




2014

Forensics of Software Copyright Infringement Crimes: The Modern POSAR Test Juxtaposed With The Dated AFC Test

Vinod P. Bhattathiripad
GJ Software Forensics

Follow this and additional works at: <https://commons.erau.edu/jdfsl>

 Part of the [Computer Engineering Commons](#), [Computer Law Commons](#), [Electrical and Computer Engineering Commons](#), [Forensic Science and Technology Commons](#), and the [Information Security Commons](#)

Recommended Citation

Bhattathiripad, Vinod P. (2014) "Forensics of Software Copyright Infringement Crimes: The Modern POSAR Test Juxtaposed With The Dated AFC Test," *Journal of Digital Forensics, Security and Law*. Vol. 9 : No. 2 , Article 7.

DOI: <https://doi.org/10.15394/jdfsl.2014.1172>

Available at: <https://commons.erau.edu/jdfsl/vol9/iss2/7>

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.



(c)ADFSL





FORENSICS OF SOFTWARE COPYRIGHT INFRINGEMENT CRIMES: THE MODERN POSAR TEST JUXTAPOSED WITH THE DATED AFC TEST

Vinod Polpaya Bhattathiripad
Cyber Forensic Consultant
GJ Software Forensics
Kozhikode - 673004, Kerala, India
vinodpolpaya@gmail.com

ABSTRACT

This paper presents a new development in the forensics of software copyright through a juxtaposed comparison between the proven AFC test and the recent POSAR test, the two forensic procedures for establishing software copyright infringement cases. First, the paper separately overviews the 3-stage, linear sequential AFC test and then the 5-phase, cyclic POSAR test (as AFC's logical extension). The paper then compares the processes involved in each of the 5 phases of the POSAR test with the processes involved in the 3 stages in the AFC test, for the benefit of forensic practitioners and researchers. Finally, the paper discusses some common areas where both the tests will need careful handling while implementing them in the judiciaries across the world.

Keywords: AFC, Abstraction-Filtration-Comparison, POSAR, software copyright, infringement, Altai

1. INTRODUCTION

The copyright laws of many countries (for example, the U.S. and UK) have classified the elements of software as literal and non-literal. Copyright protection extends not only to the literal elements of a computer program (its source and executable codes) but also to the program's non-literal elements (structure, user interfaces and menu command hierarchy) as well. In addition, copyright laws of most countries extend the legal validity of this binary classification to the *act of copying* as well. Literal copying is an act which is almost similar to the act of copy-paste. On the other hand, non-literal copying refers to when the copyrighted ideas in software appears elsewhere unauthorized even if in a different expression. Violation of copyright can be a

result of either literal or non-literal copying of both literal and/or non-literal elements in software.

One of the leading forensic methods to establish non-literal copying of either literal or non-literal elements is the Abstraction-Filtration-Comparison (AFC) test (USCA2C, 1992). The POSAR test (Bhattathiripad, 2014), a recently devised forensic procedure for establishing software copyright infringement cases, is an extension or an enhancement of the AFC test. POSAR, with its added features and additional facilities, offers something more to the legal and the judicial domain than what the AFC test offers. These additional features and facilities make the test more sensitive to the technical

and legal requirements of software copyright infringement.

The objective of this paper is to perform a juxtaposed comparison of the POSAR test with the proven AFC test and to demonstrate how OSAR provides a legitimate developmental extension to AFC¹.

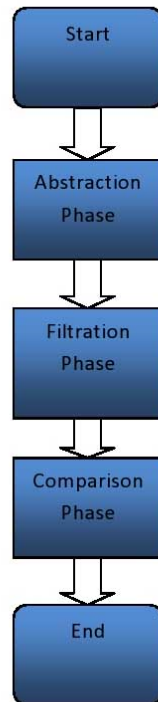


Figure 1 Process Flow Chart of the AFC Test's Linear-Sequential Process (Bhattathiripad, 2014)

¹ My sincere gratitude to Dr. P. B. Nayar, Lincoln University, UK, for his help in articulating my thoughts.

