functioning of the program. Observing the outward, general similarities, however, changes the focus of the copyright protection from the underlying program to the product of the program.

The exclusion of expert evidence on the issue of similarities further compounds the problem created by the lack of observers and the absence of an applicable aesthetic appeal. The exclusion of expert testimony in noncomputer copyright cases reflects the desire not to allow minor, technical similarities or dissimilarities to confuse the ordinary observer’s general comparative feeling about how the works compare. Because computer programs possess a general aesthetic feel only from their functioning and output, however, expert testimony is

143. See Harold Lloyd Corp. v. Witwer, 65 F.2d 1, 19 (9th Cir. 1933).
144. A copyright in a computer program does not protect the output or functioning of the program. A programmer could create a different program that would produce exactly the same results and output as a copyrighted program and not infringe the copyright. See CONTU REPORT, supra note 40, at 20-21. Therefore, one should not merely judge the similarities in the functions or output of computer programs for the purpose of determining infringement.
145. Even the high-level languages lack aesthetic appeal. See P. SANDERSON, COMPUTER LANGUAGES: A PRACTICAL GUIDE TO THE CHIEF PROGRAMMING LANGUAGES 24 (1970) (“one must not therefore repine if a COBOL program has not the immediate readability or the aesthetic qualities of a Jane Austen novel!”).
146. Technical dissimilarities are often added to a copied work by a pirate in an attempt to make it look like an original creation. Thus, the exclusion of expert testimony in noncomputer copyright cases attempts to prevent a plagiarizer from excusing “the wrong by showing how much of his work he did not
needed to uncover any similarities between the actual programs. Further, expert evidence is extremely probative of infringement in a computer copyright case, especially if there are "buried" copyright notices, similar errors, or duplicate bytes. The courts in the four cases discussed above implicitly recognized this need when they permitted each plaintiff to present expert evidence of technical similarities. Formal application of either form of the substantial similarity test would exclude expert evidence—and thus would exclude the most probative evidence of infringement—leaving plaintiffs to rely


The exclusion of expert testimony in proving or disproving substantial similarity in noncomputer copyright areas has been criticized by some commentators. See, e.g., Sorenson & Sorenson, Re-Examining the Traditional Legal Test of Literary Similarity: A Proposal for Content Analysis, 37 CORNELL L.Q. 533, 636-54 (1952) (arguing for the structured introduction of expert testimony through "content analysis"); E. Kitch & H. Perlman, LEGAL REGULATION OF THE COMPETITIVE PROCESS 665 (2d ed. 1979) (arguing that when the alleged copy is not an exact duplicate of the copyrighted work, a close analysis of specific differences and similarities is required). While expert evidence has been allowed to show similarities in actions involving infringement of musical compositions, see Wilkie v. Santly Bros., 91 F.2d 978, 979 (2d Cir. 1937); Arnstein v. ASCAP, 29 F. Supp. 388, 397 (S.D.N.Y. 1939), it has been excluded in almost all other copyright areas, see, e.g., O'Neill v. Dell Publishing Co., 630 F.2d 685, 690 (1st Cir. 1980) (novels); Midway Mfg. Co. v. Bandai-America, Inc., 546 F. Supp. 125, 138 (D.N.J. 1982) (video games). See generally Comment, Copyright Protection for Mass-Produced Commercial Products: A Review of the Developments Following Mazer v. Stein, 38 U. CHI. L. REV. 807, 813 (1971).

A "buried" copyright notice is almost conclusive evidence of copying because the notice would be extremely unlikely to appear in an independently created work. See Williams Elecs., Inc. v. Artic Int'l Inc., 565 F.2d 870, 876 n.6 (3d Cir. 1982).


Duplicate bytes, although not conclusive evidence because there may be only a limited number of ways to program a function, are a strong indication of literal copying. See Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 752-53 (N.D. Ill. 1983).


The Arnstein test would allow the evidence, but only for showing that the defendant used the copyrighted work, not for the purpose of showing simi-
merely on the factfinder's intuitive assessment of the general similarities between the programs.

