

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AKAMAI TECHNOLOGIES, INC.,

Petitioner,

v.

EQUIL IP HOLDINGS LLC,

Patent Owner.

Case IPR2023-00330

U.S. Patent No. 8,495,242

DECLARATION OF VIJAY K. MADISETTI IN SUPPORT OF PETITION

FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,495,242

TABLE OF CONTENTS

	Page
I. INTRODUCTION.....	1
II. QUALIFICATIONS.....	3
III. LEGAL UNDERSTANDING.....	11
A. My Understanding of Claim Construction.....	11
B. My Understanding of Anticipation	13
C. My Understanding of Obviousness.....	16
IV. BACKGROUND OF THE TECHNOLOGY	21
V. THE '242 PATENT AND PROSECUTION HISTORY.....	25
A. '242 Patent Overview	25
B. Prosecution History of the '242 Patent.....	26
VI. THE '242 PATENT PRIORITY DATE	27
VII. LEVEL OF ORDINARY SKILL IN THE ART	28
VIII. CLAIM CONSTRUCTION	30
A. Preambles	30
IX. GROUNDS OF UNPATENTABILITY	31
A. Ground 1: Tso in view of Huang Renders Obvious Claim 9.....	32
1. U.S. 6,421,733 (“Tso”) Overview	33
2. U.S. 6,438,576 (“Huang”) Overview and Motivation to Modify Tso with Huang’s Teachings	39
3. Invalidity of Claim 9 Over Tso in view of Huang	51
(a) Element [9.pre]	51
(b) Element [9.a].....	52
(c) Element [9.b]	59
(d) Element [9.c].....	64
(e) Element [9.d]	67
(f) Element [9.e].....	77
(g) Element [9.f]	87

(h)	Element [9.g]	92
B.	Grounds 2 and 3: Samaniego Anticipates and Renders Obvious Claim 9	94
1.	Overview of Samaniego	94
2.	Invalidity of Claim 9 in View of Samaniego.....	95
(a)	Element [9.pre]	95
(b)	Element [9.a].....	96
(c)	Element [9.b]	99
(d)	Element [9.c].....	100
(e)	Elements [9.d]-[9.f].....	103
(f)	Element [9.g]	107
C.	Ground 4: Samaniego in View of Tso Renders Obvious Claim 9.....	108
X.	SECONDARY CONSIDERATIONS.....	112
XI.	CONCLUSION	112

I. INTRODUCTION

1. I have been retained by Akamai Technologies, Inc. (I will refer to them as “Akamai”) to provide my opinions on certain issues related to U.S. Patent No. 8,495,242 (the “’242 patent,” which I understand has been designated as Exhibit 1001) in connection with the above-captioned *inter partes* review (IPR) proceeding. In particular, I have been asked to provide my insights, analysis, and opinions regarding whether claim 9 of the ’242 patent (the “Challenged Claim”) is disclosed by and/or obvious in view of the prior art references identified below.

2. I understand the ’242 patent is titled “Automated Media Delivery System,” identifies as its named inventors Sean Barger, Steve Johnson, Matt Butler, Jerry Destremps, David Pochron, and Trent Brown, and is currently owned by Equil IP Holdings LLC. I have considered the ’242 patent.

3. I understand that the file history of the ’242 patent has been designated as Exhibit 1002. I have considered this file history, and I will refer to it as the “’242 File History” or by its exhibit number.

4. I understand that the ’242 patent was filed as U.S. Patent Application 12/713,637 on February 26, 2010.

5. I have considered the prior art cited in my declaration, including:

- a. **U.S. Patent No. 6,421,733** to Tso (“Tso”), titled “System for Dynamically Transcoding Data Transmitted Between Computers,” filed September 8, 1997 and issued July 16, 2002. I understand that a copy of this patent has been designated as Exhibit 1004.
- b. **U.S. Patent No. 6,438,576** to Huang (“Huang”), titled “Method and Apparatus of a Collaborative Proxy System for Distributed Deployment of Object Rendering,” filed on March 29, 1999, and issued August 20, 2002. I understand that a copy of this publication has been designated as Exhibit 1005.
- c. **U.S. Publication No. 2002/0078093** to Samaniego (“Samaniego”), titled “Automated Media Delivery System,” filed August 14, 2001, and published June 20, 2002. I understand that a copy of this publication has been designated as Exhibit 1007.

6. I am being compensated by Akamai at my standard hourly consulting rate of \$650 for my time on this matter. My compensation is not dependent on the outcome of this proceeding.

7. In forming my opinions, I relied on the documents cited in this declaration and the documents identified in the attached **Appendix B**. These

documents comprise patents, file histories, printed publications, and other related documents. As discussed below, each document is a type that experts in my field would have reasonably relied upon when forming their opinions. Further, experts in my field would have had access to each document either through the applicable patent offices and/or well-known libraries, conferences, or publications in the field. My opinions are also based upon my personal and professional experience.

II. QUALIFICATIONS

8. I received my Bachelor of Technology (Honors) in Electronics and Electrical Communication Engineering at the Indian Institute of Technology (IIT) in Kharagpur, India in 1984. I obtained my Ph.D. in Electrical Engineering and Computer Science at the University of California, Berkeley, in 1989. I received the Demetri Angelakos Outstanding Graduate Student Award from the University of California, Berkeley and the IEEE/ACM Ira M. Kay Memorial Paper Prize in 1989.

9. I am a tenured Full Professor in the Colleges of Computing and Engineering at Georgia Tech. I am knowledgeable and familiar with software engineering, web applications, cloud computing, data analytics, wireless communications, microprocessor architecture, hardware, RF, cellular networks, ASIC design, computer engineering, embedded systems, digital signal processing, and associated software and firmware design. I have created and taught

undergraduate and graduate courses in hardware and software design for computer science and engineering applications at Georgia Tech for the past thirty years. Additionally, I have been active in the areas of wireless communications, digital signal processing, integrated circuit design (analog & digital), software engineering, system-level design methodologies and tools, and software systems. I have been the principal investigator (“PI”) or co-PI in several active research programs in these areas, including DARPA’s Rapid Prototyping of Application Specific Signal Processors, the State of Georgia’s Yamacraw Initiative, the United States Army’s Federated Sensors Laboratory Program, and the United States Air Force Electronics Parts Obsolescence Initiative. I have received an IBM Faculty Award and NSF’s Research Initiation Award.

10. I have designed several specialized computer and communication systems over the past two decades at Georgia Tech for tasks such as wireless audio and video processing and protocol processing for portable platforms, such as cell phones and PDAs. I have worked on designing systems that are efficient from performance, size, weight, area, and thermal considerations. I have developed courses and classes for the industry on these topics, and many of my lectures in advanced computer system design, developed under the sponsorship of the United States Department of Defense in the late 1990s, are available for educational use at

“<http://www.eda.org/rassp>” and have been used by several U.S. and international universities as part of their course work. Some of my recent publications in the area of design of wireless communications systems and associated protocols are listed in **Appendix A**. I graduated more than 20 Ph.D. students that now work as professors or in technical positions around the world.

11. I have been active in research in the area of the World Wide Web and some of my peer-reviewed publications in this area include (i) Arshdeep Bahga and Vijay K. Madisetti, Performance Evaluation Approach for Multi-Tier Cloud Applications, Journal of Software Engineering and Applications 74-83 (2013); (ii) Arshdeep Bahga and Vijay K. Madisetti, Synthetic Workload Generation for Cloud Computing Applications, Journal of Software Engineering and Applications 396-410 (2011); (iii) Venu Dasigi and Vijay Madisetti, A Web-Based Interface for a Digital Broadband Home, Yamacraw IAB Workshop (2000); and (iv) Arshdeep Bahga and Vijay K. Madisetti, Rapid Prototyping of Multitier Cloud-Based Services and Systems, Computer, vol. 46, no. 11, pp. 76-83 (Nov. 2013).

12. I have been an active consultant to industry and various research laboratories (including Massachusetts Institute of Technology Lincoln Labs and Johns Hopkins University Applied Physics Laboratory). My consulting work for MIT Lincoln Labs involved high resolution imaging for defense applications, where

I worked in the area of prototyping complex and specialized computing systems. My consulting work for the Johns Hopkins Applied Physics Lab (“APL”) mainly involved localization of objects in image fields, where I worked on identifying targets in video and other sensor fields and identifying computer architectures and circuits for power and space-efficient designs.

13. I have founded three companies in the areas of embedded software, military chipsets involving imaging technology, and wireless communications. I have supervised the Ph.D. dissertations of over twenty engineers in the areas of computer engineering, signal processing, communications, rapid prototyping, and system-level design methodology, of which five have resulted in thesis prizes or paper awards. The first of the companies I founded, VP Technologies, offers products in the area of semiconductor integrated circuits, including building computing systems for imaging systems for avionics electronics for the United States Air Force and the US Navy, since 1995. I remain a director of VP Technologies. The second of these companies, Soft Networks, LLC, offers software for multimedia and wireless computing platforms, including the development of a set-top box for Intel that decodes MPEG-2 video streams, wireless protocol stacks, and imaging codecs for multimedia phones. The technology involved with the design, development, and implementation of the Intel set-top box included parsing

the bit streams, decoding communications protocols, extracting image and video data, and then processing for subsequent display or storage. The third of these companies, Elastic Video, uses region of interest based video encoding or decoding for capturing high quality video at very low bit rates, with primary application for wireless video systems.

14. I have authored more than sixty refereed journal publications and around forty peer reviewed conference publications. I have been active in research in the area of wireless and mobile communications and some of my recent peer-reviewed publications in this area include: (i) Mustafa Turkboylari & Vijay K. Madisetti, Effect of Handoff Delay on the System Performance of TDMA Cellular Systems, Proceedings of the Fourth IEEE Conference on Mobile and Wireless Communications Network 411-15 (Sept. 9-11, 2002); (ii) Loran A. Jatunov & Vijay K. Madisetti, Computationally-Efficient SNR Estimation for Bandlimited Wideband CDMA Systems, 5 IEEE Transactions on Wireless Communications, no. 12 (2006) at 3480-91; and (iii) Nimish Radio, Ying Zhang, Mallik Tatipamula & Vijay K. Madisetti, Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions, 100 Proceedings of the IEEE, no. 4 (April 2012) at 841-54. I have extensive experience analyzing, designing, and testing systems based on 3GPP Technical Specifications, including specifications describing WCDMA and HSDPA

technologies. I have been active in the area of location-based services and wireless localization techniques since the mid-1990s, and have authored several papers on location-based services, including, Vijay K. Madisetti et al., Mobile Fleet Application Using SOAP and System on Devices (SyD) Middleware Technologies, Communications, Internet, and Information Technology (2002) at 426-31. I have served as associate editor or on the editorial board for technical journals, including IEEE Transactions on Circuits & Systems II, International Journal in Computer Simulation, and International Journal in VLSI Signal Processing.

15. I have authored or co-authored several books, including VLSI Digital Signal Processors (IEEE Press 1995) and the Digital Signal Processing Handbook (CRC Press, 1998, 2010). I co-authored Quick-Turnaround ASIC Design in VHDL (Kluwer Academic Press 1996) and Platform-Centric Approach to System-on-Chip (SoC) Design (Springer 2004). I am also the editor of several books, including the three-volume DSP Handbook set: Volume 1: Digital Signal Processing Fundamentals, Volume 2: Video, Speech, and Audio Signal Processing and Associated Standards, and Volume 3: Wireless, Networking, Radar, Sensory Array Processing, and Nonlinear Signal Processing, published in 2010 by CRC Press, Boca Raton, Florida. More recently I have authored Cloud Computing (2014, CreateSpace Press), and Internet of Things (2014, CreateSpace), and the book, Cloud Computing,

was nominated as a Notable Book of 2014 by the Association of Computing Machinery (ACM) in July 2015.

16. My experience is relevant to this case. In addition to authoring textbooks and research papers in the area of cloud computing, virtualization, web-services, and web-based applications, I have developed web-based document management systems and also web-based fintech applications. I am listed as an inventor on over a dozen patents on web-based applications. I also invented a new protocol that replaces the HTTP protocol for token-based networks, called VTTP, that is a subject of a few issued US patents.

17. I have been extensively involved in the activities of one of the premier standard setting organizations in the world, the IEEE, since the 1980s, and I have participated in the development of standards for hardware design and description languages, such as VHDL, used in design of computer chips – IEEE 1076.6. I was also a member of the IEEE Press Board from 1995-1997. I was the Technical Program Chair of IEEE ICASSP in 1996. I was also the Associate Editor of IEEE Transactions on Circuits & Systems II from 1994 to 1997. I am also the director of the IEEE Atlanta Chapter, and the author of several drafts of IETF proposals.

18. I have served as associate editor or on the editorial board for technical journals, including IEEE Transactions on Circuits & Systems II, International

Journal in Computer Simulation, and International Journal in VLSI Signal Processing. I have been elected a Fellow of the IEEE, for contributions to embedded computing systems. The Fellow is the highest grade of membership of the IEEE, a world professional body consisting of over 300,000 electrical and electronics engineers, with only one tenth of one percent (0.1%) of the IEEE membership being elected to the Fellow grade each year. Election to Fellow is based upon votes cast by existing Fellows in IEEE.

19. I have also been awarded the 2006 Frederick Emmons Terman Medal by the American Society of Engineering Education for contributions to Electrical Engineering, including authoring a widely used textbook in the design of VLSI digital signal processors. I was awarded VHDL International Best PhD Dissertation Advisor Award in 1997 and the NSF RI Award in 1990. I was Technical Program Chair for both the IEEE MASCOTS in 1994 and the IEEE Workshop on Parallel and Distributed Simulation in 1990. In 1989, I was recognized with the Ira Kay IEEE/ACM Best Paper Award for Best Paper presented at the IEEE Annual Simulation Symposium.

20. I have submitted approximately 100 invention disclosures and provisional patents over the past ten years. I am listed as a first inventor on over thirty issued US patents.

21. I have testified as an expert witness before. Over the past two years, I've testified as an expert in more than 20 proceedings, and several of these were in the area of enterprise, cloud and web-based software applications.

22. I have attached a more detailed list of my qualifications as **Appendix A**.

III. LEGAL UNDERSTANDING

A. My Understanding of Claim Construction

23. I have been informed that patent claims are construed from the viewpoint of a person having ordinary skill in the art of the patent at the time of the invention. I have been informed that patent claims generally should be interpreted consistent with their plain and ordinary meaning as understood by a person having ordinary skill in the art in the relevant time period (*i.e.*, at the time of the purported invention, or the so called "effective filing date" of the patent application), after reviewing the patent claim language, the specification, and the prosecution history (*i.e.*, the intrinsic record).

24. I have further been informed that a person having ordinary skill in the art must read the claim terms in the context of the claim itself, as well as in the context of the entire patent specification. I understand that in the specification and prosecution history, the patentee may specifically define a claim term in a way that differs from the plain and ordinary meaning. I understand that the prosecution history of the patent is a record of the proceedings before the U.S. Patent and

Trademark Office, and may contain explicit representations or definitions made during prosecution that affect the scope of the patent claims. I understand that an applicant may, during the course of prosecuting the patent application, limit the scope of the claims to overcome prior art or to overcome an examiner's rejection, by clearly and unambiguously arguing to overcome or distinguish a prior art reference, or clearly and unambiguously disavowing claim coverage.

25. In interpreting the meaning of the claim language, I understand that a person having ordinary skill in the art may also consider "extrinsic" evidence, including expert testimony, inventor testimony, dictionaries, technical treatises, other patents, and scholarly publications. I understand this evidence is considered to ensure that a claim is construed in a way that is consistent with the understanding of those of ordinary skill in the art at the time of the claimed invention. For example, this can be useful for a technical term whose meaning may differ from its ordinary English meaning. I understand that extrinsic evidence may not be relied on if it contradicts or varies the meaning of claim language provided by the intrinsic evidence, particularly if the applicant has explicitly defined a term in the intrinsic record.