2. OVERVIEW OF THE AFC TEST

The AFC test, a 3-phase forensic test (see Figure 1) for establishing copyright infringement cases, was first enunciated in 1992 by the Second Circuit of the US judiciary, in the case *Computer Associates v. Altai*, popularly known as the Altai case (USCA2C, 1992). This test holds good for both the literal elements and the non-literal elements of the software and also for both literal and non-literal copying of software elements. Also, this test has a strong base in the US copyright law. Ever since it has been legally validated in the Altai case, the AFC test has been put to use for evaluating copyright infringement claims involving computer software in several appeal courts in the United States, including the fourth, tenth, eleventh and federal circuit courts of appeals (ESALab, 2007; USCA2C, 1997; USDCM, 2010).

The procedural approach of the AFC test draws on familiar copyright doctrines such as merger, *scènes à faire*, and public domain. These three doctrines define three categories of programming elements. The AFC test procedurally filters out and excludes these three categories of programming elements from both the plaintiff's and the defendant's software before the two sets of software are finally compared (Walker, 1996).

The AFC test requires the investigator to:

1. abstract the software by breaking down the plaintiff's as well as defendant's program into their constituent structural parts;
2. examine and filter out the three categories of programming elements defined by the three doctrines mentioned above as well as some other unprotectable elements, all from both the software packages with the ultimate goal of preparing a set of two comparable kernels of creative expressions; and

3. compare the remaining kernel of creative expressions, if any, of the plaintiff with that of the suspect program, at each level of abstraction.

Despite its popularity in the US judiciary, the comprehensive applicability or at least the nature of implementation of the AFC test as an infallible forensic instrument in court cases involving computer software has already come under skepticism and consequent re-appraisal. The existing form of the AFC test has been shown to have some functional, technical and procedural deficiencies enumerated below (Baboo and Bhattathiripad, 2009, 2010; 2012a, 2012b; Bhattathiripad, 2014; Kremen, 1998; Zeidman, 2011).

One of the biggest deficiencies of the AFC test is its inability to view software piracy investigation as a cyclical or spiral process. Secondly, the AFC test does not seem to have elaborated on the implications on one important factor in forensic software investigation, namely, post-piracy modifications, which are potentially relevant. Thirdly, the AFC test does not yet have a final stage that stresses the need for presenting the results of the whole investigation in a judiciary-friendly, jargon-free, and non-esoteric format (preferably in quantitative/numerical terms). The fourth limitation of AFC is the weakness of its prevailing list of areas of investigation for abstraction and filtration. The fifth limitation of AFC is its inability to deal with certain design and programming patterns like programming blunders. Finally, the AFC test does not presuppose or suggest a contingency to consider evidence external to the software in hand.

3. OVERVIEW OF THE POSAR TEST

POSAR stands for Planning-Operationalization-Separation-Analysis-Reporting. While the AFC test has been in use since 1992 in the US judiciary, the POSAR test, even though an extension of the AFC test, is quite a recent one and is yet to

be introduced into any judiciary system. The POSAR test has five phases and they are: (1) Planning phase; (2) Operationalization phase; (3) Separation phase; (4) Analysis phase and; (5) Reporting phase (see Figure 2).

The procedural approach of the POSAR test also draws on familiar copyright doctrines such as merger, *scènes à faire*, and public domain.

The POSAR test requires the investigator to:

- (1) abstract the software by breaking down the plaintiff's as well as defendant's program into their constituent structural parts along a specific list of the general areas of investigation and also by taking into account the forensic importance of factors like programming remarks, programming blunders, programming errors, sequence of appearance of similarly looking items, database field properties, program manuals and documents;
- (2) examine and filter out the unprotectable elements (for example, the three categories of programming elements defined by the three doctrines, mentioned above) from both the software packages with the

ultimate goal of preparing a set of two filtered abstractions;

- (3) separate the suspected post-piracy modifications from these two filtered abstractions with the ultimate objective of preparing three separate sets of comparable items (like the comparable creative expressions, other comparable, contributing elements that can add credibility to the results of comparison etc.) of filtered abstractions;
- (4) compare the remaining creative expressions, if any, of the plaintiff's program with that of the defendant's program, at each level of abstraction (with the ultimate objective of enlisting the apparent similarities and commonalities) and also to analyze other contributing elements (with the ultimate objective of enlisting the pieces of evidence of copyright infringement of specific thumb impressions, programming errors, programming blunders etc.); and
- (5) prepare the forensic report in specific judiciary-friendly formats.