The effect of the ordinary observer standard and its exclusion of expert evidence can be seen more clearly by examining a hypothetical infringement action. Suppose, as in Williams, that a copyright holder in a mass-marketed program offered expert evidence that the defendant's silicon chips duplicated 80% of the object code from the copyrighted program and that the alleged copy contained a hidden copyright notice from the copyrighted program. Under the "abstractions" test, since access is satisfied,\textsuperscript{152} the inquiry turns on whether an ordinary observer would regard the programs as similar without the use of expert testimony.\textsuperscript{153} Excluding the expert testimony prevents the plaintiff from showing the technical similarities such as the hidden copyright notice. The factfinder must try to obtain an intuitive sense of similarity between the programs based on any nonexpert evidence the plaintiff can provide. With the \textit{Arnstein} test, although the plaintiff could show that the defendant "used" the copyrighted work in preparing the copy,\textsuperscript{154} the same ordinary observer standard precludes expert testimony on the question of appropriation of the work's expression.\textsuperscript{155} Though infringement in this example may eventually be found based on the outward evidence of similar functions and outputs, the best evidence of copying, the buried copyright notice, would never be considered by the factfinder. Further, the factfinder would be confused by attempting to judge the similarities from the perspective of a nonexistent observer. Thus, the present forms of the substantial similarity test provide poor frameworks within which to measure similarities in computer programs.

2. Overprotection of the Copyrighted Program.

Application of the present forms of the substantial similarity test to computer programs may also lead to overprotection of the copyright owner's interests to the detriment of innovations in the expressions of the two works. \textit{See supra} notes 95-97 and accompanying text.

\textsuperscript{152} Because the program is mass-marketed, the defendant would almost certainly have the \textit{opportunity} to view the work. \textit{See supra} note 88 and accompanying text.

\textsuperscript{153} \textit{See supra} notes 88-91 and accompanying text.

\textsuperscript{154} The duplicate copyright notice would indicate that the defendant had used or "copied" the copyrighted work in some manner, thus satisfying the first step of the \textit{Arnstein} test. \textit{See supra} notes 92-97 and accompanying text.

\textsuperscript{155} \textit{See supra} note 97 and accompanying text.
tion. Paradoxically, while the use of the ordinary observer standard results in preventing a plaintiff from presenting the evidence most probative of infringement in literal copying cases, the expansive view of expression under the substantial similarity test would result in increased, and perhaps overly broad, protection in nonliteral copying cases.

In noncomputer copyright cases, the copyright laws are interpreted to protect more than the literal arrangement of the program's statements from copying. Outside the computer program context, "expression" denotes not only the literal arrangement of words, sounds or numbers, but also the "pattern" of the work, or "the sequence of events and the development of the interplay of the characters." By copying this pattern of a copyrighted work, an individual has infringed the copyright. Thus, if one creates a television commercial with characters and costumes similar to a copyrighted TV show's characters, even though they act differently, the copyright is still infringed.

Though the broad scope of the term expression, and the consequent broad scope of protection, are useful in noncomputer copyright areas, such may not be the case in the computer field. With many aesthetic works, a "pirate" can simply listen, read, or observe the copyrighted work and relatively easily produce a comprehensive nonliteral copy by using slightly different words or actions to express the same idea. The infringer can thus accomplish the purpose of "pirating," capturing the economic market of the copyrighted work at minimal cost, without making a literal copy. Computer programs, however, involve exact instructions with each statement precisely linking with other statements to produce a functional pro-

---

156. Literal copying consists of reproducing the copyrighted work word for word. Nimmer classifies copyright infringement cases as either comprehensive nonliteral copying or fragmented literal copying. See 3 M. Nimme, supra note 36, at § 13.03[A][1], [2].

157. Nonliteral copying, or as Nimmer calls it, comprehensive nonliteral copying, changes the words used in the copyrighted work but maintains the "fundamental essence or structure." Id. at § 13.03[A][1].

158. Chafee, supra note 71, at 514. This is essentially the same as Nimmer's comprehensive nonliteral copying. See 3 M. Nimme, supra note 36, at § 13.03[A][1].

159. Sid & Marty Krofft Television Prods. v. McDonald's Corp., 562 F.2d 1157, 1167 (9th Cir. 1977) (McDonaldland infringed the copyright in H.R. Pufnstuf).

160. For example, one could take an author's book and by using a thesaurus change most of the words while keeping the same ideas and produce a reasonable substitute. Undoubtedly, the "pirate" would have invested far less work in producing the nonliteral copy than the original author.
While one could change the noncritical parts of a source code program, such as the numbers for each program statement or the names of any variables, changing the operative statements would require precise replacements and the investment of substantial knowledge and time by the pirate. One cannot simply "approximate" the entire copyrighted computer program and create a similar operative program without the expenditure of almost the same amount of time as the original programmer expended. If one approximated the copyrighted program solely for the purpose of reproducing it, the added time would substantially reduce the incentives, especially in the fast-moving software industry. Creating a nonliteral copy of a computer program in object code, which is the form of almost all mass-marketed programs, is even more

161. See J. O'Brien, supra note 10, at 227-28. (Program coding "involves a rigorous process which requires the computer programmer to strictly follow specific rules concerning format and syntax."). Errors as simple as a misplaced punctuation mark in a computer program can result in the program not operating correctly. Id. at 231; see also Davidson, supra note 32, at 377 ("A computer is very literal minded and the instructions in the manual translation must be very precise in order to work.").