B. My Understanding of Anticipation

26. I understand that a patent claim is invalid as “anticipated” if all of the limitations of the claim are present, either expressly or inherently, in a single pervious device, or described, either expressly or inherently, in a single prior art publication or patent. I understand that, in the relevant time period, 35 U.S.C. § 102 stated that a person shall be entitled to a patent unless:

- The invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent, or
- The invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States, or
- He has abandoned the invention, or
- The invention was first patented or caused to be patented, or was the subject of an inventor’s certificate, by the applicant or his legal representatives or assigns in a foreign country prior to the date of the application for patent in this country on an

application for patent or inventor's certificate filed more than twelve months before the filing of the application in the United States, or

- The invention was described in—(1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language; or
- He did not himself invent the subject matter sought to be patented, or
- (1) during the course of an interference conducted under section 135 or section 291, another inventor involved therein establishes, to the extent permitted in section 104, that before

such person's invention thereof the invention was made by such other inventor and not abandoned, suppressed, or concealed, or (2) before such person's invention thereof, the invention was made in this country by another inventor who had not abandoned, suppressed, or concealed it. In determining priority of invention under this subsection, there shall be considered not only the respective dates of conception and reduction to practice of the invention, but also the reasonable diligence of one who was first to conceive and last to reduce to practice, from a time prior to conception by the other.

27. I also understand that a prior art reference can disclose a claimed feature because the feature is expressly described or because the feature is inherent in the disclosure. I understand that something is inherent in a prior art reference if the missing descriptive matter must necessarily be present, not merely probably or possibly present, and it would be so recognized by a person of ordinary skill in the art as of the priority date of the patent.

28. Claim limitations that are not expressly found in a prior art reference are inherent if the prior art necessarily functions in accordance with, or includes, the claim limitations. I understand that it is acceptable to examine evidence outside of

the prior art reference (i.e., extrinsic evidence) in determining whether a feature, while not expressly discussed in the reference, is necessarily present in it.

C. My Understanding of Obviousness

29. I understand that a claim may be invalid if the subject matter described by the claim as a whole would have been obvious to a hypothetical person of ordinary skill in the art in view of a prior art reference or in view of a combination of references at the time the claimed invention was made. Therefore, I understand that obviousness is determined from the perspective of a hypothetical person of ordinary skill in the art and that the asserted claims of the patent should be read from the point of view of such a person at the time the claimed invention was made. I further understand that a hypothetical person of ordinary skill in the art is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit.

30. I have been informed that there are two criteria for determining whether prior art is analogous and thus can be considered prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the patentee's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the patentee is involved. I have also been informed that the field of endeavor of a patent is not

limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. I have also been informed that a reference is reasonably pertinent if, even though it may be in a different field from that of the patentee's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to a patentee's attention in considering their problem.

31. I have also been advised that an analysis of whether a claimed invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the claimed invention, and the level of ordinary skill in the pertinent art involved. I understand as well that a prior art reference should be viewed as a whole.

32. I have also been advised that in considering whether a claimed invention could be obvious over a combination of prior art references, I may assess whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the marketplace, and/or the background knowledge possessed by a person having ordinary skill in the art. I understand that other principles may be relied on in evaluating whether a

claimed invention would have been obvious, and that these principles include the following:

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or in a different one, so that if a person having ordinary skill can implement a predictable variation, the variation is likely obvious;
- If a technique has been used to improve one device, and a person having ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond their skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of

obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine;

- Market demand, rather than scientific literature, can drive design trends and may show obviousness;
- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation nor the avowed purpose of the named inventor controls whether the claim is obvious;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims;
- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a person having ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle;

- A person having ordinary skill in the art is also a person having ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was “obvious to try,” particularly when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions such that a person having ordinary skill in the art would have had good reason to pursue the known options within his or her technical grasp; and
- One should be cautious of using hindsight in evaluating whether a claimed invention would have been obvious.

33. I further understand that, in making a determination as to whether the claimed invention would have been obvious to a person having ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of products practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as “secondary considerations” or “objective indicia” of nonobviousness. I understand, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a

“nexus”) between the claim and the evidence and that this nexus must be based on a novel element of the claim rather than something in the prior art. I also understand that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (such as motivation to combine with predictable results) that is sufficiently strong.

34. I have been asked to consider the validity of claim 9 of the '242 patent, which I refer to as the “Challenged Claim.” I understand that for *inter partes* reviews, invalidity must be shown under a preponderance of the evidence standard. I have concluded that the Challenged Claim is anticipated by and rendered obvious by the prior art based on the references described below and as explained herein.

IV. BACKGROUND OF THE TECHNOLOGY

35. As discussed in greater detail in Section V, the '242 patent states that it is directed to a method of automatically transcoding media content. The method uses information about a client's system (*e.g.*, browser attributes) to select and further process a cached version of content and returns that processed version to the client. *See* '242 patent, 20:7-9, 23:45-24:16.

36. Specifically, challenged claim 9 recites receiving a request for media content, for which there are cached versions of that content and each cached version is associated with a different base “transcoding profile[.]” When the request is

received, the system automatically identifies transcoding parameters, based on one or more formats of the client presentation system, to be applied to the content before the content is returned to the client. Using these parameters, the system takes a base transcoding profile and performs additional incremental transcoding on top of the base profile in order to complete transcoding, and then returns the content to the client and caches the further transcoded content.

37. When allowing claim 9, the examiner stated that the prior art does not disclose the steps recited in claim elements [9.e]-[9.f], which recite “selecting a pre-existing base transcoded version” of the content that has been transcoded “with only a portion of the identified transcoding parameters,” and “creating a final version by incrementally performing further transcoding” of that “base transcoded version in accordance with a remaining portion of the identified transcoding parameters.” Ex. 1002, 824-25.

38. Based on my experience in the field, however, the features recited in elements [9.e]-[9.f] of the '242 patent were well-known in the art prior to the '242 patent's alleged priority date. One example of such prior art is U.S. Patent No. 6,521,733, which was originally assigned to Intel, and identifies Tso as its first named inventor. I understand that a copy of Tso has been designated as Exhibit 1004. Tso discloses a server requesting a specific, “particular version of” a media

object using a “GetScaledObject” call, and teaches using this call to retrieve a “particular version” of content such as a version that has already been processed and cached. Tso, 6:9-13.

39. Another relevant prior art reference is U.S. Patent No. 6,438,576, which was originally assigned to IBM, and which identifies Huang as the first named inventor. I understand that a copy of Huang has been designated as Exhibit 1005. Huang teaches retrieving a partially-rendered, intermediate media item from a cache and completing required rendering for that media item using this retrieved cached version of the media item. Huang, 7:23-8:11. As explained in greater detail below in Section IX.A.2, a person having ordinary skill in the art would have been motivated to modify Tso with Huang’s teachings to advantageously reduce the unnecessary re-transcoding and re-retrieval of media content. Applying Huang’s teachings to Tso yields a system that retrieves an already partially-rendered content item from a cache, and then uses that retrieved item to complete any remaining, necessary rendering for the client.

40. Another example disclosure that I have considered is U.S. Patent Application Publication No. 2002/0078093, which is a publication of Application No. 09/929,904, and which identifies Samaniego as the first named inventor. I understand that a copy of Samaniego has been designated as Exhibit 1007. As can

be seen from comparing the cover pages of Samaniego and the '242 patent, Samaniego does not share any common inventors with the '242 patent. I have been instructed to assume that Samaniego qualifies as prior art to the '242 patent, and have not formulated my own separate opinion as to whether Samaniego qualifies as prior art. Samaniego's specification is substantively identical to the '242 patent's specification. I have compared the text of Samaniego and the text of the '242 patent and generated a redline comparison between the specifications. I understand that a copy of that redline comparison has been designated as Exhibit 1013. I have identified specification support in Samaniego that not only renders obvious, but actually discloses, claim 9 of the '242 patent.

41. None of the features recited in the Challenged Claim of the '242 patent was novel at the time of the alleged invention, and there was nothing novel as to how the inventors of the '242 patent combined those features. As I discuss below, the Challenged Claim is anticipated by and obvious in view of the prior art. Further, there were no technological barriers to combining these features to form the claimed invention, and a person having ordinary skill in the art would have been motivated to combine these elements and would have understood that the combination would have yielded predictable results.

V. THE '242 PATENT AND PROSECUTION HISTORY

A. '242 Patent Overview

42. Claim 9 of the '242 patent is a method claim that generally recites a method for processing media by using partially-processed content stored in a cache, and further processing that content to generate a final version of media content to be served in response to a user request. '242 patent, 1:19-23, 17:35-39, 22:24-28. The method starts with a client presentation system issuing a request for media content. '242 patent, 23:45-48; *see also* '242 patent, 19:12-15. The requested content item has a limited number of associated “base transcoding profiles” that each correspond to a cached version of the requested content. '242 patent, 23:49-52, 17:27-34, 19:63-67. The system uses a “user profile procedure” to identify “transcoding parameters” to be applied to the content. '242 patent, 23:53-65. The transcoding parameters are based on “specified browser or client attributes,” and are used by the system to determine whether the same transcoding parameters are being applied to the same content prior to delivery to another “client presentation system.” '242 patent, 23:53-65; *see also* '242 patent, 19:15-22, 19:48-52, 20:7-9, Figs. 21-22. To transcode the data, the system takes a pre-existing cached version of the requested content that has been transcoded according to only some of the “transcoding parameters,” and then applies the remaining “transcoding parameters” to that version. '242 patent, 23:65-24:14; *see also* '242 patent, 19:64-20:1. This transcoded version is then delivered

to both the originally-requesting client presentation system, as well as the other “client presentation system” that requested the same content. ’242 patent, 24:15-16; *see also* ’242 patent, 20:9-11.

B. Prosecution History of the ’242 Patent

43. In formulating my opinions, I considered the contents of the prosecution history of the ’242 patent, a copy of which I understand has been designated as Exhibit 1002. During prosecution, the draft claim that issued as the Challenged Claim was thrice rejected as obvious in view of prior art. Ex. 1002 (’242 FH), 81-109, 124-53, 744-91. After several amendments to the Challenged Claim, the Applicant added claim limitations to the Challenged Claim reciting that the “act of transcoding the requested media content in accordance with the identified transcoding parameters” comprises (1) “selecting a pre-existing base transcoded version of the requested media content comprising intermediate derivative media that has been transcoded in accordance with only a portion of the identified transcoding parameters” and (2) “creating a final version by incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.” Ex. 1002, 792-814. The Examiner then allowed Challenged Claim 9 on March 30, 2013, and, emphasizing these two elements, stated:

The Examiner finds the combination of prior art of records fail to disclose the features of the invention including *an act of selecting a pre-existing base transcoded version of the requested media content comprising intermediate derivative media that has been transcoded in accordance with only a portion of the identified transcoding parameters; and an act of creating a final version by incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters*; as claimed in the invention to reduce the system need for a web author to create different versions of a web site, while also generating media content only if requested by a client browser....

Ex. 1002, 824-25 (emphasis in the original), 820-26. The '242 patent issued on July 23, 2013. Ex. 1002, 877.

VI. THE '242 PATENT PRIORITY DATE

44. The '242 patent matured from U.S. Application 12/713,637, filed on February 26, 2010. The '242 patent claims priority to U.S. Patent Applications 09/425,326 (filed October 21, 1999), 09/929,904 (filed August 14, 2001), 11/269,916 (filed November 7, 2005), and 12,173,747 (filed July 15, 2008).

45. I have not formulated any opinion about the correct priority date for the '242 patent. For purposes of this declaration, I was asked to assume, with respect to each of the prior art Grounds discussed in this declaration, that the '242 patent is entitled to the priority date of November 7, 2005.

46. With respect to my opinions regarding both claim construction and prior art Ground 1 set forth below, I was also asked to consider whether my opinions would be affected if the '242 patent was instead entitled to a priority date no later than October 21, 1999. My claim construction opinions and opinions with respect to prior art Ground 1 below would not be materially impacted if the '242 were entitled to this October 21, 1999 date, and these opinions as set forth in this declaration apply equally under either priority date.

VII. LEVEL OF ORDINARY SKILL IN THE ART

47. In rendering the opinions set forth in this declaration, I was asked to consider the patent claims and the prior art through the eyes of a person having ordinary skill in the art at the time of the alleged invention. I understand that the factors considered in determining the ordinary level of skill in a field of art include the level of education and experience of persons working in the field; the types of problems encountered in the field, the teachings of the prior art, and the sophistication of the technology at the time of the alleged invention. I understand that a person having ordinary skill in the art is not a specific real individual, but rather is a hypothetical individual having the qualities reflected by the factors above. I understand that a person having ordinary skill in the art would also have knowledge from the teachings of the prior art, including the art cited below.

48. With respect to all of the Grounds, I have been asked to assume that the '242 patent is entitled to the priority date of November 7, 2005. Upon consideration of these factors, in my opinion, on or before this priority date, a person having ordinary skill in the art relating to the technology of the '242 patent would have had a minimum of either a bachelor's degree in computer systems, computer science, or the equivalent thereof, and at least two years of experience with networked media delivery or related technologies. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.

49. I have also been asked to consider whether my understanding of a person of ordinary skill in the art would be different if the '242 patent were entitled to a priority date no later than October 21, 1999 rather than November 7, 2005. My definition of a person of ordinary skill in the art would remain the same with respect to the October 21, 1999 priority date, and any difference in the level of ordinary skill between these priority dates does not materially impact my opinions set forth in this declaration.

50. Regardless of which priority date applies, I am qualified to opine concerning what a person having ordinary skill in the art would have known and understood during that time. Well before both November 7, 2005 and October 21,

1999, my level of skill in the art was at least that of a person having ordinary skill. I am qualified to opine concerning what a person having ordinary skill in the art would have known and understood at the time of the '242 patent's purported invention. My analysis and conclusions herein are from the perspective of a person having ordinary skill in the art as of November 7, 2005.

VIII. CLAIM CONSTRUCTION

51. I have been instructed that the Challenged Claim 9 of the '242 patent is to be given its ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the alleged invention in light of the specification and the prosecution history pertaining to the patent. I have followed these principles in my analysis set forth in this document.

52. I understand that there is a pending district court litigation involving the '242 patent, and the parties have not engaged in claim construction in that litigation. I reserve the right to supplement my opinions in this declaration to the extent that the parties later propose claim constructions for any claim terms in the '242 patent.

A. Preambles

53. I understand the preamble of a claim may or may not be limiting. For purposes of this declaration, I have not been asked to formulate any opinion as to whether the Challenged Claim's preamble is limiting, and have not formulated any such opinion. Instead, I have been instructed to consider whether the prior art

discloses the preamble, to the extent it is limiting. To the extent it is limiting, it is disclosed by the prior art as discussed in Section IX.

IX. GROUNDS OF UNPATENTABILITY

54. As I explained above in Section IV, although Challenged Claim 9 of the '242 patent purports to claim a method of accessing dynamically transcoded media content, this method was well-known in the art. As I describe below, the cited prior art provides teachings of this method.

55. In this section, I will explain why Samaniego anticipates and renders obvious, and why further combinations of teachings from the following references also render obvious the Challenged Claim:

Name	US Patent/ Publication	Exhibit	Filed	Published/ Issued
Tso	6,421,733	1004	9/8/1997	7/16/2002
Huang	6,438,576	1005	4/29/1999	8/20/2002
Samaniego	2002/0078093	1007	8/14/2001	6/20/2002

I have been instructed to assume that each of these references qualifies as prior art.

56. Specifically, as I explain below, the Challenged Claim is invalid based upon the following prior art grounds:

- a. **Ground 1:** Challenged Claim 9 of the '242 patent is rendered obvious by Tso in view of Huang;

- b. **Grounds 2:** Challenged Claim 9 is further anticipated by Samaniego;
- c. **Ground 3:** Challenged Claim 9 is further rendered obvious by Samaniego;
- d. **Ground 4:** Challenged Claim 9 is rendered obvious by the teachings of Samaniego in view of Tso.