In addition, by being cyclic, the POSAR test procedure finds a way to tackle an error that may have happened in any previous stage of the forensic process.

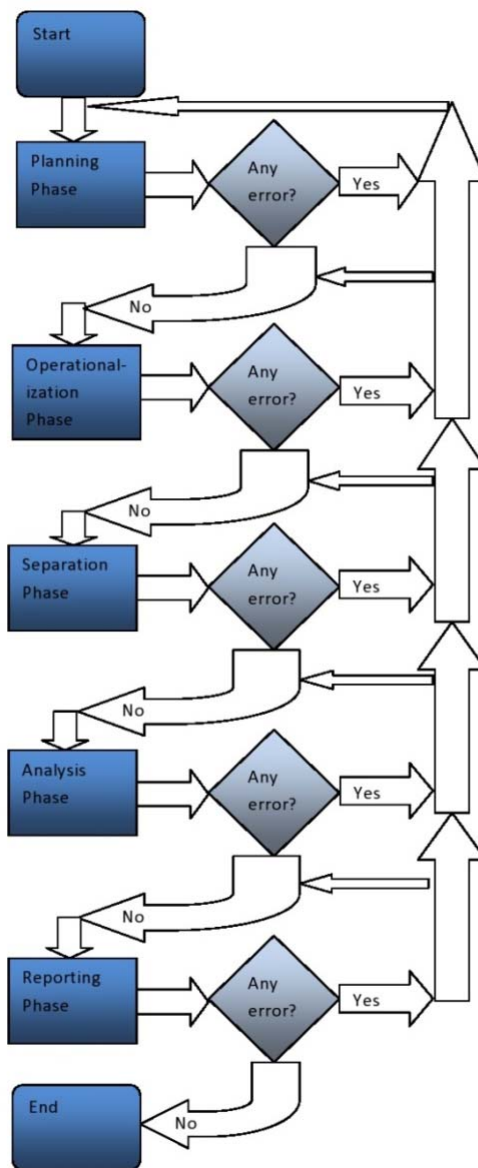


Figure 2 Process Flow Chart of the POSAR Test's Forensic Cycle (Bhattathiripad, 2014)

4. SIMILARITIES, COMMONALITIES AND DIFFERENCES

From the above, it should be clear that both AFC and POSAR are forensic test procedures used to investigate software copyright infringement cases and that the approaches of the AFC as well as POSAR test draw on familiar copyright doctrines such as merger, *scènes à faire*, and public domain. Both AFC

and POSAR tests specify that these three categories (defined by the three doctrines, mentioned above) of programming elements need to be excluded from the software before the sets of software are compared (Bhattathiripad, 2014; Walker, 1996). In addition, both AFC and POSAR tests recognize that, apart from the above three categories, a fourth category of exclusion exists specifically to address the universal facts that are used in the program.

In general, both AFC and POSAR tests first abstract the two software systems, then filter out the unprotected elements from these two abstracted systems, and finally compare the residual protectable elements before submitting the forensic report to the court. Nevertheless, the procedural steps involved in these two tests vary in varying degrees in the various stages/phases of the two tests.

5. ABSTRACTION PROCEDURES IN AFC AND POSAR

The abstraction step of the AFC test is generally performed in accordance with the US judiciary Tenth Circuit elaboration that "...a computer program can often be parsed into at least six levels of generally declining abstraction: (i) the main purpose; (ii) the program structure or architecture; (iii) modules; (iv) algorithms and data structures; (v) source code; and (vi) object code" (USCA10C, 1993). The planning phase of the POSAR test too abstracts the two sets of software generally in accordance with the (above given) US judiciary Tenth Circuit elaboration. Both AFC and POSAR allow: (1) that the abstraction need not be performed in any one specific way and that different ways of abstractions can be considered legal and valid so long as what is left over after abstraction is operational in the subsequent stages/phases of the tests; (2) that the above mentioned 6 levels of abstraction are just for the guidance of the experts and the expert designated to investigate a particular case of copyright infringement can even convincingly suggest more levels of abstraction to the judiciary; and (3) that most judiciaries would welcome such suggestions as helpful guidance. (These three are allowances and not presumptions.)