162. Each statement in a source code program may be numbered, but the numbers serve only a nominal purpose. A "pirate" could simply change the numbers by keeping them in numerical order but counting by twos.

163. Each variable is given a name in the program. The names have no independent meaning to the computer beyond identifying the variable and so could be changed by the pirate without changing the program's operation.

164. See Keplinger, supra note 8, at 487 (Especially in microcomputer programs, "there is a significant investment in the coding, testing, and debugging of the program rather than in the underlying algorithm or process."). One commentator argues that a programmer can in a matter of days cosmetically change a source code program to make it look completely different from the original copyrighted program. Davidson, supra note 32, at 378. But the examples the commentator provides of cosmetic changes to a program consist of selecting different dimensions for variables and incorporating data tables into the program. Id. The critical parts of the program (i.e., the actual operating functions), however, would still be identical to the copyrighted work, even with these cosmetic changes. Thus, evidence of a literal reproduction would still appear in the "pirating" work. To change these underlying processes of the copyrighted program would require much more time than a few days and require significant amounts of technical skill from the pirate.

165. See D. Remer, supra note 31, at 20 (the computer industry "moves too fast to be satisfied with old ideas"). Programmers who use copyrighted programs solely to reproduce their functions must be distinguished from programmers who use the copyrighted program as a basis for further improvements or innovation. The former category is discouraged from making nonliteral copies of computer programs by the long development period required, while the latter category, as a policy matter, will not be as discouraged since they, by definition, have a stake in producing new and improved works. Further, the latter category, as a policy matter, should not be discouraged from using the copyrighted program. See infra notes 169-77 and accompanying text.

166. See supra note 31 and accompanying text.
difficult and time-consuming than source code copying since one has to decompile the program back to a type of source code before examining its individual statements and replacing them.\textsuperscript{167} Instead of proceeding through this elaborate and time-consuming process, a computer program pirate is much more likely to literally copy or mechanically translate the copyrighted program and perhaps make some minor, noncritical changes to the literal duplication.\textsuperscript{168} Thus, the threat of a computer software pirate evading the copyright law's prohibition against copying by making a comprehensive nonliteral copy is much less than in other media.

Even though creating comprehensive nonliteral copies of computer programs is unlikely because of the significant expenditure of time and knowledge required, one might view the broad protection against nonliteral copying as a useful safety net for protecting the programmer's investment in the copyrighted program. In the computer program area, however, more so than in other copyright areas, the increased protection from the broad substantial similarity test results in the unfortunate side effect of lessened innovation. The balance between protectionism and dissemination underlying the policy of copyright law becomes skewed too far toward protection, with a resultant decrease in software innovation. To understand the reasons for the lessened innovation, one must first comprehend the process of technological growth in the computer software field.

The computer software industry progresses by a stepping-stone improvement process, with each innovation building on past innovations to produce an improved product.\textsuperscript{169} Although it appears to be the result of quantum leaps, the development of computer software has actually been "founded on a succession of seemingly minor improvements on existing tech-

\footnotesize{\textsuperscript{167} See D. Remer, supra note 31, at 20. Even Mr. Davidson agrees that anything but "slavish" copying of programs expressed in object code is extremely difficult. See Davidson, supra note 32, at 380.}

\footnotesize{\textsuperscript{168} See Selinger, Protecting Computer Software in the Business Environment: Patents, Copyrights and Trade Secrets, 3 COMPUTER L.J. 65, 75 (1983) (many copyists do not even attempt to make any changes to the copied program). See also Davidson, supra note 32, at 377 ("much of the concern in the software industry is against exact copiers").}

\footnotesize{Though literal copying, with minor changes, will usually be the only feasible way for a pirate to benefit economically from a copyrighted program, this process is also much easier to detect because it will result in duplication of many of the original bytes and any hidden copyright notices.}

\footnotesize{\textsuperscript{169} As the old saying goes, "A dwarf standing on the shoulders of a giant can see farther than the giant himself." Chafee, supra note 71, at 511.}
niques." Whereas only a few programmers have the capability to create totally new methods of operations, there are many programmers who can mimic the pioneers and add improvements. Cumulatively, the minor innovations produce technological growth of major significance.