57. It is my opinion that Challenged Claim 9 of the '242 patent is unpatentable as explained below. I note that in this discussion, I may focus on the particular language in the Challenged Claim, but it should be understood that my analysis applies to the corresponding elements of the Challenged Claim containing identical or substantially identical language.

A. Ground 1: Tso in view of Huang Renders Obvious Claim 9

58. Challenged Claim 9 of the '242 patent is obvious over Tso in view of Huang. Tso teaches a system that dynamically transcodes and then caches media based on features of a client system in response to a client request. Huang teaches a method for object rendering (including by transcoding) in which partially-rendered objects are retrieved from cache and used to complete rendering of an object for a client without unnecessary re-retrieval or re-rendering of content. Thus, as further set forth below, Tso in view of Huang renders obvious claim 9 of the '242 patent.

1. U.S. 6,421,733 (“Tso”) Overview

59. Tso discloses a “system for dynamically transcoding data transmitted between two computers over a communications link.” Tso, 1:13-14. An embodiment of this system is shown in Fig. 3, which is annotated below:

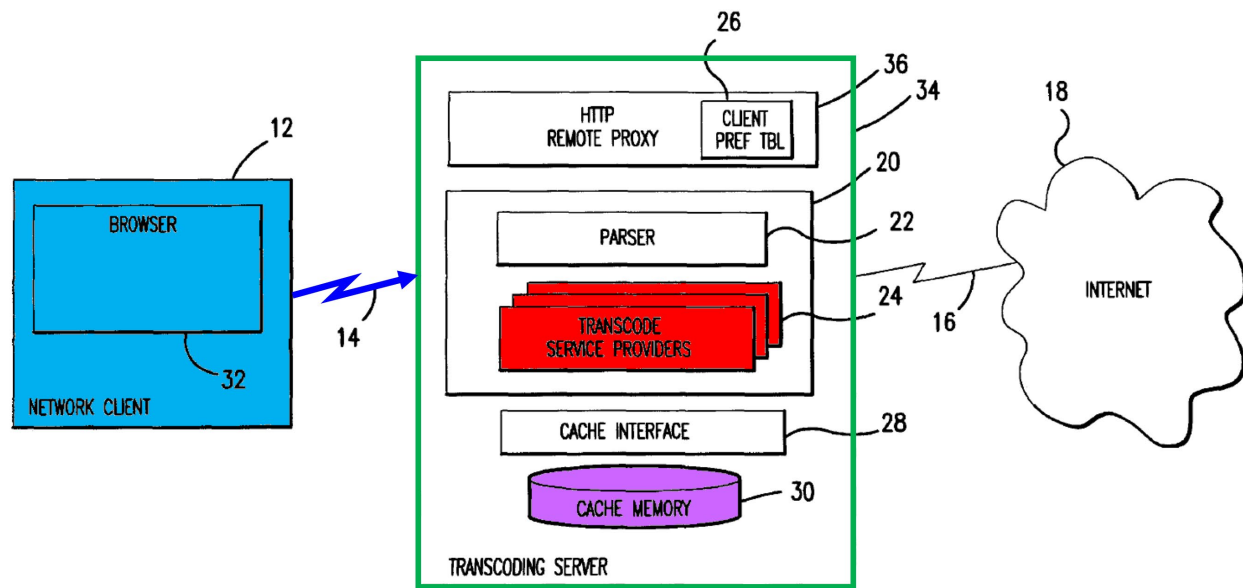


FIG.3

60. As is shown and described, a **network client** includes a browser and sends a **request for media content, indicated as arrow 14**, such as a web page including “image, video, or HTML” content, to a **transcoding server**. Tso, 3:10-14, 3:45-53, 4:6-10, 9:54-65, 14:1-7. The “request packet” sent by the client carries “information” in both its “header” and “data portion[s]” that includes “selection criterion,” which is used to specify what transcoding should be applied to the

content. Tso, 6:64-8:9.¹ Tso further discloses that the request packet stores different selection criteria. For example, selection criteria may specify the “particular [media] content” being requested, as well as “network client” information “such as a display dimension, resolution, number of colors, processor type, memory/disk configuration, modem or network interface type”; these types of selection criteria describe what are referred to as formats of a client presentation system as claimed in the ’242 patent. Tso, 7:4-7, 7:20-33. For example, the ’242 patent describes a “user profiling system” that is used to customize an image “for specified browser or client attributes,” and customize content for “client connection speed.” ’242 patent, 20:7-10 (“The resulting image is then customized by the user profiling system (2280) for specified browser or client attributes.”), 15:46 (“Optimizes images for client connection speed.”), 23:56-59 (“wherein identification of transcoding parameters is based on one or more formats of any client presentation system”). The system uses

¹ Tso sometimes inconsistently uses the terms “selection criterion” and “selection criteria” to collectively refer to either singular selection criterion and combinations of different selection criteria. *See, e.g.*, Tso, 7:20-21 (using the singular “criterion” to refer to a list of criteria: “The predetermined selection criterion may comprise: [examples of selection criteria]”), 8:10-13 (using the plural “criteria” to refer to the same list of criteria: “Applying the above-listed selection criteria, or combinations thereof, embodiments of the present invention may be used to provide a virtually limitless range of dynamic transcoding services.”), 15:36-38 (“location-specific characteristics are to be used as predetermined selection criteria”). I have used “criteria” and “criterion” consistent with Tso’s teachings of these terms, and generally use “criterion” to refer to the singular and “criteria” to refer to the plural.

these criteria and “combinations thereof” to provide a “virtually limitless range” of “dynamic transcoding services,” including by selecting which of the “**transcode service providers 24** are invoked” to generate the version of content to be provided in response to the client’s request. Tso, 3:48-65, 7:15-17, 8:10-14.

61. After the request is received by the **transcoding server**, it is handed to a transcoder 20, which includes a “**parser**,” “**transcode service providers**,” and a “**cache interface**” connected to a “**cache memory**.” Tso, 3:8-17, 3:66-4:5, 4:14-19.

62. Below Figures 8 and 9 are annotated to summarize relevant portions of the process disclosed by Tso:

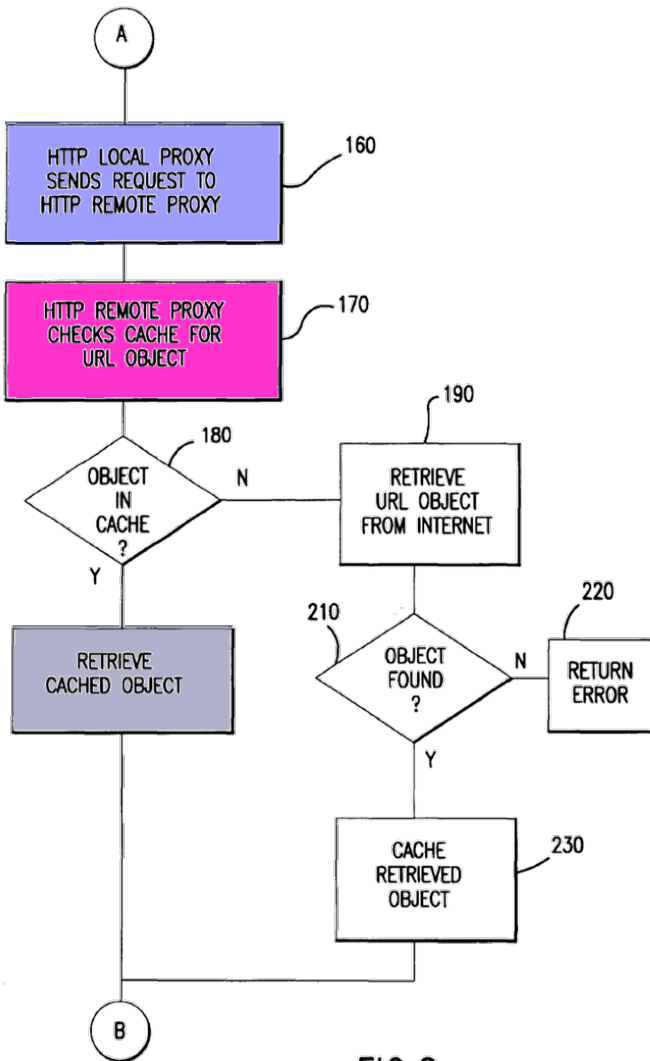


FIG. 8

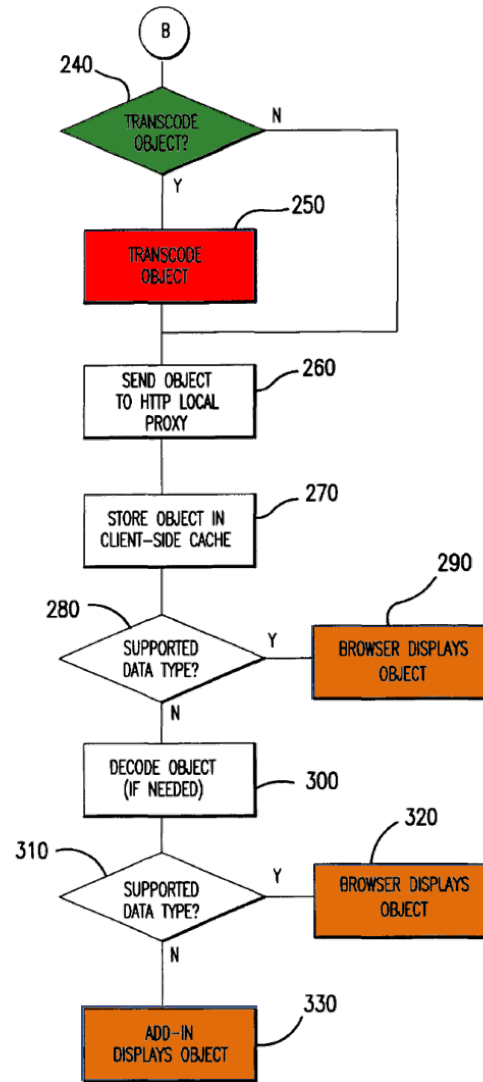


FIG. 9

63. At step 160, the client **sends a request** to the server for a media content item. Tso 3:11-16, 14:21-25. After receiving this **request**, at **step 170**, the HTTP remote proxy invokes the parser to issue a “GetScaledObject” call to **determine whether a particular version of the requested content item is cached**. Tso, 14:25-30. For example, this request can be used to request a version with a “stage” property specifying the progress of “staged refinements” where the transcoding

process is staged into incremental transcoding steps, which not only renders obvious, but discloses using a “stage” parameter to retrieve partially-transcoded content items that are at a partially complete transcoding stage. Tso, 5:36-42, 6:9-23 (“Unlike traditional caching proxies, transcode service providers 24 can use server-side cache memory 30 to store several different versions of an object to support clients with different communications and/or presentation capabilities. Thus, an additional “Stage” parameter may be used to indicate which version of the cached object is to be returned to network client 12.”). If the requested content item is cached, then it is **retrieved from cache**, and **returned to the client**. Tso 14:30-32. If the requested item is not in cache then, when a content item is retrieved, at **step 240 the parser determines whether additional transcoding is required** by using “predetermined selection criteri[a].” Tso, 14:47-50, 7:15-8:9. The system performs any required transcoding by using one or more **transcode service providers**. Tso, 14:50-55, 10:37-49. Each of the different transcode service providers is used to perform different transcoding operations such that, if there is a need for additional transcoding techniques to be implemented by the system, additional providers are “dynamically added to the system to provide new features and/or better transcoding algorithms.” *Id.*; Tso, 4:23-37.

64. Tso further teaches that as part of the transcoding process, both the “original” version, as well as “several different” “transcoded” “versions of content are cached. Tso, 4:1-5, 6:13-23. Once the requested version of the content is generated and cached, the server uses additional GetScaledObject() requests to retrieve that desired version of the content and, at **step 260**, the server **returns the media content item to be displayed on the client’s browser**. Tso, 6:42-50, 14:52-57.

65. Tso further discloses that the system uses “multiple-thread processing” to “process requests for the same hypertext object from multiple network clients.” Tso, 5:43-50, 6:51-63. Multiple-thread processing allows a device to perform multiple tasks, such as processing requests for content, in parallel, as opposed to single thread processing which requires incoming tasks to be handled serially. For example, Tso teaches that if a server receives requests from two different clients, both of which are requesting different versions of the same media content item, the server can retrieve the content item once from the origin of the content (*e.g.*, the site that originally generated the content), process “appropriate versions” of the item for the different “client characteristics” that each requester will have (for example, for different types of computers with screens that have different resolutions), and then transmit those differently transcoded versions “to such multiple network clients 12

concurrently.” Tso, 6:51-63, 7:60-8:9. The same content item may be “concurrently read” by multiple clients. Tso, 5:43-51. In addition to teaching that transcoding service providers within a server are used for transcoding, Tso also teaches that different devices can distribute the transcoding process such that, for example, a client computer enabled to perform transcoding of partially-transcoded content that is received from a server (Tso, 14:47-15:6), and proxy servers communicating with one another are also able to “collaboratively” complete transcoding of content by having one proxy server provide a partially-completed transcoding of content to another proxy server, which completes transcoding of the partially-transcoded content item (Tso, 15:66-16:14). Tso thus discloses that servers are configured to both send and receive partially-transcoded content, and are also configured to cache content that has been transcoded.

2. U.S. 6,438,576 (“Huang”) Overview and Motivation to Modify Tso with Huang’s Teachings

66. Huang teaches a system and “object rendering method for a collaborative data network” such as the Internet that caches both “partially” and “completely” “rendered objects” in order to avoid repeating rendering operations on a content object. Huang, Abstract, 1:17-22, 6:63-67. Huang uses the term “rendering” to refer to image manipulation steps that would have been understood by a person having ordinary skill in the art to encompass steps implemented in

transcoding. Huang, 2:37-39. For example, Huang describes rendering processes in which content is rendered into “different forms or resolutions,” which describes processes such as scaling or changing the encoding format for content, as well as converting content from color to “black-and-white,” which describes processes in which color resolution content is converted into black and white content. Huang, 1:17-22, 1:57-58, 6:59-63. Tso describes similar processes as “transcoding” processes. For example, Tso teaches using selection criteria to transcode content based on a client device’s “resolution” and “number of colors,” as well as installed hardware and software for “decompression” and the “data type” for content—both of which are used to determine whether the client system can handle transcoded content of different forms. Tso, 7:15-33. Thus, Huang uses the term “rendering” to refer to processing steps including steps commonly referred to as “transcoding” steps in the art. *See id.* Importantly, this stands in contrast to the use of “rendering” as it is sometimes used in the art to describe the display or other performance of previously-generated or transcoded content. Huang separately uses the term “consum[ing]” to refer to “consuming” the content at a client (*e.g.*, preparing and displaying the content to a user) to refer to this latter process. Huang, 1:59-64. In this declaration, I use the term “rendering” and variations thereof to refer to the “transcoding” taught in Huang as part of the “rendering.”