Although the abstraction stage of the AFC test as well as the planning phase of the POSAR test abstracts the two sets of software systems, the planning phase of the POSAR test (unlike the abstraction stage of the AFC) has specific five objectives and

rationale and they are: (1) To make the POSAR test a team-event; (2) To ensure that all the forensic areas are covered; (3) To ensure that all the functional areas in the software are covered; (4) To ensure that all the software parts are covered; and (5) To ensure that nothing that could potentially be evidence is filtered out. To make it clearer, unlike the AFC test, the POSAR test recommends that the process of abstraction requires further elaboration (of these six levels) and needs to be performed along a prefixed list of areas of investigation (see Bhattathiripad, 2014), for this list). Essentially, the planning phase of POSAR requires that factors like programming remarks, programming blunders², programming errors, sequence of appearance of similarly looking items, database field properties and also that programming documents like program manuals also need to be taken into account during the process of abstraction. Thus, unlike AFC, POSAR ensures that the pieces of evidence related to the similarities and commonalities in programming remarks, programming blunders, programming errors, sequence of appearance of similarly looking items, database field properties, program manuals and other related documents are retained and not abstracted away.

The abstraction stage of the AFC test generates six documents (on each of the two software systems), which form the basic input to the next stage. Although the planning phase of the POSAR too generates six documents (again, on each of the two software systems), they are generated along the pre-

² A programming blunder (Bhattathiripad, 2012a) or a "copied misbehaviour" (Zeidman, 2011) is often sufficient to support the complaint (Hollaar, 2002) and the POSAR test, unlike the AFC test, ensures that this valuable evidence stays for final comparison. Code segments which are eligible to be called programming blunders include not only the dead codes, and the non-executable execution paths but also some of the deliberately inserted identity stamps (like watermarks).

fixed list of areas (specifically defined by the POSAR protocol), and these six documents form the basic input to the next phase of the POSAR test.

Both AFC and POSAR believe that these six documents (or the results of the six levels of abstractions) require subsequent forensic treatments in varying degrees to make them conformable to the standards of the copyright law (of the particular country) and intelligible to the judiciary so as to help the judicial system to take the final decision.

In general, the outputs of both the abstraction stage of the AFC test and the planning phase of the POSAR test are the two raw sets of items which can be taken to the next phase. These two raw sets are further refined in the filtration stage (in case of the AFC test) and operationalization and separation phases (in case of the POSAR test) to turn them into two sets of fully comparable items.

6. FILTRATION STAGE OF THE AFC TEST JUXTAPOSED WITH THE OPERATIONALIZATION AND THE SEPARATION PHASES OF THE POSAR TEST

Both the plaintiff's and the defendant's software can look similar when both of them graduate through the same educational system, or have used the same implementation technique from a particular text book or have strictly followed the same international programming standards. Thus, any software generally contains elements that are universal and global and such elements can automatically create similarity when this software is compared with any other software in the same functional area. Such similarities need to be set aside and filtered out in order to properly identify the similarities that can be assigned to copyright infringement. The filtration stage of the AFC test as well as the

(corresponding) operationalization phase of the POSAR test tries to address this issue. Both the AFC test and the POSAR test do this by removing the following 4 types of global elements (Walker, 1996; USCA2C, 1992): (a) elements that are dictated by efficiency considerations (idea-expression merger considerations); (b) elements whose use are dictated by external factors such as functional area, calling sequence for a library routine, operating system function etc.; (c) elements formed out of materials in the public domain; and (d) elements that are facts (except the particular selection or arrangement of facts), all according to the suggestions put forward by the US judiciary. However, the POSAR test maintains that the filtering out of the above needs to be sensitive to two very important elements and these are: (i) all suspected post-piracy modifications and; (ii) all considerations related to design and programming pattern (like programming blunders etc.). This means that the POSAR test (unlike the AFC test) ensures that those elements which are either suspected post piracy modifications or whose design aspects can in some way provide supporting evidence to establish piracy are not filtered out.