Unlike most other fields of endeavor covered by the copyright laws, the imitation and stepping-stone progress in computer programs requires plagiarizing in some manner the underlying copyrighted work. Computer programs are fundamentally different from most other literary works that produce technological growth in that the program itself both expresses the innovation and performs the new operation. For example, a book detailing a new heart transplant procedure conveys the idea of the procedure for other doctors to use in their operations. The book, however, does not perform the operation. A computer program, on the other hand, both conveys a new innovation or process and also carries out the operation. A new operating system for a microcomputer details the input and output formats necessary for compatibility with the microcomputer, and also performs the operating functions. While this might seem to be a minor distinction, it is of critical importance to copyright protection. In the heart transplant example, innovation can occur by another doctor reading the copyrighted book, modifying the technique in some small way, and performing the modified procedure without ever worrying about a copyright infringement action. In fact, the second doctor could publish the modified technique in a new book without infringing the original work's copyright. With the new com-

171. Id. at 77.
172. Id. at 37 ("Indeed, the bulk of technical progress is very often attributable to modifications of a given technique.").
173. Other works that are similar to the computer programs in this regard include maps, directories, and business forms (also known as "fact" works). The author of a fact work draws upon objective data set in a particular expression and compiles the data for publication. Because the ideas in a "fact" work are already set in a particular expression, a more limited definition of infringement is advocated. See Gorman, supra note 111, at 1570-71.

It can be argued that computer programs are analogous to "fact" works. The first computer programmer, especially for an operating program, details the necessary steps for input, processing, and output with the computer, which is comparable to accumulating the necessary information for a business form. Subsequent programmers then build on these ideas to create innovative works. Certainly, both with computer programs and "fact" works, the innovator could start from the beginning and recreate the original work; that process, however, would be duplicative and wasteful of society's resources.

174. The second doctor would only be prohibited from using the first doc-
puter operating program, however, one cannot slightly modify the copyrighted program's procedure without first recreating in some manner the original program's expression. A new innovator must therefore plagiarize and apply much of the original operating system to the new "fixed" computer program in order to add a new improvement.

The increased protection against nonliteral uses of a copyrighted work that would result from application of the broad substantial similarity test may prohibit much of this cumulative innovation and "second-sourcing." By preventing the innovator from producing a new program encompassing the patterns of the original copyrighted program, the stepping-stone innovation cannot occur. Even if an innovator changed every statement in a copyrighted program, the nonliteral "patterns" or aesthetic qualities still may be similar between the copyrighted work and the new work, and infringement could be found under the present forms of the substantial similarity test. The only reasonably safe manner to produce a program accomplishing the same purposes as a copyrighted program is to avoid any access to the copyrighted work. In effect, the programmer would have to "reinvent the wheel" in order to produce an innovation that furthers technological progress. The effect of applying the present forms of the substantial similarity test to computer programs, therefore, would be to lessen innovation in the software field.

See supra notes 70-77 and accompanying text. Second-sourcing refers to the production of an alternative software system by an independent programmer for a specified manufacturer's computer hardware. See A. CHANDOR, supra note 15, at 159.

Theoretically, if one does not view the underlying computer program, there is no possibility of infringing the copyright in the computer program. But, since the definition of access is merely the opportunity to view, see supra note 88 and accompanying text, even if a programmer does not actually view the copyrighted program, the access requirement could still be met. Further, in many instances, two independently written programs for the same computer, in the same language, and performing the same general functions will look substantially similar. Davidson, supra note 32, at 377. Since the substantial similarity test would then be met, the defendant would then have the burden of persuasion to show that the work was independently created. Showing that the alleged "pirate" did not view a particular program would be a difficult task. Thus, even the programmer of an independently created program could lose a copyright infringement suit.

A programmer may be able to view the operating manuals or the program description of the copyrighted program, if they are available. See Comment, Copyright Protection for Programs Stored in Computer Chips: Competing with IBM and Apple, 7 HAMLINE L. REV. 103, 123-27 (1984). Even the program descriptions, however, may be protected from use in preparing a copy of the program by the the copyright laws. See Davidson, supra note 32, at 379-80.
Another hypothetical will help illustrate the overprotection problem. Suppose a programmer were to dissect a copyrighted mass-marketed operating program and write a new program that accomplished the same purpose, yet improved many aspects of the original program. Applying the abstractions test reveals that the access requirement is met by the mass-marketed nature of the program. The analysis then turns on the ordinary observer's judgment of the similarities between the expressions of the two works without expert testimony. The owner of the copyrighted program could provide evidence that the defendant's reverse engineered program accomplished the same functions, ran on the same computer, and contained similar input formats as the copyrighted program. Certainly the general patterns of the two works, their sequences of events and processes, resemble each other to a large degree. To that extent, the reverse engineered program could be characterized as substantially similar to, and thus infringing, the copyrighted program. Under the Arnstein test, after finding access to and use of the copyrighted work, the factfinder could find similarity in expression since the second stage of the Arnstein test is identical to the abstractions test. The Synercom court's comment that the addition of creative effort in a technological field should remove the alleged "pirate" from the realm of infringing reproductions provides little comfort from the possibility that either version of the substantial similarity test could characterize the program as infringing the copyright.