67. Below, Figure 4 of Huang is annotated to summarize relevant portions of Huang's teachings of a rendering (including transcoding) process:

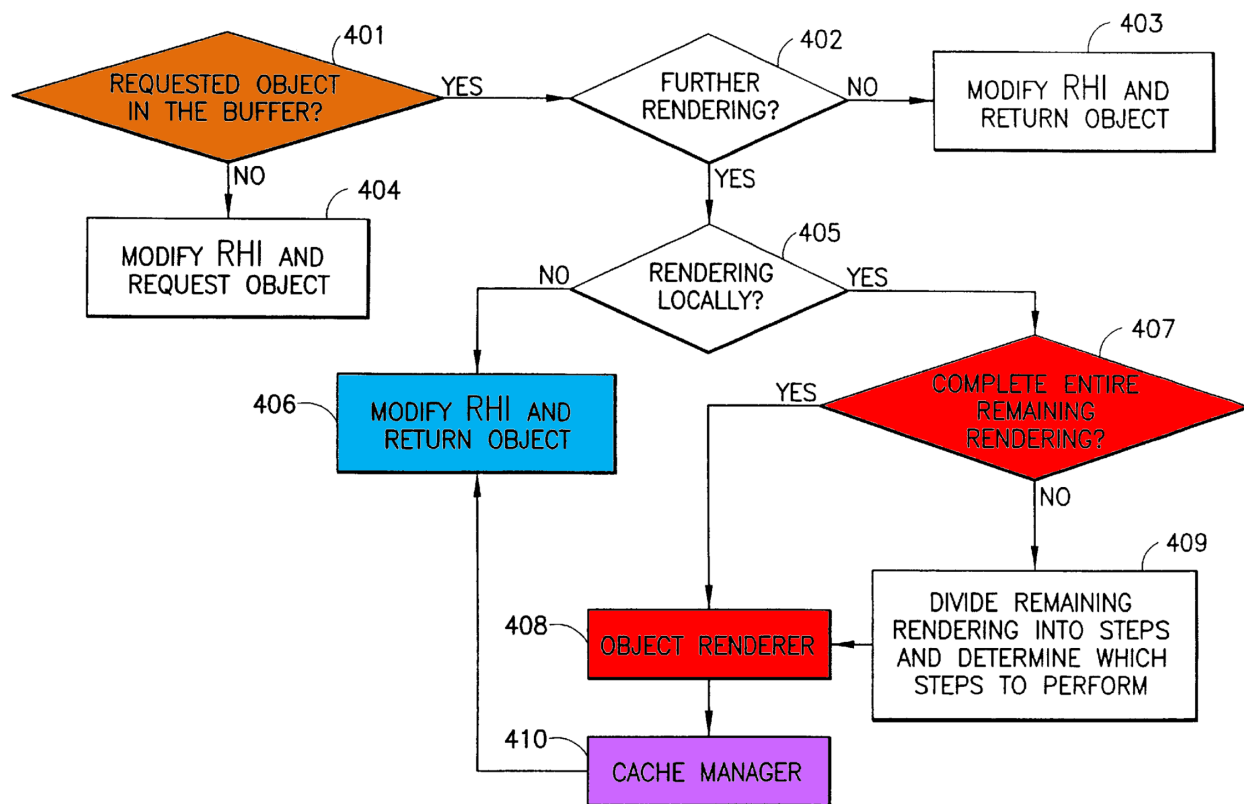


FIG.4

68. At **step 401**, the proxy server searches the cache to determine whether there is already a stored version of the requested content that is sufficiently “detailed” to handle the request. Huang, 7:23-36. A version is “less detailed” if the version “does not satisfy the requirement” for the requested item, while a “more detailed version” may be “further rendered” to “satisfy the request.” Huang, 7:30-36. If further rendering (including, for example, transcoding) of the requested object is necessary, then at **steps 407-408** the proxy server completes the remaining

rendering steps locally. Huang, 7:53-8:4. The system then caches a copy of the requested object (**step 410**), and then returns the requested content to the user (**step 406**). Huang, 8:4-8:11.

69. Huang further teaches how to track different versions of content in order to allow the system to retrieve partially-transcoded versions of content in order to further transcode such content. Huang teaches that each proxy server maintains “Receiver Hint Information” (RHI). Huang, 3:50-67. The RHI information provides “meta-information describing the capabilities of the object requesting node” as well as information “associated with a particular object.” Huang, 3:50-67, 6:56-58. For example, the RHI provides information that allows the server (and other servers in communication with the server) to determine “what portion ... of the remaining required rendering is to be performed by” the server, as well as to determine how to divide “a remaining rendering operation into steps” to be performed by a server. Huang, 4:24-37. For example, a server performs “complete or partial rendering based on the associated RHI” by “partition[ing]” the rendering process into “two or more steps” and modifying the RHI to “reflect the processing that [the server] performed.” Huang, 6:9-23. When a new request for previously-cached content is received, the server uses a cached object’s RHI to determine if “further rendering is

necessary” and whether to “complete the entire rendering process” locally. Huang, 7:43-64.

70. A person having ordinary skill in the art would have been motivated to modify Tso with relevant teachings from Huang. Tso and Huang are analogous prior art and in the same field as the '242 patent which, as I discussed above, with respect to the level of ordinary skill in the art, is networked media delivery. *See* '242 patent, 1:20-23 (“[T]he invention relates to an Internet server-based software system that provides delivery of automated graphics and other media to Web sites for access by an end user or consumer.”). Both Tso and Huang are also reasonably pertinent to the alleged problems identified in the '242 patent, namely, automating media delivery and reducing required overhead and time to generate Web pages. '242, 1:20-23 (“an Internet server-based software system that provides delivery of automated graphics and other media”), 4:50-56, 4:66-5:8, 6:4-8; Tso, 1:10-31 (“a system for dynamically transcoding data transmitted between two computers over a communications link”), 1:59-2:5, 4:1-5; Huang, 1:20-26 (“proxy servers perform distributed object rendering so that object contents can be displayed on or consumed by various kinds of client devices”), 2:1-10, 2:37-44.

71. A person having ordinary skill in the art would have understood that Tso not only renders obvious, but actually discloses, retrieving partially-transcoded

content from cache including as part of a staged transcoding process as I discussed in Section IX.A.1, as Huang provides explicit teachings of retrieving a partially transcoded version of a content item from cache, and then further processing that partially transcoded version of content to generate the requested version of content. *E.g.* Tso, 5:39-41 (disclosing retrieving stored information about cached objects such as “current compression progress for scaled data access and staged refinements”), 14:64-15:6; Huang, 6:63-67 (“Proxy server logic 203 may also include a cache manager 207 which maintains a local copy of the partially rendered or completely rendered objects in order to avoid repeating some object rendering operations with the same proxy server.”), 6:56-58 (“according to the RHI associated with a particular object”). Huang also explicitly discloses tracking the different versions of the generated and cached content using the RHI information that is included with the content. Huang, 6:9-23. Huang further states that retrieving such content advantageously avoids unnecessary re-rendering of content. Huang, 7:23-57. A person having ordinary skill in the art thus would have been motivated to modify Tso’s teachings of retrieving a specific version of content and performing “staged” transcoding by implementing these detailed teachings from Huang of retrieving and tracking partially-transcoded content items from a cache to achieve the beneficial and predictable result of tracking the transcoding status of content on

a server and reducing the processing involved in re-processing previously-processed data for at least the following independent reasons.

72. **First**, Tso's and Huang's teachings both include mechanisms for improving processing efficiency that are complementary with one another. As I described in greater detail in the prior section, Tso provides multiple teachings of using partially-transcoded content in order to transcode content in response to client requests. For example, Tso teaches that "both original and transcoded versions of content" are cached for transmission to a client in order to avoid "the need to re-retrieve the content from Internet 18 or to re-transcode the content," teaching that servers use previously-transcoded versions of content to satisfy later client requests and avoid re-performing transcoding that was previously performed. Tso, 4:1-5. Tso also teaches that a client requests a partially-transcoded version of content on which "additional transcoding" is required—teaching that transcode service providers will request partially-transcoded content to satisfy a content request. Tso, 14:57-15:6. Tso also teaches that, when content is retrieved from cache, a "stage" property is used so that the requestor can determine how much the content has already been transcoded, and using this information during the process of determining how to transcode content. Tso, 5:35-42, 6:9-23. Tso thus teaches that a proxy server completes transcoding of partially-transcoded content from another

proxy server, and that proxy servers send, receive, and further transcode partially-transcoded content, for example, as part of a “staged” or “collaborative” transcoding process. *E.g.*, Tso, 4:39-61, 5:35-42, 14:57-15:6, 15:66-16:14. For example, Tso teaches that generated content may need to be converted to greyscale for certain users; when implementing this teaching, partially or fully-transcoded content may be further transcoded “to greyscale for users lacking a color display.” Tso, 17:9-15. These teachings not only render obvious, but actually disclose, retrieving and further transcoding partially-transcoded content from cache. Huang extends these teachings from Tso by providing an explicit teaching of performing such retrieval and further rendering of content that was locally cached. Huang, 6:63-67, 7:42-8:11. Further, Tso teaches that it is advantageous to eliminate “the need to re-retrieve” or “re-transcode” “content.” Tso, 4:1-5. Complementing this disclosure, Huang teaches that by utilizing previously-cached versions of content, including such versions that were “partially rendered,” a server avoids “repeating some object rendering operations.” Huang, 6:63-67, 2:9-13. Thus, whether retrieving a fully transcoded file from the Tso cache and treating as an intermediate version of requested content for further transcoding as suggested by Huang, or retrieving a partially transcoded version (for example, as discussed at Tso, 14:57-15:6) for further transcoding, both paths would provide an obvious improvement to Tso in processing speed and

efficiency as taught by Huang. A person having ordinary skill in the art would have been motivated to modify Tso with this teaching to advantageously allow a server with its own locally-cached and partially-rendered or transcoded copy of content to also perform the rendering (which, as previously explained, encompasses transcoding) process locally on that server. Huang, 7:23-50, 2:8-13; Tso, 4:1-5.

73. **Second**, Huang provides additional detailed teachings of how to track transcoding status of content that allow Huang to effectuate its teachings of using partially-rendered content to respond to user requests for content objects in a manner that complements Tso's teachings of how to retrieve specific versions of content. Tso broadly teaches using the "GetProperties()" and "SetProperties()" calls to track transcoding status by retrieving and setting the properties of content, and teaches that this includes tracking properties describing the status of a "staged" transcoding process. *E.g.*, Tso, 5:36-42, 14:30-36. However, Tso provides only broad teachings of how such information about the transcoding status of content is tracked. Huang provides additional detailed teachings of tracking the rendering (including transcoding) status by using "RHI" information that includes and "object-specific descriptor information" to track transcoding status. Huang, 5:60-6:23, 6:56-58. Huang teaches that when a particular version of content is requested and not available, the server can use a "more detailed version" in cache instead of re-

retrieving the original object to “avoid repeating some object rendering operations.” Huang, 7:23-42, 2:8-13, 6:63-67. The server uses the “RHI” value to track the content available on the server, as well as RHI information associated with a particular content object, including the different transcoding statuses of cached versions of the content. *Id.*; Huang, 6:9-23, 6:56-58, 7:43-64. For example, Huang teaches “stag[ing]” a rendering process by breaking the process into steps, determining which steps to be performed first, and tracking the progress of performing the transcoding steps using RHI information. Huang, 3:29-49, 4:24-52, 6:9-23, 6:56-58, 8:50-9:3. A person having ordinary skill in the art therefore would have found it obvious to modify Tso with Huang’s teachings by using Tso’s GetScaledObject() and GetProperties() calls together with Huang’s teachings of RHI information to find and retrieve content with properties specifying a partially-rendered, “more detailed version” to be further refined into the requested content item, *e.g.*, as part of staged transcoding, and using Tso’s SetProperties() calls to update RHI values, which allows the system to use the RHI to track the transcoding profiles of different content items in a manner that allows for the location and retrieval of partially-transcoded content, thereby advantageously reducing unnecessary repeated retrieval and rendering of content.

74. In light of the above teachings, a person having ordinary skill in the art would have had a reasonable expectation of success when modifying Tso with Huang's teachings of reusing partially-processed content. Indeed, Tso already teaches many of the features described in Huang, including partially processing content to generate requested content, and also teaches mechanisms that are able to take advantage of Huang's teachings, including performing such processing at both servers and clients, and processing content using a "staged" process or using discrete "transcode service providers." Tso, 5:36-42; 12:53-13:11. Further, such use of intermediate content items to minimize the amount of further processing that is required was well understood in the art and known to be advantageous when serving content to clients. For example, U.S. Patent No. 5,420,967, which is assigned to IBM and names Ms. Delp as its first named inventor, a copy of which I understand has been designated as Exhibit 1008, similarly teaches that "intermediate images generated for one requested image can be used by other requested images." Delp, Abstract. Delp teaches a system in which "[e]ach intermediate image corresponds to a change of an image state between an existing image and a requested image," including, for example, intermediate images in which "the scaling of the existing image is a change of image state and will result in the addition of the representation of an intermediate image to the image table." *Id.* As a result of maintaining such

intermediate images, the system can develop an “optimal image path” that minimizes resource consumption to generate a requested image. Delp, 2:6-42. As another example, U.S. Patent No. 5,969,716, which on its face states it was assigned to Interval Research Corporation, and names Mr. Davis as its first named inventor, a copy of which I understand has been designated as Exhibit 1009, teaches a system that “permits pre-existing media content to be reused in a meaningful way.” Davis, 4:2-4. The system uses “descriptive data that represent[s] the content of media signals” to allow the system to compose “new content representations” in audio and video contexts as well by “reus[ing]” preexisting content. Davis, 2:20-67. These examples confirm that it was well understood to reuse previously-transcoded content to generate additional content items, further demonstrating that a person having ordinary skill in the art would have had a reasonable expectation of success when modifying Tso with Huang’s teachings of caching intermediate content. A person having ordinary skill in the art would have understood that modifying Tso to include Huang’s disclosures would require little more than updating the code implementing Tso to carry out the additional functionalities taught by Huang. Modifying Tso with Huang’s teachings would have been a straightforward application of those teachings in a manner that advantageously improves the efficiency of the system.

3. Invalidity of Claim 9 Over Tso in view of Huang

(a) Element [9.pre]

75. Element [9.pre] recites: “A method for accessing dynamically transcoding media content, the method comprising.” This element is met by Tso.

76. To the extent the preamble is limiting, Tso discloses a method for accessing (e.g., providing “information transmitted” between a “network server computer and a network client computer”) dynamically transcoding media content (e.g., “image” or “video” “data content” that has been “dynamically transcode[d]”). See, e.g., Tso:

- 2:44-49 (“Embodiments of the present invention provide the ability to *dynamically transcode information transmitted between, for example, a network server computer and a network client computer.*”)
- 3:45-54 (“Looking more closely at the embodiment in FIG. 3, transcoder 20 is coupled to HTTP remote proxy 36. Parser 22 manages *the transcoding of data to be transmitted from transcoding server 34 to network client 12.* To this end, parser 22 controls transcode service providers 24 to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers 24 may provide the capability to compress and/or scale *different types of data content, such as image, video, or HTML (HyperText Markup Language).*”)

- **8:10-13** (“Applying the above-listed selection criteria, or combinations thereof, embodiments of the present invention may be used to provide a virtually limitless range of *dynamic transcoding services*.”)
- ***See also* Abstract.**

77. **Tso** discloses a method in which “image” or “video” content is “dynamically transcode[d]” to be provided to and accessed by a “client computer.”