The outputs of both the filtration stage of the AFC test and the operationalization phase of the POSAR test are a set of two filtered abstractions. While the AFC protocol believes that these two sets are ready for final comparison, the POSAR protocol believes that these are not yet ready for final comparison and should necessarily go through another phase for separating post-piracy modifications, thereby creating a pre-modified (infant) version of both these filtered abstractions (in order to facilitate a more reliable final comparison).

The separation phase of the POSAR test deals with the suspected post-piracy modifications in the above two filtered abstractions. Post-piracy modifications cause apparent surface differences that would decrease the similarity and thus favor the defendant or weaken the evidence of copying,

if any (Baboo and Bhattathiripad, 2011; Bhattathiripad, 2014). By eliminating these differences, the ‘visibility’ of actual similarities in the ideal ‘original’ and the ‘pirated’ will be enhanced. In order to increase this visibility, the idealized forms (forms at the time of the alleged copyright infringement) of both versions of the software need to be obtained by first identifying and marking the suspected post-piracy modifications (potential evidence) and then, purging these marked elements from the two versions of the software. The task of identifying, marking and purging of post-piracy modifications is not a simple task and might require many pieces of external data (for example, log book of software development/maintenance, dates and content of government directives, say, on tax tariff restructuring etc.) to provide evidential support for the purging.

The output of the separation phase of the POSAR test is in three sets, each containing two separate and comparable sub-sets of code segments, one for the complainant’s and the other for the defendant’s code. They are as follows:

- a) The first set contains two comparable sub-sets of source codes.
- b) The second set also contains two comparable sub-sets of source codes but these source codes are the idealized or pre-modified forms (of the first set, mentioned under ‘a’ above) generated after purging the suspected post-piracy modifications.
- c) The third set is derived from the first two. This set contains two separate and non-comparable lists of suspected post-piracy modifications in the two software systems. This third set, not a class similar to the first two sets, is not used for comparison but for general analysis and reporting. The primary objective of creating this set and then performing this analysis is for adding credibility to the results of

comparison of the elements of first two sets.

The above three sets are the inputs to the next phase (the analysis phase) of the POSAR test for comparison and analysis.

7. THE COMPARISON STAGE OF THE AFC TEST AND THE ANALYSIS PHASE OF THE POSAR TEST

The comparison stage of the AFC test, and the corresponding analysis phase of the POSAR test are concerned with the final comparison of the comparable parts of the plaintiff’s and the defendant’s software systems. The result of this comparison will determine whether the protectable elements of the two sets of programs at issue are substantially similar so as to warrant a verdict of copyright infringement.

In this stage, the POSAR can claim to be functionally more efficient than the AFC. While the comparison stage of AFC will do only a juxtaposed comparison of the comparable elements, the equivalent analysis phase of the POSAR test achieves not just that, but in addition, also analyzes the suspected post-piracy modifications and the other marked elements (thumb impressions, programming errors, programming blunders etc.). The result of this extended analysis may well yield valuable supporting evidence and thus contribute to the credibility of the result. The results of the analysis phase of POSAR will be approximately in the following form.

- a) A list of apparent similarities.
- b) A list of similarities from design and programming pattern considerations (for instance, similarities of thumb impressions, programming errors, programming blunders etc. and statistical occurrences, percentages and counts of nomenclature level similarities and commonalities etc.).

- c) A list of similarities arising from suspected post-piracy modifications along with an analysis report of suspected post-piracy modifications in the pirated.

These lists are separately made over as inputs to next phase (the reporting phase) of the POSAR test.