V. ALTERNATIVE "ITERATIVE" APPROACH TO SUBSTANTIAL SIMILARITY OF COMPUTER PROGRAMS

The courts have avoided the inherent problems of applying either version of the substantial similarity test to computer pro-

---

178. Again, the mass-marketed nature of the software practically guarantees the opportunity to view the copyrighted work. See supra note 88 and accompanying text.
179. See supra notes 88-91 and accompanying text.
180. See supra note 106-08 and accompanying text.
181. Obviously, given that the ordinary observer test is inapplicable to computer programs, it is difficult to state precisely whether infringement would be found.
182. Use of the copyrighted work could be shown by the defendant's viewing of the program and attempt to simulate the program.
183. See supra notes 92-97 and accompanying text.
grams by simply not applying the test. While avoiding the problems, the courts have left the state of the law unclear by failing to articulate their reasons for not applying the test and by failing to propose a new standard for use in computer copyright infringement cases. Thus, the primary goal of the "iterative" approach proposed here is to produce a system for proper analysis of similarities between computer programs and to yield predictable results.

Only relatively minor modifications are necessary to correct the deficiencies in the current forms of the substantial similarity test. By eliminating the ordinary observer standard from the Arnstein test and reformulating the idea-expression distinction, an "iterative" substantial similarity test can be developed. Under the "iterative" test, a plaintiff would establish a prima facie case of copyright infringement of a computer program by showing: (1) that the defendant used the copyrighted work in preparing the alleged copy by offering evidence of access and similarities between the two works, and (2) that the defendant's work is an "iterative" reproduction that is, a restatement in substantially the same form, such as a literal copy or translation, of a substantial portion of the copyrighted program. Once the plaintiff has shown an iterative reproduction, the burden shifts to the defendant to show independent creation or fair use.

185. See supra notes 105-36 and accompanying text.
186. This Note does not argue with the results reached by any of the four courts, but rather takes exception to the confusion caused by the courts' lack of clarity. In fact, the use of the iterative test as proposed here would probably not change the results of Synercom, Williams, Hubco, or Midway.
187. See supra note 6.
188. Under the present forms of the substantial similarity test, a taking of a copyrighted work must not be trivial nor insubstantial in order to be infringing. But as slight an appropriation as literal duplication of three lines of text may constitute a nontrivial taking depending on the nature of the matter appropriated and the length of the copyrighted work. Henry Holt & Co. v. Liggett & Myers Tobacco Co., 23 F. Supp. 302, 304 (E.D. Pa. 1938) (taking of three lines held to infringe the copyright). Thus, to determine whether a taking is nontrivial, one should consider both quantitative and qualitative evidence regarding the material appropriated.
189. See supra note 72 and accompanying text.
190. Even if the evidence is sufficient to meet the "iterative" test, infringement of the copyright would not be found if the copying was "fair use" of the copyrighted work. See generally Williams & Wilkins Co. v. United States, 497 F.2d 1345, 1350-63 (Ct. Cl. 1973) (discussing the "fair use" doctrine both generally and specifically in the context of photocopying of articles in medical journals by a library for use by a governmental research organization), aff'd per curiam by an equally divided Court, 420 U.S. 376 U.S. 376. Fair use is an equitable doctrine that was created by the courts, see University City Studios, Inc.
A. DESCRIPTION OF THE ITERATIVE SUBSTANTIAL SIMILARITY TEST

The first step of the test, defendant’s use of the copyrighted work, is essentially the same as the first step of the Arnstein approach. The defendant’s access to the copyrighted work and similarities between the two works can be offered to establish use of the copyrighted work. If the copyrighted program were marketed to the general public, access to the work would be almost indisputable because the defendant would have had the opportunity to view the work. The trier of fact in such a case would only need to find sufficient similarities to “reasonably infer” use of the copyrighted work. Either general similarities, such as similar functions, methods, and outputs, or specific similarities, such as buried copyright notices or duplicate bytes could be used by the plaintiff to establish the defendant’s use of the copyrighted work. As under the Arnstein approach, the trier of fact would weigh the evidence according to its probative value in establishing defendant’s use of the work, not by the effect on the ordinary observer. If access to the copyrighted work cannot be established, however, the plaintiff could still establish use by showing that the works are “so strikingly similar as to preclude the possibility of independent creation.”