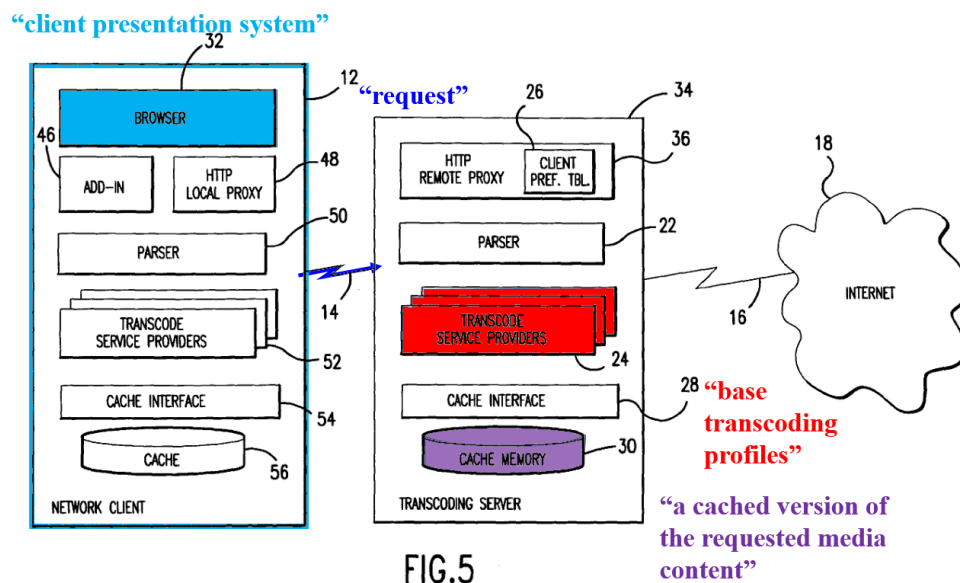
(b) Element [9.a]

78. **Element [9.a]** recites “An act of receiving a request for media content to be delivered to a client presentation system, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content.” This element is met by Tso.

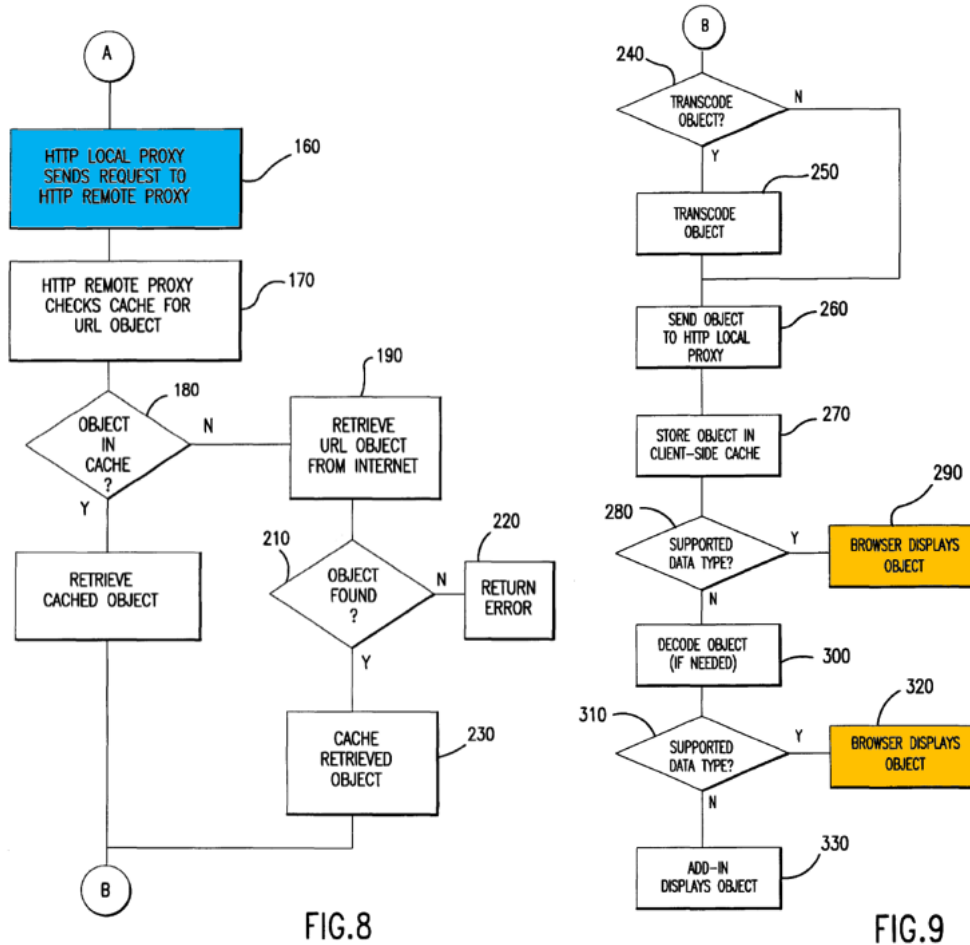
79. **Tso discloses an act of receiving a request for media content** (*e.g.*, “Network client 12 communicates requests for information to” “network server 10” for “image” or “video” “data content”) **to be delivered to a client presentation system** (*e.g.*, “Network client 12” “receives information from” “network server 10” and a “browser displays” the content), **wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media**

content (e.g., “multiple representations of a given cached object” are cached, along with “descriptive information about each representation,” which is “used to determine transcoding properties and transcoding status of a cached object”). *See, e.g., Tso:*

- **Figure 5**



• **Figures 8-9**



- **2:56-65** (“*Network client 12 communicates requests for information to, and receives information from, network server 10* over a client/server communications link 14. Client/server communications link 14 may comprise, for example, a so-called “slow network” using, for example, POTS (Plain Old Telephone System) dial-up technology or wireless connections. Alternatively, client/server communications link 14 may comprise a so-called “fast network,” such as a LAN or WAN (Wide Area

Network), which is capable of operating at much higher speeds than are possible with slow networks.”)

- **3:45-54** (“Looking more closely at the embodiment in FIG. 3, transcoder 20 is coupled to HTTP remote proxy 36. Parser 22 manages *the transcoding of data to be transmitted from transcoding server 34 to network client 12*. To this end, parser 22 controls transcode service providers 24 to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers 24 may provide the capability to compress and/or scale different types of *data content*, such as *image, video*, or HTML (HyperText Markup Language).”)
- **4:1-5** (“Server-side *cache memory 30 may be used to store both original and transcoded versions of content* for later transmission to network client 12 *without the need to re-retrieve the content from Internet 18 or to re-transcode the content.*”)
- **4:62-5:7** (“Unlike most cache memories, server-side cache interface 28 and server-side cache memory 30 enable maintenance of *multiple representations of a given cached object*, with *descriptive information about each representation* included in server-side cache memory 30. In addition, server-side cache interface 28 and server-side cache memory 30

serve as a synchronization point for multi-threaded accesses to cached objects. It should be noted that the illustrated embodiment does not require any particular configuration for server-side cache interface 28 and/or server-side cache memory 30. Indeed, functionality attributed to these components in the various embodiments described herein may be readily implemented in other system components.”)

- **5:35-42** (“The GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. *Transcode service provider 24 may use such information, for example, to determine current compression progress for scaled data access and staged refinements.*”)
- **14:21-55** (“Referring now to FIG. 8, if the requested URL object is not found in client-side cache memory 56, *HTTP local proxy 48 transmits a request for the object to transcoding server 34* using, for example, a Post of a *GetStage(URL, Stage=0)* call (Step 160). Upon receipt of this call, HTTP remote proxy 36 *invokes parser 22, which in turn issues a GetScaledObject[0] call to server-side cache interface 28 to determine whether a non-transcoded version of the requested hypertext object*

already exists in the server-side cache memory 30 (Step 170). If the hypertext object is cached, server-side cache interface 28 issues a GetEntry call to open a stream for the cached object (Step 200). In addition, parser 22 may issue *a GetProperties (URL, ...) call to server-side cache interface 28 to retrieve information about the transcoding properties and transcoded status (such as the refinement level) of the cached object.* “Referring now to FIG. 9, once the requested hypertext object has started to be obtained, *parser 22 determines whether (and how) to transcode the object before transmitting it to network client 12 (Step 240).* Both this decision making process and exemplary transcoding processes are described in detail above. For purposes of the present illustration, assume parser 22 determined that transcoding was appropriate and therefore generated a transcoded version of the requested hypertext object (Step 250).”)

- *See also 4:39-48, 5:8-61, 6:9-28, 10:37-49, 13:39-44, 14:1-36, 14:64-66, 15:1-6, Fig. 7.*

80. Tso discloses that a server receives a “request[] for information” from “network client 12” for “image” or “video” “data content” to be “display[ed]” on the client’s “browser.” Tso, 2:56-65, 3:45-54, Figs. 8-9. Requested content items are stored in a cache that stores “multiple representations of a given cache object.” Tso,

4:64-5:7. The cache also stores “descriptive information” about cached objects that specifies “transcoding properties and transcoding status,” which includes the transcoding “stage” of a cached version of content specifying how much transcoding has been already performed. The cached information is a transcoding profile because the profile maintains descriptive information about the item, including its properties describing the way that it has been transcoded, and the status of transcoding operations that are to be applied to the content item. For example, the “GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. Transcode service provider 24 may use such information, for example, to determine current compression progress for scaled data access and staged refinements.” Tso, 5:36-42. Such information describing the transcoding status is used to retrieve specific content items using a GetScaledObject() call. Tso, 6:9-23. The information stored to service these SetProperties() and GetProperties() calls for a given content item discloses what the ’242 patent claims as a base transcoding profile for each of those cached objects, on top of which additional transcoding is performed for, as an example, staged transcoding. For example, while the ’242 patent does not explicitly define a transcoding profile within its specification, the ’242 patent does describe

“references” to images that maintain similar information to the transcoding-related properties and information taught by Tso, and also recites in claim 10 adding a “new transcoding profile ... corresponding to the identified transcoding parameters,” which is also consistent with these teachings of a profile in Tso. ’242 patent, 8:51-55 (“The media cache database 640 contains references to the created images 645. In a preferred embodiment, the references are the script used to create the image, the names of the images used to create the image, the dates of those files, and the HTML that represents the created image.”), Claim 10; *see also* Tso, 5:36-42, 6:17-23, 4:62-5:7, 4:35-40. A person having ordinary skill in the art would have understood that there are only a limited number of such profiles because only previously-transcoded copies of content are stored in memory: the cache is used to store previously-retrieved, cached, and transcoded items, of which there are only a finite number in any system. *See* Tso, 4:1-5 (“Server-side cache memory 30 may be used to store both original and transcoded versions of content for later transmission to network client 12 without the need to re-retrieve the content from Internet 18 or to re-transcode the content.”).

(c) Element [9.b]

81. **Element [9.b] recites: “at the time of the request, and without input by a network administrator, an act of automatically identifying transcoding**

parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system.” This element is met by Tso.

82. Tso discloses at the time of the request, and without input by a network administrator (e.g., “when network client 12 requests a hypertext object”), an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system (e.g., “[p]arser 22” “manages the transcoding of data to be transmitted” by invoking “transcode service providers”) **wherein identification of transcoding parameters is based on one or more formats of any client presentation system** (e.g., “transcode service providers” are invoked based on “predetermined selection criterion” including characteristics of “network client 12”). See, e.g., Tso:

- **3:45-54** (“Looking more closely at the embodiment in FIG. 3, transcoder 20 is coupled to HTTP remote proxy 36. Parser 22 manages *the transcoding of data to be transmitted from transcoding server 34 to network client 12*. To this end, parser 22 controls transcode service providers 24 to selectively transcode content based on a predetermined selection criterion. For example,

one or more transcode service providers 24 may provide the capability to compress and/or scale *different types of data content, such as image, video, or HTML (HyperText Markup Language).*”)

- **6:9-41** (“*The GetScaledObject() call* is similar to GetObject(), and is also used to request an object from server-side cache memory 30; however, it *adds support for requesting a particular version of that object, such as a high-quality rendition.* Unlike traditional caching proxies, *transcode service providers 24 can use server-side cache memory 30 to store several different versions of an object to support clients with different communications and/or presentation capabilities.* Thus, *an additional “Stage” parameter may be used to indicate which version of the cached object is to be returned to network client 12.* Where transcode service provider 24 is configured to scale network content, *it may use this parameter to request a version of a cached object having, for example, a default scaled quality, a refinement to a better-quality version, or the original non-scaled version.* *In this embodiment, when network client 12 requests a hypertext object, HTTP remote proxy 36 uses either the GetObject() or GetScaledObject() call* (depending on if network client 12 is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser 22.

If the hypertext object is not found, parser 22 uses the CreateEntry() call to *create an entry (in effect, a placeholder) in server-side cache memory 30 for the new object.* The new entry is returned to HTTP remote proxy 36, which requests the hypertext object from Internet 18. As a data stream for the hypertext object is returned, HTTP remote proxy 36 calls parser 22 using the PutObject() call, passing into this call the new entry and the handle to the data stream to be placed into the entry. *Parser 22 selects an appropriate transcode service provider 24 based, for example, on the content type of the data stream.* In this context, *the term content type encompasses a datatype, an HTTP MIME (Multipurpose Internet Mail Extensions) type, a content format, and so on.*")

- **6:64-7:14** (“As noted above, *parser 22 may selectively invoke one of transcode service providers 24 based upon satisfaction of a predetermined selection criterion.* Such selection criterion may comprise, for example, information contained in a server 34, such as a MIME type, a URL (Uniform Resource Locator), a last modified time indicator and so on. Alternatively, *the predetermined selection criterion may comprise information contained in a data portion of such a data packet,* such as particular content, key words, structures (for example, heading levels), and so on. Still further, *the*

predetermined selection criterion may comprise a condition of the device on which transcoding server 34 is installed (for example, a current processing load), a condition of a device to which transcoding server 34 is coupled, or a condition of a communications link. Transcoding server 34 may provide the ability to dynamically update such predetermined selection criteria.”)

- 7:15-29 (“*The following discussion provides still more examples of the types of information which may be used to dictate which of transcode service providers 24 are invoked.* It should be noted, however, that these examples are provided by way of illustration only, and are not intended to limit in any way the scope of the invention claimed herein. *The predetermined selection criterion may comprise: (1) network client 12, such as a display dimension, resolution, number of colors, processor type, memory/disk configuration, modem or network interface type, installed add-in boards* (for example, hardware compression/decompression), *software configuration* (for example, availability of pre-installed software decompression modules), *physical location/proximity* (for example, as determined by a telephone area code), *and user identity....*”);
- *See also 5:62-67, 6:64-7:14, 12:55-59.*

83. See my discussion of element [9.a] for further discussion on Tso's disclosures of this element. **Tso** also discloses that when a "network client" requests a "hypertext object," the system automatically identifies the "transcode service providers" used to "selectively transcode content" based on "selection criteri[a]" associated with the request. Tso, 3:45-54, 6:64-7:14. This process is controlled by a "parser" and does not require input by a network administrator. *See id.* The "predetermined selection criterion" include formats of "network client 12, such as a display dimension, resolution, number of colors, processor type," and other client formats. Tso, 7:15-29.

(d) Element [9.c]

84. **Element [9.c] recites "an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system."** This element is met by Tso.

85. **Tso discloses an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system (*see* [9.b]) are the same as transcoding parameters that are being applied to the requested media content (*e.g.*, determining that**

“multiple network clients 12” have made “requests for the same hypertext object”) **prior to delivery to another client presentation system** (e.g., prior to delivering “appropriate versions” of the content to both clients “concurrently”). *See, e.g., Tso:*

- **6:51-63** (“Multiple-thread processing improves the efficiency of the present embodiment by not waiting for a hypertext object to be received in its entirety by HTTP remote proxy 36, or added in its entirety to server-side cache memory 30, before beginning to send the object to network client 12.

Another benefit of multiple-thread processing is that parser 22 may efficiently process requests for the same hypertext object from multiple network clients 12. The hypertext object need only be retrieved from Internet 18 once, and appropriate versions may be transmitted to such multiple network clients 12 concurrently. The hypertext object need only be retrieved from Internet 18 once, and appropriate versions may be transmitted to such multiple network clients 12 concurrently. It should be noted, however, that embodiments of the present invention may be implemented without multiple-thread processing.”)

- **10:22-27** (“HTTP remote proxy 36 may also inform parser 22 of any applicable user preferences (e.g., from client preference table 26). Upon being invoked, *parser 22 first calls cache interface 28 with the requested*

hypertext object to determine whether a copy of the required version already resides in server-side cache memory 30.”)