8. AFC'S UNSTRUCTURED AND POSAR'S STRUCTURED REPORTING STYLES

As the forensic consummation for the whole effort, the AFC test does not call for or provide a structure to collate and present the final results which are available at the end of its comparison phase. Because of the absence of a structured format of collecting the results, the structuring and presentation of the result by experts may lack logical patterning and expert-independent consistency (Bariki, Hashmi, and Baggili, 2010). Unlike the AFC test, the POSAR test has a reporting phase which helps and encourages the cyber forensic expert to collect and collate the results concerning similarities and commonalities in a structured manner, statistically as well as verbally. As a result, the cyber forensic expert will be able to easily prepare his/her expert report in a systematic and methodical manner. This will certainly protect against accidental oversight of vital evidence and/or possible loss of it in an unstructured clutter of elements. Also, because the report format is structured and consistent, the leader of a group of cyber forensic experts can easily consolidate the findings of the member experts in the group and thus, unlike the AFC test, the POSAR test not only enables but also capitalizes on group tasking in software copyright infringement forensics. Finally, because the report format is structured and consistent, the report becomes inherently more intelligible to the judiciary.

While the AFC as well as the POSAR tests can both claim a conclusive and rigorous final decision on the merit of the copyright infringement allegation, the POSAR test additionally backs up the expertise with visible, intelligible and apparently credible evidence and with higher degree of objectivity. While the AFC test often demands the judiciary's reliance almost entirely on the credibility of the experts, the POSAR test's additional way of presenting certain pieces evidence quantitatively (for example, the percentage similarity in nomenclature) can perhaps additionally help the judiciary feel that their decisions may well have been made on objective tangible evidence universally accessible in non-esoteric form than mere reliance on the credibility of expert. This objectively documentary aspect of the results of POSAR makes it more forensically powerful.

In addition to this, the POSAR test tries to convincingly demonstrate the relative priorities between different types and pieces of evidence in terms of their role in establishing copyright infringement. Even if there is 95% similarity between, say, two globally available (or so, to-be-filtered-out) modules, the POSAR test presents them to the judiciary in the priority that they deserve. In the presentation of the results, the order of priorities will also thus be more transparent and intelligible to the judges.

9. THE POSAR ADVANTAGE

In general, the five-phase POSAR test can draw upon and bring out more pieces of evidence of software copyright infringement than the 3-stage AFC test. (See Bhattathiripad (2014), for some lists of features of the POSAR test, some of which have roots in the AFC test., and also for the test's empirical validation). The POSAR test's comprehensibility as well as advantage (over the AFC test) has already been proved using some artificial as well as live test cases. These test results show that not only has the

POSAR test never fallen short of the AFC test in any way but has, in addition, brought out more pieces of evidence.

10. A NOTE ON CYCLIC POSAR AS AGAINST THE LINEAR SEQUENTIAL AFC

One of the biggest achievements of the POSAR test is its ability (unlike the AFC test) to consider software piracy forensics as a cyclical process. By introducing cyclicity, the POSAR test finds a way to tackle an error that may have happened in any previous stage of the forensic process. For instance, if an error in abstracting the software was found in the Separation phase, the cyclical nature of the POSAR test helps the cyber forensic expert to return to the Planning phase (where the software abstraction was done) for necessary correction and then proceed to the subsequent phases for a possible modification of all the subsequent abstractions, filtrations and separations.

In addition, the POSAR test, with its cyclical and algorithmic nature, is helpful in carrying out the software copyright infringement forensics as a team event, unlike the AFC test, which is non-cyclical, not as algorithmic as POSAR and so, is more of a single individual effort. (The judiciary is most likely to find a team-effort with multiple expertise at work more credible and reliable than a single-expertise effort, particularly in the case of large software systems which can run into millions of codes spanning hundreds of files and databases). Moreover, the cyclical, algorithmic POSAR test can make the forensics management more controllable, which may result in better reliability in the integrity and quality of the forensic result apart from making the related risk-management easy. The forensics management would then be more controllable, and ultimately, the integrity and quality of the forensics result, more reliable.

11. COMMON SNAGS IN THE USE OF BOTH TESTS

A major snag in the use of both AFC and POSAR at present is that while the former has *de facto* status in North American judiciary, neither the former nor the latter is a fully legally enforced *de jure* process in any country. They both need to be ‘customized’ properly in the sense that they need to conform or be suitably aligned to the existing copyright laws of the country.

Another major snag in the use of both AFC and POSAR at present is that they both are not well tuned for certain specific forensic needs of digital watermarks, digital image files, external sources of data (other than data bases and data files), hand shaking routines in embedded systems etc.