Assuming that the “use” requirement is satisfied, the fact-finder must then determine whether the use constitutes an “iterative” reproduction. This “iterative” determination accounts for the differences between computer programs and other copyrighted works by reformulating the idea/expression distinction of the substantial similarity test in two ways. First, the test changes the focus of the idea/expression inquiry from the effect of similarities on the “ordinary observer” to the existence or nonexistence of a certain fact as determined by the fact-finder views the same evidence as it would under the substantial similarity standard but the plaintiff is required to meet a higher standard of proof.

---


191. See supra notes 92-95 and accompanying text.

192. See supra note 88 and accompanying text.

193. Ferguson v. National Broadcasting Co., 584 F.2d 111, 113 (5th Cir. 1978); see also 3 M. Nimmer, supra note 36, at § 13.02[B]. The striking similarity standard is also explicitly mentioned by the Arnstein court, which implied that if access is not shown, a higher standard of similarities is required. 154 F.2d at 468.

194. See supra note 91 and accompanying text.
The question for the factfinder is whether it is more probable than not that the defendant created an iterative reproduction of a substantial part of the copyrighted work. In determining this question, the factfinder must observe specific similarities or differences that establish or refute iterative copying, rather than attempt to achieve an intuitive sense of the degree of similarity between the programs. In addition, expert testimony should be encouraged from both the plaintiff and the defendant regarding the defendant's "iterative" use of the copyrighted program. The plaintiff should not only provide expert evidence of specific similarities, but also present some evidence linking these similarities to the probability that the defendant made a literal reproduction or translation.

Second, and more important, the iterative standard clarifies the distinction between the "ideas" and "expression" of a computer program and restricts the definition of "expression" in

---

195. The abandonment of the ordinary observer standard for judging similarities in copyright infringement cases is not unprecedented. Nathan Burkan once successfully argued that the court must "have a more Olympian viewpoint than the average playgoer" to judge an infringement action. Sheldon v. Metro-Goldwyn Pictures Corp., 7 F. Supp. 837, 842 (S.D.N.Y. 1934), rev'd, 81 F.2d 49 (2d Cir.). In fact, the substantial similarity test in practice "is much modified; its contours depend on the subject matter of the copyright and a number of other factors." Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124, 131 (E.D. Mich. 1979). Thus, creating a modified version of the substantial similarity test exclusively for computer programs would not be a novel development for the judiciary in the copyright area.

The change from an ordinary observer analysis to a question of fact as determined by the factfinder resembles somewhat the movement from negligence to strict liability in tort. In both instances a fictitious person standard is replaced by a determination of a certain factual event. As one court stated in a strict liability tort case:

[We] reject standards based upon what the "reasonable" consumer could be expected to know, or what the "reasonable" manufacturer could be expected to "foresee" about the consumers who use his product. . . . Rather, the sole question here is whether the seller accompanied his product with sufficient instructions and warnings so as to make his product safe.

Berkebile v. Brantly Helicopter Corp., 462 Pa. 83, 101, 337 A.2d 893, 902 (1975). Likewise, under this Note's proposal for the copyright area, the question whether the alleged copy is an iterative reproduction (a factual question) replaces the question whether the ordinary observer regarded the alleged copy as substantially similar.

196. Of course, the standard depends on the particular procedural posture of the legal action. In a preliminary injunction action, the plaintiff is required to show only a reasonable probability of success on the merits. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1245-46 (3d Cir. 1983), cert. denied, 104 S. Ct. 690 (1984).

197. For example, expert evidence regarding the number of possible different ways to program a compatible input format would be helpful in deciding whether 50% similarity in object code between two programs constitutes formidable evidence of "iterative" copying.
the context of computer programs. Only translations, such as creating an object code copy of a source code program by use of an assembler or compiler, and literal restatements of the program would constitute infringement under the iterative test. This does not mean that a pirate could avoid the copyright laws by cosmetically changing one small part of a copyrighted program and reproducing the rest. Rather, the plaintiff must only show an iterative reproduction of a substantial portion of the copyrighted work. Thus, mere cosmetic changes would not render the “pirating” program noninfringing since a substantial part of the copyrighted program would still be translated or literally copied. The determination of whether an alleged copy constitutes an iterative reproduction of a substantial portion of a copyrighted program should be made by examining both quantitative and qualitative evidence of similarities. Quantitative evidence could consist of showing the percentage of bytes in the copyrighted program that were duplicated in the alleged copy. Qualitative evidence could consist of expert testimony regarding the meaning of the quantitative similarities in light of the functions the program serves. For example, for purposes of determining iterative copying, duplication of the input formats of a copyrighted program is of less significance than duplication of the operating procedures of the program.200

Obviously, the iterative standard does not produce a final resolution of the idea/expression distinction. Under the iterative standard, the factfinder still must determine whether the evidence demonstrates a restatement in substantially the same