- *See also 4:62-5:18, 5:42-51, 6:9-28, 6:45-50, 14:64-15:6.*

86. See my discussion of elements [9.a] and [9.b] for further discussion on Tso’s disclosures of this element. **Tso** also discloses that each time a proxy receives a request for a content item, the proxy first checks whether a “copy of the required version” is available in cache. Tso, 10:22-27. As I discussed for [9.b], the “required version” is generated using transcoding parameters specified using “selection criteri[a].” *See* [9.b]. When using “multiple-thread processing” to process “requests for the same hypertext object from multiple network clients,” the server determines that the two clients are requesting the same content; the content is therefore “retrieved from Internet 18 once” to handle both requests. Tso, 6:51-61. The system determines “appropriate versions” for both clients and, if the same request came from two similar client systems, the “appropriate versions” for both clients are the same version, and therefore the same version is generated using the same transcoding parameters. *Id.* As a result, in the same way that Tso teaches that multiple-thread processing results in the same content being retrieved “once” in response to both requests, when the same transcoded version is to be generated, meaning that the “appropriate versions” of content to provide to two clients are both

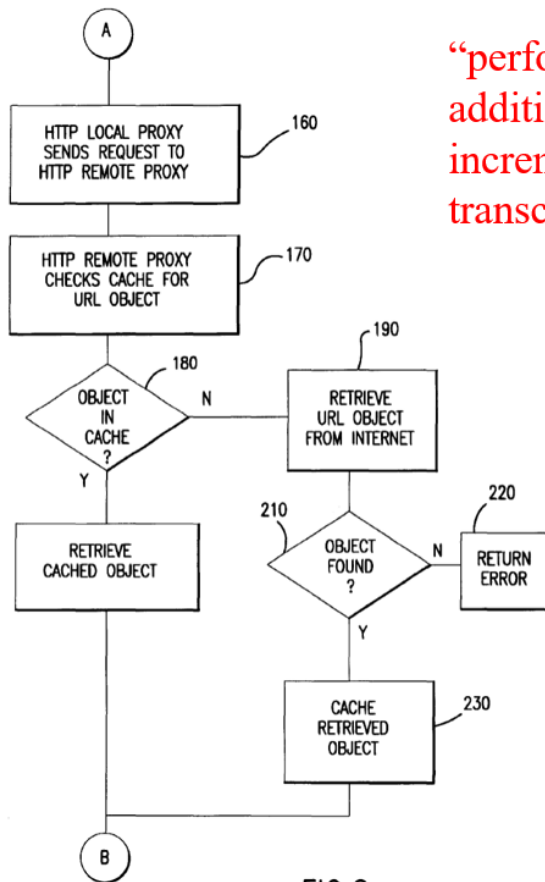
the same version, Tso not only renders obvious that the system determines that the content is already being transcoded according to specified parameters in response to the earlier request, but affirmatively discloses it. *See id.* The server then transmits that same version “to such multiple network clients 12 concurrently.” Tso, 6:51-61, 7:60-8:9.

(e) Element [9.d]

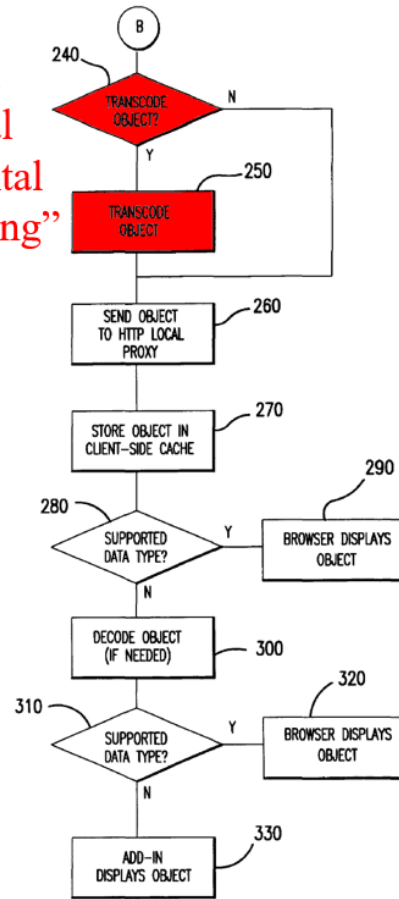
87. **Element [9.d] recites “an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile.”** This element is met by the combined teachings of Tso and Huang.

88. **Tso discloses an act of transcoding the requested media content in accordance with the identified transcoding parameters** (*e.g.*, “transcod[ing] the object before transmitting it to network client 12” using “an appropriate transcode service provider 24” that is “invoke[d] ... based upon satisfaction of a predetermined selection criterion”). *See, e.g.*, Tso:

- Figures. 8 and 9



“perform additional incremental transcoding”



- 5:35-42 (“The GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. *Transcode service provider 24 may use such information, for example, to determine current compression progress for scaled data access and staged refinements.*”)

- **6:9-41** (“The GetScaledObject() call is similar to GetObject(), and is also used to request an object from server-side cache memory 30; however, it adds support for requesting a particular version of that object, such as a high-quality rendition. Unlike traditional caching proxies, transcode service providers 24 can use server-side cache memory 30 to store several different versions of an object to support clients with different communications and/or presentation capabilities. Thus, *an additional “Stage” parameter may be used to indicate which version of the cached object is to be returned to network client 12.* Where transcode service provider 24 is configured to scale network content, it *may use this parameter to request a version of a cached object having, for example, a default scaled quality, a refinement to a better-quality version,* or the original non-scaled version.

In this embodiment, when network client 12 requests a hypertext object, HTTP remote proxy 36 uses either the GetObject() or GetScaledObject() call (depending on if network client 12 is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser 22. If the hypertext object is not found, parser 22 uses the CreateEntry() call to *create an entry (in effect, a placeholder) in server-side cache memory 30 for the new object.* The new entry is returned to HTTP remote proxy 36, which requests

the hypertext object from Internet 18. As a data stream for the hypertext object is returned, HTTP remote proxy 36 calls parser 22 using the PutObject() call, passing into this call the new entry and the handle to the data stream to be placed into the entry. *Parser 22 selects an appropriate transcode service provider 24 based, for example, on the content type of the data stream.* In this context, *the term content type encompasses a datatype, an HTTP MIME (Multipurpose Internet Mail Extensions) type, a content format, and so on.*")

- 7:15-29 (“*The following discussion provides still more examples of the types of information which may be used to dictate which of transcode service providers 24 are invoked.* It should be noted, however, that these examples are provided by way of illustration only, and are not intended to limit in any way the scope of the invention claimed herein. *The predetermined selection criterion may comprise: (1) network client 12, such as a display dimension, resolution, number of colors, processor type, memory/disk configuration, modem or network interface type, installed add-in boards (for example, hardware compression/decompression), software configuration (for example, availability of pre-installed software*

decompression modules), *physical location/proximity* (for example, as determined by a telephone area code), *and user identity....*”)

- **14:21-55** (“Referring now to FIG. 8, if the requested URL object is not found in client-side cache memory 56, HTTP local proxy 48 transmits a request for the object to transcoding server 34 using, for example, a Post of a *GetStage(URL, Stage=0)* call (Step 160). Upon receipt of this call, HTTP remote proxy 36 *invokes parser 22, which in turn issues a GetScaledObject[()]call to server-side cache interface 28 to determine whether a non-transcoded version of the requested hypertext object already exists* in the server-side cache memory 30 (Step 170). If the hypertext object is cached, server-side cache interface 28 issues a GetEntry call to open a stream for the cached object (Step 200). In addition, parser 22 may issue a *GetProperties (URL, . . .) call to server-side cache interface 28 to retrieve information about the transcoding properties and transcoded status (such as the refinement level) of the cached object*. Referring now to FIG. 9, once the requested hypertext object has started to be obtained, *parser 22 determines whether (and how) to transcode the object before transmitting it to network client 12 (Step 240)*. Both this decision making process and exemplary transcoding processes are described in detail above.

For purposes of the present illustration, assume parser 22 determined that transcoding was appropriate and therefore generated a transcoded version of the requested hypertext object (Step 250).)

- *See also 5:20-31, 6:64-7:3, 12:53-13:5, 14:59-15:6, 15:66-16:14.*

89. **Huang further discloses that the identified transcoding parameters are used to perform additional incremental transcoding** (*e.g.*, “perform the rendering locally” if “further rendering is necessary”) **on top of the base transcoding profile** (*e.g.*, use the “RHI” of an object in “local cache” “to see if any further rendering is necessary” before “determin[ing] whether or not” to perform “further rendering”). *See, e.g.*, Huang:

- **Abstract** (“The criteria for such a method can include the bandwidth and current load of the network links among proxies, and/or the respective CPU usage of the proxies. If an *object rendering can be staged, e.g., different resolution rendering, it can be performed by more than one of the proxies.* The determination of *which proxy performs which stage of the multistage rendering can also be adaptive* to the dynamic load conditions, as well as network conditions.”)
- **4:8-15** (“*At an intermediate node having a less detailed version of the requested object the method includes such information in the RHI,*

forwards the request to another node, and at another intermediate node that has a more detailed version of the requested object, *the node decides* whether to return the more detailed version of the requested object, without further local rendering, or *to instead perform some rendering and return a partially rendered object*, or to instead return a completely rendered object.”)

- **6:9-23** (“When a requested object passes through the proxy network, *any proxy server 110, 111, 112 can perform a complete or partial rendering based on the associated RHI*. For example, *if the entire rendering process can be partitioned into two or more steps, a given one of the proxy servers (e.g., 110) may decide to perform only one of the rendering steps*, and to then forward the partially rendered object to another proxy server (e.g., 111). *The proxy server 110 also modifies the RHI to reflect the processing that it performed*, and forwards the modified RHI as well to the proxy server 111.”)
- **6:56-58** (“An object renderer 206, which performs object rendering according to the *RHI associated with a particular object*, may also be included in the proxy server logic 203.”)

- **6:63-67** (“*Proxy server logic 203* may also include a cache *manager 207* which maintains a local copy of the partially rendered or completely rendered objects in order to avoid repeating some object rendering operations with the same proxy server.”)
- **7:43-64** (“If a copy of the requested object can be found in the local cache, at Step 402, *the proxy server checks the cached object against the RHI to see if any further rendering is necessary*. Note that the RHI contains the capability specification of the receiving device (i.e. the device that originally requested the object that was just found in the cache). *By checking the RHI, the proxy server 110, 111, 112 can determine if any further rendering is necessary*. If no further rendering is necessary, the proxy server modifies the RHI to indicate its local condition for providing rendering services and returns the object at step 403. *If further rendering is found to be necessary, based on the RHI or the requesting device, then the proxy server executes at step 405 a selection function to determine whether or not it wishes to perform the rendering locally*. If the proxy server decides not to perform any rendering locally, the proxy server modifies the RHI to indicate its local load condition for providing such rendering services and returns the object along with the modified RHI at step 406. *If the proxy server instead decides*

to perform the rendering locally, it checks the RHI at step 407 to determine if it wishes to complete the entire rendering process, or just some part of the required rendering process.”)

- **10:56-11:5** (“*If, however, for some reason the proxy server elects to not completely render the image object, or to not render the image object at all, due to, for example, loading considerations or a lack of suitable software, then the PICS label of the image object will not reflect a condition compatible with the display capabilities of the PDD. For example, assume that a given one of the proxy servers 110, 111, 112 elects to only modify the color encoding of the received image object from 16 level to 1, then the modified PICS label as received by a next proxy server will be r(c 1 s 1000), which is a form still not compatible with the PDDs’ RHI of d(c 1 s 2). The next proxy server 110, 111, 112 may then elect to render the received image object to reduce the imageSize from 1000 megabytes to 2 megabytes, resulting in the modified PICS label of r(c 1 s 2), which is a form that is compatible with the PDDs RHI of d(c 1 s 2).*”)
- **See also 10:20-34.**

90. See my discussion of elements [9.a], [9.b], and [9.c] for further discussion on Tso’s disclosures of this element. Additionally, **Tso** also teaches that,

when a user issues a request for content, the server uses a “GetScaledObject()” call to retrieve a particular version of content from the cache. Tso, 6:9-41. **Tso** further teaches that cached content has properties including the “current compression progress for scaled data access and staged refinements.” Tso, 5:36-42. Once the content item is retrieved, the server then performs transcoding based on transcoding parameters specified by the “selection criterion” in the request. Tso, 7:15-29, 14:47-55.

91. Further to this, **Huang** teaches that when the RHI information on the system includes a profile indicating that “a copy of the requested object can be found in the local cache,” the system retrieves that copy and, using the RHI information “associated with a particular object,” checks whether “any further rendering is necessary” and, if further rendering is deemed necessary, performs “further rendering” on top of the stored base transcoding profile (in other words, the “more detailed version” that has all of the required detail retained in the data such that it is available to be further transcoded (*e.g.*, by cropping, reducing resolution, etc.—thereby reducing the “detail” as part of the transcoding process)) as required to the copy retrieved from cache. Huang, 7:23-42, 2:8-13, 6:56-58, 6:63-67. The system then updates the associated RHI to reflect that the content has been further transcoded. Huang, 6:9-23.

92. As I discussed in Section IX.A.2, a person having ordinary skill in the art would have been motivated to modify **Tso** with these teachings to advantageously reduce repetitive transcoding and retrieval of content while tracking the available versions of the content, yielding a system that retrieves a partially-transcoded version of content using a `GetScaledObject()` call and then incrementally further transcodes the content item based on a cached base transcoding profile specified using RHI information about particular objects. *Tso*, 6:9-41; *Huang*, 7:43-57.

(f) Element [9.e]

93. **Element [9.e] recites “wherein the act of transcoding the requested media content in accordance with the identified transcoding parameters comprises: an act of selecting a pre-existing base transcoded version of the requested media content comprising intermediate derivative media that has been transcoded in accordance with only a portion of the identified transcoding parameters.”** This element is met by the combined teachings of *Tso* and *Huang*.

94. **Tso discloses wherein the act of transcoding the requested media content is in accordance with the identified transcoding parameter (*see* [9.d]) comprises: an act of selecting a pre-existing base transcoded version of the requested media content (*e.g.*, an act of “requesting a particular version of [the requested] object” using a “Stage” parameter that is tracked with the “descriptive**

information” stored about each content item) **comprising derivative media that has been transcoded** (e.g., comprising “a version of a cached object having, for example, a default scaled quality” that can be “refine[d] to a better-quality version” as part of “staged” refinement). *See, e.g., Tso:*

- **4:62-5:7** (“Unlike most cache memories, server-side cache interface 28 and server-side cache memory 30 enable maintenance of *multiple representations of a given cached object*, with *descriptive information about each representation* included in server-side cache memory 30. In addition, server-side cache interface 28 and server-side cache memory 30 serve as a synchronization point for multi-threaded accesses to cached objects. It should be noted that the illustrated embodiment does not require any particular configuration for server-side cache interface 28 and/or server-side cache memory 30. Indeed, functionality attributed to these components in the various embodiments described herein may be readily implemented in other system components.”)
- **5:35-42** (“The GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. Transcode service provider 24 may

use such information, for example, to *determine current compression progress for scaled data access and staged refinements.*”)

- **6:9-41** (“*The GetScaledObject() call* is similar to GetObject(), and is also used to request an object from server-side cache memory 30; however, it *adds support for requesting a particular version of that object, such as a high-quality rendition.* Unlike traditional caching proxies, *transcode service providers 24 can use server-side cache memory 30 to store several different versions of an object to support clients with different communications and/or presentation capabilities.* Thus, *an additional “Stage” parameter may be used to indicate which version of the cached object is to be returned to network client 12.* Where transcode service provider 24 is configured to scale network content, *it may use this parameter to request a version of a cached object having, for example, a default scaled quality, a refinement to a better-quality version, or the original non-scaled version.* *In this embodiment, when network client 12 requests a hypertext object, HTTP remote proxy 36 uses either the GetObject() or GetScaledObject() call* (depending on if network client 12 is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser 22. If the hypertext object is not found, parser 22 uses the CreateEntry() call to

create an entry (in effect, a placeholder) in server-side cache memory 30 for the new object. The new entry is returned to HTTP remote proxy 36, which requests the hypertext object from Internet 18. As a data stream for the hypertext object is returned, HTTP remote proxy 36 calls parser 22 using the PutObject() call, passing into this call the new entry and the handle to the data stream to be placed into the entry. *Parser 22 selects an appropriate transcode service provider 24 based, for example, on the content type of the data stream.* In this context, *the term content type encompasses a datatype, an HTTP MIME (Multipurpose Internet Mail Extensions) type, a content format, and so on.*")

- **14:59-63** ("In addition, client-side parser 50 *determines whether any further processing is required before the hypertext object is rendered* (e.g., a new MIME type has been established by transcoding server 34) (Step 280).")
- **15:66-16:14** ("In view of the foregoing description, it should be apparent that it is possible for there to be more than one so-called "smart" proxy arranged between a client device and a content server device. If left unchecked, such a condition can result in content being altered excessively (for example, too many ads inserted, multiple lossy compressions resulting

in indecipherable images). To address this problem, an embodiment of the present invention may use *a special proxy-to-proxy protocol that extends the existing request/ response structures to indicate whether and what sort of transcoding has already been performed on the content*. Such a *specialized protocol*, in addition to other proxy-to-proxy messages which may be implemented on an as-needed basis, *enables multiple proxies to work collaboratively*, yet still transparently to users, client software, existing “standard” proxies and content servers.”).