Again, neither the AFC test nor the POSAR test deals with external evidence. As things stand, both these tests do only actual comparison of two software systems and there is scope for incorporating the role and function of (the external evidence of) evident copyright infringement of data as part of establishing software copyright infringement.

12. CONCLUSION

The 5-phase, cyclic POSAR test is an extension or an enhancement of the 3-stage, linear sequential AFC test. The POSAR test has been formulated keeping in mind possible weaknesses and deficiencies of the AFC test. Without discounting the AFC test, the POSAR test improves on it, performs the task more effectively, efficiently and reliably (from the expert’s point of view), and presents the results more clearly and convincingly to the judiciary. The POSAR test backs up the expertise with visible, intelligible and apparently credible data evidence with high degree of objectivity. Nevertheless, both might require customization before their actual judicial implementation in various countries. Also, either of them does not deal with external evidence.

REFERENCES

- USCA2C. (1992). United States Court of Appeals, Second Circuit, Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 1992 U.S. App. LEXIS 33369; 119 A.L.R. Fed. 741; 92 Cal. Daily, Op. Service 10213, January 9, 1992, Argued on December 17, 1992.
- Bhattathiripad, P. V. (2014). Judiciary-friendly forensics of software copyright infringement, IGI-Global.
- ESALab. (2007). European Software Analysis Laboratory's "SIMILE Workshop": Automating the detection of counterfeit software, available at www.esalab.com
- USCA2C. (1997). United States Court of Appeals, Second Circuit, Corrected Amicus Brief filed by Mark M. Arkin, Counsel of 5 computer scientists regarding software copyright and trade secret cases – Observation of Abstraction, Filtration and Comparison test, on appeal from the United States district court for the southern district of New York, on the suit Harbour Software Inc. Vs Applied Systems Inc., 97-7197L
- USDCM. (2010). United States District Court of Massachusetts, Memorandum and Order, Civil Action number 07-12157 PBS, Real View LLC. Vs. 20-20 Technologies, p.2
- Walker, J. (1996). Protectable 'Nuggets': Drawing the Line between Idea and Expression in computer Program Copyright Protection, 44, Journal of the Copyright Society of USA, 44(79).
- Baboo, S. S. and Bhattathiripad, P. V. (2009). Software Piracy Forensics: Exploiting Nonautomated and Judiciary-Friendly Technique, *Journal of Digital Forensic Practice*, 2(4), 177-179.
- Baboo, S. S. and Bhattathiripad, P. V. (2010). Software Piracy Forensics: The need for further developing AFC, Second International ICST Conference on Digital Forensics & Cyber Crime, 4-6 October 2010, Abu Dhabi, UAE.
- Bhattathiripad, P. V. (2012a). A Proposal for Incorporating Programming Blunder as Important Evidence in Abstraction-Filtration-Comparison Test, Conference on Digital Forensics, Security and Law, Richmond, VA.
- Bhattathiripad, P. V. (2012b). Software Piracy Forensics: A Proposal for Incorporating Dead Codes and other Programming Blunders as Important Evidence in AFC Test, IEEE 36th International Conference on Computer Software and Applications Workshops, Turkey.
- USCA10C. (1993). United States Court of Appeals, tenth Circuit, Gates Rubber Co. v. Bando Chem. Industries. 9 F.3d 823, 28 USPQ2d 1503
- Baboo, S. S. and Bhattathiripad, P. V. (2011). Software Piracy Forensics: Impact and Implications of post-piracy modification, Conference on Digital Forensics, Security and Law, Richmond, VA, 2011.
- Bariki, H., Hashmi, M., and Baggili, I. (2010). Defining a Standard for Reporting Digital Evidence Items in Computer Forensic Tools, Proceedings of the Second International ICST Conference on Digital Forensics & Cyber Crime, 4-6 October 2010, Abu Dhabi, UAE.
- Hollaar, L. A. (2002). Legal Protection of Digital Information, BNA Books
- Zeidman, B. (2011). The Software IP Detectives's Handbook, NJ: Prentice Hall.
- Kremen, S. H. (1998). Presentation of Technical Evidence by Experts in Computer Related Intellectual Property Litigation, *Computer Forensics*, 2(1), November 1998.