198. A restricted definition of expression has been applied in other copyright areas. For example, a copyrighted biography of a famous person is not infringed by another biography that portrays the same events and relies heavily upon the copyrighted work. Rosemount Enterprises, Inc. v. Random House, Inc., 368 F.2d 303, 306-07 (2d Cir. 1966), cert. denied, 388 U.S. 1009 (1967). A more general distinction is made in Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124, 131 (E.D. Mich. 1979), between works of a commercial and useful character on one hand and literary works on the other hand. The Kepner-Tregoe court stated: “Two useful works might bear resemblance that would show infringement if literary, but which is not because they are technical. The ‘substantial similarity’ test is modified.” Id. Thus, creating a new, restricted definition of substantial similarity for computer programs would not be an unprecedented occurrence in copyrighted law. The Kepner-Tregoe court views this as “thin” copyright protection. Id. at 130-31.

199. See supra note 15 and accompanying text.

200. The input formats are not as substantial a part of the computer program as are the total operations of the program. Further, certain input formats may be required by the computer in its operation. Thus, 100% duplication of the input formats probably would not be considered an iterative copy of a substantial part of the copyrighted program, whereas 70% duplication of the operating program’s bytes could indicate an iterative reproduction.
form of a substantial portion of the copyrighted program. The continuum between the idea of the computer program and its expression still is present.\textsuperscript{201} Focusing on the literal similarities between the two works and their significance in the alleged pirate's program, however, will yield a more clearly defined and ascertainable distinction between the idea and its expression.

B. THE CHANGES AND EFFECTS OF THE ITERATIVE SUBSTANTIAL SIMILARITY TEST

The differences between the iterative standard and the present forms of the substantial similarity test become clearer by reexamining the two hypothetical computer infringement problems previously discussed. In the first hypothetical, where 80\% of the object code was duplicated and a buried copyright notice appeared in the alleged copy,\textsuperscript{202} iterative copying would be found. The plaintiff could clearly demonstrate access;\textsuperscript{203} and the buried copyright notice alone provides a reasonable inference that the defendant used the copyrighted work in preparing the alleged copy.\textsuperscript{204} Proceeding to the second, "iterative" inquiry, both the expert's testimony that 80\% of the entire program's object code is duplicated in the copy (supplemented by an explanation of the alternative ways to produce a program with the same functions) and the buried copyright notice provide overwhelming evidence that the defendant made a literal reproduction of a substantial portion of the copyrighted program, thus establishing iterative copying.\textsuperscript{205} Therefore, in contrast to the exclusion of expert testimony under the present substantial similarity tests,\textsuperscript{206} the plaintiff could satisfy both

\textsuperscript{201} See supra notes 59-63 and accompanying text. The progression from idea to expression in computer programs can be expressed linearly as follows:

```
EXPRESSION  \\
Literal  Mechanical  Cosmetic  Structural  Reverse  Independent  \\
Copy  Translation  Changes  Changes  Engineered  Creation
```

The iterative test attempts to draw a line between cosmetic changes to a literal copy and structural changes in the use of a copyrighted work. Obviously, a fine line cannot be drawn; however, a better definition of an infringing reproduction and a more structured analysis of the infringement question can result from using the iterative approach.

\textsuperscript{202} See supra notes 152-55 and accompanying text.

\textsuperscript{203} This is a mass-marketed program, thus giving the opportunity to copy, and thereby satisfying the access requirement. See supra note 88 and accompanying text.

\textsuperscript{204} See supra note 147.

\textsuperscript{205} In other words, it is more probable than not that the defendant created a literal reproduction of the copyrighted work.

\textsuperscript{206} See supra notes 152-55 and accompanying text.
steps of the iterative approach, and thus prove infringement, with the relevant expert testimony.

Returning to the second hypothetical,207 where a programmer reverse engineered a program similar to a copyrighted program,208 the iterative standard would not allow a finding of infringement. While the similar input formats209 and program functions probably demonstrate use of the copyrighted program by the defendant and thus satisfy the first step,210 the evidence falls short of meeting the iterative copying requirement. A reasonable factfinder could not find that the defendant literally reproduced any part of the copyrighted work simply because the alleged copy generally resembled and served the same purpose as the copyrighted program. Some type of specific evidence regarding similarities in the form of the programs, such as duplicate commands, object code, and similar errors, would be necessary for a finding of iterative reproduction. Thus, unlike the present substantial similarity standards, the iterative approach would classify the reverse engineered program as a noninfringing use of the copyrighted program.