95. **Huang further discloses selecting a pre-existing base transcoded version of the requested media content** (e.g., “determin[ing] if the requested object is available in the cache” in the form of “a local copy of the partially rendered ... object”) **comprising intermediate derivative media that has been transcoded in accordance with only a portion of the identified transcoding parameters** (e.g., comprising a “partially rendered” media item for which the “RHI” information indicates that “only one” of “two or more” steps is performed and “further rendering is necessary”). *See, e.g., Huang:*

- **4:8-15 (“At an intermediate node having a less detailed version of the requested object the method includes such information in the RHI,**

forwards the request to another node, and at another intermediate node that

has a more detailed version of the requested object, the node decides whether to return the more detailed version of the requested object, without further local rendering, or to instead perform some rendering and return a partially rendered object, or to instead return a completely rendered object.”)

- **6:9-23** (“*When a requested object passes through the proxy network, any proxy server 110, 111, 112 can perform a complete or partial rendering based on the associated RHI.* For example, if the entire rendering process can be partitioned into two or more steps, a given one of the proxy servers (e.g., 110) may decide to perform only one of the rendering steps, and to then forward the partially rendered object to another proxy server (e.g., 111). *The proxy server 110 also modifies the RHI to reflect the processing that it performed, and forwards the modified RHI as well to the proxy server 111.*”);
- **6:56-67** (“*An object renderer 206, which performs object rendering according to the RHI associated with a particular object,* may also be included in the proxy server logic 203. Object renderer may be a computer program which renders, by example, a color image into a black-and-white image, or one that reduces a complex HyperText Markup Language (HTML) text into a simple HTML text containing only a summary of the HTML

headers. *Proxy server logic 203 may also include a cache manager 207 which maintains a local copy of the partially rendered or completely rendered objects* in order to avoid repeating some object rendering operations with the same proxy server.”)

- **7:25-36** (“At Step 401, the object request handler 205 *checks with the cache manager 207 to determine if the requested object is available in the cache.*

It should be noted that *the cache* may contain a less detailed version of the requested object, or it *may contain a more detailed version. A less detailed version* of the object does not satisfy the requirement and a request for the object must be sent out, typically to the appropriate one of the content servers 120, 121 or to another proxy server. However, *a more detailed version of the object may be further rendered by the proxy server 110, 111, 112 in order to satisfy the request.*”)

- **7:43-64** (“If a copy of the requested object can be found in the local cache, at Step 402, *the proxy server checks the cached object against the RHI to see if any further rendering is necessary.* Note that the RHI contains the capability specification of the receiving device (i.e. the device that originally requested the object that was just found in the cache). *By checking the RHI, the proxy server 110, 111, 112 can determine if any further rendering is*

necessary. If no further rendering is necessary, the proxy server modifies the RHI to indicate its local condition for providing rendering services and returns the object at step 403. *If further rendering is found to be necessary, based on the RHI or the requesting device, then the proxy server executes at step 405 a selection function to determine whether or not it wishes to perform the rendering locally*. If the proxy server decides not to perform any rendering locally, the proxy server modifies the RHI to indicate its local load condition for providing such rendering services and returns the object along with the modified RHI at step 406. *If the proxy server instead decides to perform the rendering locally, it checks the RHI at step 407 to determine if it wishes to complete the entire rendering process, or just some part of the required rendering process.*”)

- **9:5-7** (“At step 501, when a proxy server receives an object, it *first tests the associated RHI to determine if further rendering is necessary.*”)
- **10:46-56** (“It can be appreciated that a proxy server 110, 111, 112 that receives an image object having the above-noted PICS label $r(c\ 16\ s\ 1000)$, in response to a request from the PDD having the above-noted RHI $d(c\ 1\ s\ 2)$, will be informed that the PDD is incapable of displaying the image object as received, and that *the image object will need to be rendered into a form*

that the PDD is capable of displaying. The proxy server may perform the entire rendering process, and will then modify the PICS label of the image object to be r(c 1 s 2), i.e., to indicate a format compatible with the requesting PDD's display capabilities.”)

- *See also 11:15-55.*

96. See my discussion of elements [9.a], [9.b], [9.c], and [9.d] for further discussion on Tso’s disclosures of this element. **Tso** also discloses that the cache stores “several different versions” of the media content items, each of these versions is stored with “descriptive information” that describes a pre-existing base transcoded version of the requested media content. Tso, 6:9-23. **Tso** further discloses that the cached versions may be used for “staged refinements” such that the content is cached with parameters specifying how much transcoding has been applied already, and how much remains to be applied. Tso, 5:36-42 (“The GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. Transcode service provider 24 may use such information, for example, to determine current compression progress for scaled data access and staged refinements.”). When receiving a request for media content, **Tso** discloses that the system requests a

“particular version of the object” such as a “staged” “refinement” of the content, which is a derivative version of the original media content that has already been transcoded, using a GetScaledObject() call. Tso, 6:9-23.

97. See my discussion of element [9.d] for further discussion on Huang’s disclosures of this element. **Huang** also discloses that a cache stores versions of content that are “partially” or “completely” rendered (which includes transcoded, as I discussed in Section IX.A.2). Huang, 6:63-67. When a server receives a request for content, the cache manager retrieves a “more detailed version” that “may be further rendered ... in order to satisfy the request.” Huang, 7:43-57. Each content object is stored with “RHI” information about that content object specifying information including the “processing” that has already been “performed” on the content, including for example whether only one of “two or more steps” of the rendering process has been performed on the content item. Huang, 6:9-23, 6:56-67. For example, the RHI may indicate that the content has been transcoded with only one of several “steps” of transcoding, and still requires completion of final transcoding steps to cause the content to be “rendered into a form” that the client “is capable of displaying.” Huang, 6:9-23, 6:56-67, 10:46-56.

98. As discussed in Section IX.A.2 and for [9.d], A person having ordinary skill in the art would have been motivated to modify **Tso** with this teaching of further

rendering partially-rendered content, and tracking the rendering status of content using RHI information, to advantageously avoid unnecessarily re-transcoding of content, yielding a system that uses a `GetScaledObject()` call to request a “more detailed,” partially rendered, derivative version of the original media content stored in cache for which further transcoding is required, for example, to take content that has undergone “staged” transcoding, and complete the final stage of transcoding to convert the content to a form the client is capable of displaying. Huang, 7:43-57.

(g) Element [9.f]

99. **Element [9.f] recites “an act of creating a final version by incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.”** This element is met by the combined teachings of Tso and Huang.

100. **Huang discloses an act of creating a final version by incrementally performing further transcoding of the pre-existing base transcoded version** (*e.g.*, “complet[ing] the entire rendering process” to produce a “completely rendered... object”) **in accordance with a remaining portion of the identified transcoding parameters** (*e.g.*, in accordance with “further rendering [that] is found to be necessary” for a “cached object”). *See, e.g.*, Huang:

- **Figure 5:**

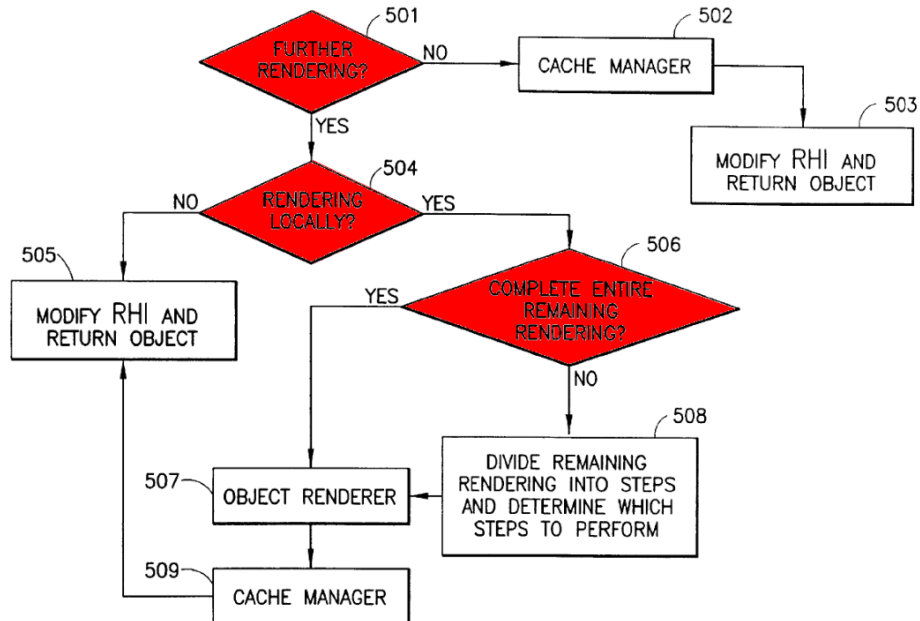


FIG.5

- **6:9-23** (“When a requested object passes through the proxy network, *any proxy server 110, 111, 112 can perform a complete or partial rendering based on the associated RHI*. For example, *if the entire rendering process can be partitioned into two or more steps, a given one of the proxy servers (e.g., 110) may decide to perform only one of the rendering steps*, and to then forward the partially rendered object to another proxy server (e.g., 111). *The proxy server 110 also modifies the RHI to reflect the processing that it performed*, and forwards the modified RHI as well to the proxy server 111.”)

- 6:63-67 (“*Proxy server logic 203 may also include a cache manager 207 which maintains a local copy of the partially rendered or completely rendered objects in order to avoid repeating some object rendering operations with the same proxy server.*”)
- 7:43-64 (“If a copy of the requested object can be found in the local cache, at Step 402, *the proxy server checks the cached object against the RHI to see if any further rendering is necessary.* Note that the RHI contains the capability specification of the receiving device (i.e. the device that originally requested the object that was just found in the cache). *By checking the RHI, the proxy server 110, 111, 112 can determine if any further rendering is necessary.* If no further rendering is necessary, the proxy server modifies the RHI to indicate its local condition for providing rendering services and returns the object at step 403. *If further rendering is found to be necessary, based on the RHI or the requesting device, then the proxy server executes at step 405 a selection function to determine whether or not it wishes to perform the rendering locally.* If the proxy server decides not to perform any rendering locally, the proxy server modifies the RHI to indicate its local load condition for providing such rendering services and returns the object along with the modified RHI at step 406. *If the proxy server instead decides*

to perform the rendering locally, it checks the RHI at step 407 to determine if it wishes to complete the entire rendering process, or just some part of the required rendering process.”)

- **8:1-11** (“In either case, and *after making the decision to perform local rendering at step 405, the proxy server 110, 111, 112 calls the object renderer 206 to perform the object rendering at Step 408. After the rendering process is complete* (either a complete or partial rendering), the cache manager 207 is called at step 410 to determine whether or not to cache a copy of the rendered object locally. The proxy server 110, 111, 112 then modifies the RHI at step 406 to reflect its local condition and returns the rendered (completely rendered or partially rendered) object along with the modified RHI.”)
- **9:4-14** (“FIG. 5 is a flow chart diagram illustrating the object handler 204 of the proxy server logic 203. At step 501, when a proxy server receives an object, it first tests the associated RHI to determine if further rendering is necessary. If not, it passes the object to the cache manager 207 for caching consideration at step 502. At step 503, *the received object is returned to the requesting client device 130, 131, or to another proxy server 110, 111, 112*

on the path to the receiving client device. In returning the object, the proxy server may modify the associated RHI to indicate its local condition.”)

- **10:46-56** (“It can be appreciated that a proxy server 110, 111, 112 that receives an image object having the above-noted PICS label $r(c\ 16\ s\ 1000)$, in response to a request from the PDD having the above-noted RHI $d(c\ 1\ s\ 2)$, will be informed that the PDD is incapable of displaying the image object as received, and that the image object will need to be rendered into a form that the PDD is capable of displaying. *The proxy server may perform the entire rendering process, and will then modify the PICS label of the image object to be $r(c\ 1\ s\ 2)$, i.e., to indicate a format compatible with the requesting PDD’s display capabilities.*”)
- *See also 4:8-15.*

101. See my discussion of elements [9.d] and [9.e]. In addition to **Tso’s** teachings of retrieving a partially-rendered version of content using a GetScaledObject() call (and using such a call as rendered obvious by **Tso** in view of **Huang**) as discussed for [9.e], **Huang** further teaches that when a server retrieves a partially-rendered content object from a local cache, as I also discussed in [9.e], the server determines that “further rendering” is necessary, and then will “complete the entire rendering process” in order to “return[] the ... completely rendered ... object”

“to the requesting client device.” Huang, 7:43-57, 8:1-11, 9:4-14, 10:46-56. In light of the fact that Huang teaches that the rendering process includes “two or more” steps, some of which have already been performed, “completing” the rendering process includes completing the remaining rendering (transcoding) steps that had not yet been performed. *Id.*

102. As I discussed for [9.e] and in Section IX.A.2, a person having ordinary skill in the art would have been motivated to modify **Tso** with these teachings from Huang teaching to advantageously reduce unnecessary re-transcoding of content, yielding a system in which, once an intermediate content item is retrieved by using a `GetScaledObject()` call, further transcoding is performed based on the retrieved content to generate the completely rendered object. Tso, 4:39-61, 14:57-15:6; Huang, 7:43-57, 8:1-11, 9:4-14.

(h) Element [9.g]

103. **Element [9.g] recites “An act of delivering the transcoded media content to both client presentation systems concurrently.”** This element is met by Tso.

104. **Tso discloses an act of delivering the transcoded media content to both client presentation systems concurrently (e.g., “transmit[ing]” “appropriate**

versions” of “a hypertext object” “to such multiple network clients 12 concurrently”). *See, e.g., Tso:*

- **6:51-63** (“Multiple-thread processing improves the efficiency of the present embodiment by not waiting for a hypertext object to be received in its entirety by HTTP remote proxy 36, or added in its entirety to server-side cache memory 30, before beginning to send the object to network client 12. Another benefit of multiple-thread processing is that *parser 22 may efficiently process requests for the same hypertext object from multiple network clients 12*. The hypertext object need only be retrieved from Internet 18 once, and *appropriate versions may be transmitted to such multiple network clients 12 concurrently*. It should be noted, however, that embodiments of the present invention may be implemented without multiple-thread processing.”)
- *See also 4:1-5, 5:25-35, 5:42-61, 6:13-23, 6:37-50, 10:37-49, 12:53-67, 14:47-63.*

105. See my discussion of elements [9.c] and [9.f]. **Tso** also discloses that, using multiple-thread processing, the system is able to “efficiently process requests for the same hypertext object from multiple network clients” and, once the content

has been transcoded properly, the appropriate version of the content is “transmitted to such multiple network clients 12 concurrently.” Tso, 6:51-63, 7:60-8:9.