By using the iterative standard, courts would avoid both of the problems resulting from application of the "abstractions" and *Arnstein* forms of the substantial similarity test to computer programs.211 As is obvious from the hypotheticals, the iterative approach eliminates the ordinary observer standard and allows the use of expert testimony to show similarities of expression. The new test substitutes an objective factual determination for the intuitive decision of the ordinary observer. The trier of fact focuses on finding the existence of iterative copying as it would view any other factual issue in a copyright infringement trial. The iterative standard also recognizes the need for expert testimony to show similarities. By allowing expert testimony and focusing on specific similarities, the iterative test produces a more applicable and realistic standard to judge similarities between computer programs.

The "iterative" substantial similarity test also adjusts the idea/expression distinction to allow more cumulative innova-

---

207. *See supra* notes 178-84 and accompanying text.
208. *See supra* note 108.
210. The defendant viewed the copyrighted work and based his or her program on that work, thus meeting the first step of the "iterative" test.
211. As discussed earlier, *see supra* notes 140-84 and accompanying text, application of the substantial similarity test to computer programs is inappropriate because computer programs lack ordinary observers or audiences and because it would result in overprotection of copyrighted computer programs.
tion and thereby foster advancement in the computer software field. By focusing on the literal similarities in copyrighted programs, the test allows reverse engineering\textsuperscript{212} and second sourcing\textsuperscript{213} of copyrighted programs. A programmer may plagiarize a computer program to produce an innovation as long as he or she does not literally reproduce a significant portion of the copyrighted work. Further, since only copying of a significant portion of the copyrighted program is prohibited, literal duplication of the input formats of a copyrighted program would in many cases be noninfringing. The iterative approach allows programmers to use the copyrighted work to a greater extent than do the present forms of the substantial similarity test. Thus, the iterative test promotes dissemination of the copyrighted work’s ideas and advances the policy of the copyright laws to promote progress.\textsuperscript{214}

The iterative test also retains the basic protection necessary to achieve the incentive to create new software.\textsuperscript{215} The test clearly prohibits literal reproductions of copyrighted programs, thus providing protection against slavish copying. And since reverse engineering or second sourcing of a copyrighted program takes considerable time,\textsuperscript{216} the copyright holder would have a significant time advantage over any imitators of the program. With computer software progressing so rapidly that programs become obsolete within months of their development,\textsuperscript{217} the copyright holder’s time advantage would appear sufficient to provide the necessary incentives to create. Indeed, in the 1960s the total lack of proprietary protection for mathematical programming was credited with producing harder work, new in-

\textsuperscript{212} See supra note 108.
\textsuperscript{213} See supra note 175.
\textsuperscript{214} Some courts have implied that the promotion of progress by cumulative innovation of copyrighted works is not an appropriate policy of the copyright laws. See, e.g., Toksvig v. Bruce Publishing Co., 181 F.2d 664, 666-67 (7th Cir. 1950). Yet, one commentator has said:

It is unreasonable to say, as some courts have, that the copyright act bars one person from saving labor and time by using the works of another. On the contrary, avoidance of wasteful duplication of effort is one of the policies underlying the law of copyright.

Gorman, supra note 111, at 1604 (footnotes omitted).

\textsuperscript{215} Recall that the copyright laws are designed to give only a limited monopoly to the author in order to encourage new works. See supra note 69 and accompanying text.

\textsuperscript{216} See supra notes 161-75 and accompanying text.

\textsuperscript{217} One commentator suggests that computer software progresses so fast that “any program which is running is obsolete.” Kellam, The Future of Computing: Personal with a Capital P, PERSONAL COMPUTING, May 1983, at 211, 214.
novations, and rapid dissemination of the improvements.\textsuperscript{218} Certainly in the 1980s computer programs should be given some proprietary protection. The scope, however, of the protection should be balanced with the need for disseminating information in order to advance the computer field. The iterative test provides a reasonable approach to balancing the interests of the programmer with the needs of society.

CONCLUSION

Copyright protection of computer programs has recently expanded in scope. Consequently, infringement actions have become more prevalent and courts have become confused about the appropriate standard of analysis to apply in determining infringement. The substantial similarity test under either the "abstractions" or the \textit{Arnstein} approach provides a poor framework to judge similarities and would lead to stifling innovations of programs if it were applied. An iterative standard of substantial similarity has been suggested by this Note to correct these problems. The iterative test changes the current forms of the substantial similarity test with a factual determination of whether the alleged copy is a restatement in substantially the same form of a substantial portion of the copyrighted program. By refocusing the inquiry, the iterative test provides a clearer standard to judge infringement of copyrighted computer programs, and furthers the policies of copyright law by promoting dissemination of ideas resulting in more cumulative innovation. Thus, while no test will completely solve the problem that Learned Hand described as unsolvable,\textsuperscript{219} the iterative approach improves upon the present confusion and provides a better alternative.

\textit{Howard Root}

---

\textsuperscript{219} Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930).