B. Grounds 2 and 3: Samaniego Anticipates and Renders Obvious Claim 9

1. Overview of Samaniego

106. As previously stated in Section I, Samaniego is a publication of U.S. Patent Application No. 09/929,904. Although the face of the '242 patent identifies the '242 patent as a “continuation-in-part” of the Samaniego application, none of the inventors named on the '242 patent are named as inventors on Samaniego. I have therefore been asked to assume that Samaniego qualifies as prior art to the '242 patent and, based on that assumption, consider whether Samaniego alone and alternatively in view of other prior art discloses and renders obvious the Challenged Claim. I have not formulated any separate opinion as to whether Samaniego qualifies as prior art.

107. I have reviewed Samaniego and compared the language in its specification to the language in the '242 patent. Based on my review, the text of the specifications of the '242 patent and Samaniego do not have any substantive differences, as the only changes to the text are syntactical edits and a paragraph setting forth priority claims for the '242 patent. I have prepared a redline document showing the language that was changed in the '242 patent's specification, which I

understand has been designated as Exhibit 1013. With respect to figures, a comparison of Samaniego's and the '242 patent's figures shows that '242 patent's figures are formalized versions of the figures that are in Samaniego. *Compare* '242 and Samaniego Figs. 1-23. I have not identified any material in the '242 patent's figures that disclose material that is material to my opinions with respect to Challenged Claim 9 and that is not already disclosed in Samaniego. As a result, if the '242 patent's specification supports Challenged Claim 9, then Samaniego anticipates that Claim because it has these same disclosures. However, for certain elements, I have also considered Samaniego's teachings in combination with other prior art teachings in formulating my opinions as set forth in Ground 4 below.

2. Invalidity of Claim 9 in View of Samaniego

108. In light of the fact that Samaniego's specification is substantially identical to the '242 patent's specification, in my analysis below, I have color coded the pertinent sections of Samaniego's specification to show the specification sections corresponding to the claim language of Challenged Claim 9, and have provided additional explanation as helpful to explain a person having ordinary skill in the art's understanding of the disclosures in Samaniego.

(a) Element [9.pre]

109. As set forth in the table below, Samaniego discloses this element:

'242	Samaniego
[9.pre] A method for accessing dynamically transcoding media content , the method comprising:	• [0125] (“A preferred embodiment of the method scripts the image creation process in HTML to create a dynamic Web site.”) • <i>See also</i> [0094], [0143], [0143]-[0144], [0147]-[0148], [0151]-[0152], [0155], [0157], Figs. 21-22.

110. Samaniego discloses a “preferred embodiment” that scripts “image creation” to “create a dynamic Web site”—such dynamic image creation discloses this element. Samaniego, [0125].

(b) Element [9.a]

111. As set forth in the table below, Samaniego discloses this element:

'242	Samaniego
[9.a] an act of receiving a request for media content to be delivered to a client presentation system, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;	Fig. 6² [0103] (“If the image has not been created, the media caching subsystem 620 hands the HTML tag to a media creation subsystem 630. The media creation subsystem 630 returns an image to the media caching subsystem 620. The media caching subsystem 620 adds the created image and the HTML tag to a media cache database 640.”) [0104] (“The media cache database 640 contains references to the created images 645. In a preferred embodiment, the references are the script used to create the image, the names of the images used to create the image, the dates of those files, and the HTML that represents the created image. The media caching subsystem 620 performs lookups in this

² Although readable, several of the items in Figure 6 are hard to read. For ease of reference, Item 120 states “Browser.” Item 610 states “HTML Parser.” Item 630 states “Media Creation System.” Item 640 states “Media Cache Database.” Item 650 has handwritten text stating “Media Script.” And item 660 states “Media Repository.”

'242	Samaniego
	database to determine if the image has been created. If the image has not been created the media caching subsystem 620 calls upon the media creation subsystem 630 to create the image and then store the results in the media cache database 640.") • [0108] ("A user requests a Web page from a Web browser 120.") • [0138] ("The delivery of an HTML document and media according to a preferred embodiment of the invention is described with reference to FIG. 18. Resource locators (URLs) are placed within HTML documents 301 accessible to the Web server 110. These URLs direct browsers to generate requests for media to the system 100.") [0151] ("The final media is passed to a user profile system 2160 wherein browser characteristics specified through proprietary tags within the URL are inspected, and appropriate modification to the media is performed, based on such characteristics. The resulting image is handed to the media cache system 2120 for caching and returned to the browser 120.") • <i>See also</i> [0097], [111], [0133], [0138], [0148], [0151]-[0152], [0154], Figs. 7, 10, 21-22.

112. As annotated in Figure 6 in the above table, **Samaniego** discloses that a user device with a browser issues a request for a web page, and this web page has embedded URLs. Samaniego, [0108], [0138]. Those URLs "direct" a user's "**browser**" to "**generate requests for media content.**" Samaniego, [138]. **Cached versions of the requested media content are stored in a "media cache database."**

Samaniego, [0103]-[0104], Fig. 6; *see also* Figs. 7, 10.³ The cache database also stores **“references to the created images.”** Samaniego, [0104], Fig. 6. These references **“are the script used to create the image, the names of the images used to create the image, the dates of those files, and the HTML that represents the created image,”** which is a “base transcoding profile.” Samaniego, [0104]. These profiles are “limited” because they are created for only the previously-cached versions of the requested media content.

(c) Element [9.b]

113. As set forth in the table below, Samaniego discloses this element:

'242	Samaniego
[9.b] at the time of the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation	• [0148] (“Upon receipt of the page 301, the browser contacts the system 100 requesting media specified in the URL. The system parses the URL 2100 to determine the content generation procedure 2140 to execute, any corresponding input parameters to be used by such procedure, any dynamic content processing 2150 to be performed by dynamic media procedures, any user profile information 2160 to be used to modify the resulting image, and any cache control HTTP headers 2190 the site instructs to accompany the resulting image.”) • [0155] (“Table D below is a list of example proprietary URL tags used for content generation within the system according to the preferred embodiment of the invention. Additional tags may be added to the system as necessary.”)

3

'242	Samaniego
system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system;	• Table D (“p-user profile string Specifies a user profile identifier used to modify the final image prior to returning the image to the browser, thus allowing clients to modify the image returned to the browser to account for individual browsing conditions.”) • <i>See also</i> [0065], [0108], [0115], [0138], [0145]-[0147], [0151]-[0152], [0154]-[0155], [0157], Figs. 21-22.

114. Samaniego discloses that a browser contacts the system “requesting media specified in the URL”—a request that does not involve a network administrator. Samaniego, [0148]. Samaniego further discloses transcoding steps including a “user profile procedure” that meet this element. Samaniego, [0155], Table D.

(d) Element [9.c]

115. As set forth in the table below, Samaniego discloses this element:

'242	Samaniego
[9.c] an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the	• [0065] (“The system takes as input the client connection, server traffic, content generation procedures, and proprietary tags placed within the URL to generate optimized media for the client. The need for the Web author to create different versions of a Web site is reduced because the image content of the site is automatically handled by the system. In addition, generated media is cached such that further requests for the same media require little overhead.”)

'242	Samaniego
same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system;	• [0103] (“The media caching subsystem 620 determines if an image has been created for the requested proprietary tag. If the image has already been created and the files that built that image have not been modified, the media caching subsystem 620 returns an HTML tag that refers to a previously-generated image. If the image has not been created, the media caching subsystem 620 hands the HTML tag to a media creation subsystem 630[.] The media creation subsystem 630 returns an image to the media caching subsystem 620. The media caching subsystem 620 adds the created image and the HTML tag to a media cache database 640.”) • [0104] (“The media cache database 640 contains references to the created images 645. In a preferred embodiment, the references are the script used to create the image, the names of the images used to create the image, the dates of those files, and the HTML that represents the created image. The media caching subsystem 620 performs lookups in this database to determine if the image has been created. If the image has not been created the media caching subsystem 620 calls upon the media creation subsystem 630 to create the image and then store the results in the media cache database 640.”) • [0135] (“The system takes as input the client connection, server traffic, content generation procedures, and proprietary tags placed within the URL to generate optimized media for the client. The need for the Web author to create different versions of a Web site is reduced because the image content of the site is automatically handled by the system. In addition, the generated media is cached so that further requests for the same media require little overhead.”)

'242	Samaniego
	• [0137] (“A preferred embodiment of the invention is described with reference to FIG. 17. FIG. 17 is a schematic diagram of an image system within a typical Web infrastructure according to the invention. The image system 100 is placed in parallel to an existing Web server 110. The image system 100 may be on-site or off-site to the Web infrastructure. Multiple client browsers 120a-120d communicate with both the Web server 110 and the image server 100 via the Internet 130.”) • <i>See also, e.g.,</i> [0064], [0090]-[0092], [0094], [0102], [0105], [0108], [0111], [0126], [0134]-[0135], [0138]-[0140], [0143]-[0144], [0147]-[0148]-[0149], [0151]-[0152], [0155], [0157], Figs. 7, 9, 10, 21-22, Tables C, D, F.

116. **Samaniego** discloses that when a request for media content is made, a “media caching subsystem” checks whether a previously-generated media content item has been cached that was generated using the transcoding parameters specified in the current request as inputs. Samaniego, [0103], [0065]. If not, **the request is passed to the “media creation subsystem” to generate the content** based on the inputs specified in the request. Samaniego, [0103]. The “media creation subsystem” then returns the created content item to the “media caching subsystem” for caching. Samaniego, [0103]. **Samaniego** further discloses that “[m]**ultiple client browsers” communicate with the system**. Samaniego, [0137]. When multiple requests for

the same content are received, **subsequent requests are served using previously-generated media content items to minimize “overhead.”** Samaniego, [0135].

117. For example, based on these teachings, a person having ordinary skill in the art would have understood and found it obvious that, when the claimed request (*see* [9.a](§IX.B.2(b))) is received, the system would determine whether the same version of the content item is already being created at the media creation subsystem in response to an earlier request, and would return that version in order to reduce “overhead” associated with generating and caching the media. Samaniego, [0135]. Indeed, a person having ordinary skill in the art would have understood Samaniego to not only render obvious multiple clients requesting content with the same transcoding parameters would be common, but to actually disclose it, particularly because the system is implemented using a “Web server” in the “Internet.” Samaniego, [0137]. As was well-understood to a person having ordinary skill in the art, there were numerous users in the Internet with similar devices, and it would be common for such devices to be issuing similar requests for the same content. Samaniego not only renders obvious that such devices would be served with content generated using the same transcoding parameters, but actually discloses it. *See id.*

(e) Elements [9.d]-[9.f]

118. As set forth in the table below, Samaniego discloses these elements:

'242	Samaniego
[9.d] an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile; [9.e] wherein the act of transcoding the requested media content in accordance with the identified transcoding parameters comprises: an act of selecting a pre-existing base transcoded version of the requested media content	• [0149] (“The parser generates a unique primary lookup key 2110 for the specified resulting media. If the key corresponds to an existing generated media 2180, such media is returned immediately to the browser 120 through a media cache 2120, and the transaction is complete. Otherwise, a media generation occurs. In the case of media generation requiring dynamic content processing, a unique secondary lookup key corresponding to intermediate media is generated 2130. If intermediate media 2170 corresponding to this key is found, such media is passed directly to the dynamic media content system 2150 having dynamic media procedures, wherein appropriate action is taken to generate the required derivative from the intermediate media data. A unique key is generated for the derivative 2130 and passed to the media caching system 2120. If the media caching system finds no such intermediate image, such intermediate image is generated according to instructions specified by the content generation procedure, cached by the media cache system 2120 as a secondary cached media 2170, and passed to the dynamic media system 2150. Again, appropriate action is taken to generate the required derivative from the intermediate image data.”) • [0152] (“FIG. 22 shows a flowchart of the content generation procedure according to a preferred embodiment of the invention. A URL containing proprietary tags (2200) is parsed (2210) to determine the content generation procedure to execute, any dynamic modifications to the media, user profile characteristics, and proxy-cache control. A unique final lookup key is generated for the media (2220) and the media cache is checked (2230). If the indicated media exists, control passes to proxy-cache

'242	Samaniego
comprising intermediate derivative media that has been transcoded in accordance with only a portion of the identified transcoding parameters; and [9.f] an act of creating a final version by incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters;	control (2290) and the media is delivered to the browser (2295). Otherwise, dynamic media system tags are separated from content generation control tags (2240) and a unique intermediate image lookup key is generated (2250). The cache is then checked for such intermediate media (2261). If such intermediate media is found, it is used directly for dynamic processing, if required. Otherwise, content is generated (2262) and cached (2263), and the result is evaluated for dynamic processing (2270). If dynamic processing is required, the media is operated upon by the dynamic content generator (2271), otherwise it is evaluated for valid content type (2272). If the content type is invalid, the media is automatically converted to a valid type (2273). The resulting image is then customized by the user profiling system (2280) for specified browser or client attributes. Finally, any cache-control directives specified are attached to the response (2290) and the media is delivered to the browser (2295).” • <i>See also</i> [0144], [0146]-[0147], [0149]-[0152], Figs. 21-22.

119. Samaniego’s disclosures corresponding to elements [9.d] to [9.f] describe, on a high level (1) identifying additional transcoding parameters to be used to transcode content, some of which will be applied on top of a base transcoding profile (*e.g.*, element [9.d]); (2) selecting that pre-existing base transcoding profile

(e.g., element [9.e]); and then (3) creating a final version by performing the steps that remain to be performed on top of the profile (e.g., element [9.f]). Because of the overlapping nature of Samaniego's disclosures, I have grouped these elements together. As set forth below, these disclosures meet elements [9.d] to [9.f].

120. **Samaniego** discloses that **“intermediate media”** items are generated using “content generation” procedures corresponding to a set of transcoding parameters, and then are cached as content **corresponding to a “base transcoding profile”** that can serve as the starting point for generating further content. *See* Samaniego [0149]; [9.a](§ IX.B.2(b)). After the system generates an “intermediate media” item, when a subsequent request, such as the claimed request (which I explained in my discussion of element [9.a]), is received that specifies a larger set of transcoding parameters, the system selects the previously-generated “intermediate media” item, using this **“base transcoding profile”** that has been transcoded according to only some of the transcoding parameters specified in the request (for example, an intermediate media item on which additional **“dynamic processing”** and processing based on **“user characteristics”** is still required in order to fulfill this request), and then **further transcodes that intermediate media item in accordance with the identified transcoding parameters** (for example, the additional, needed **“dynamic processing”** parameters and transcoding that are based

on the “**user characteristics**” which were identified by parsing the URL).
Samaniego, [0149], [0152]; *see also* [9.b].

(f) Element [9.g]

121. As set forth in the table below, Samaniego discloses this element:

'242	Samaniego
[9.g] an act of delivering the transcoded media content to both client presentation systems concurrently.	• [0151] (“The final media is passed to a user profile system 2160 wherein browser characteristics specified through proprietary tags within the URL are inspected, and appropriate modification to the media is performed, based on such characteristics. The resulting image is handed to the media cache system 2120 for caching and returned to the browser 120.”) • <i>See also, e.g.,</i> [0092], [0136]-[0137], [0151]-[0152], Figs. 1, 17-19, 21-22.

122. As I discussed above for element [9.c](§IX.B.2(d)), **Samaniego** further discloses that “[m]ultiple client browsers” communicate with the system. Samaniego, [0137]. When multiple requests for the same content are received from these multiple client browsers, the system determines that the same content is already being created in response to an earlier request, and uses this previously-generated media content to serve the second request as well to minimize “overhead.” Samaniego, [0135]. For similar reasons, when “[m]ultiple client browsers” communicate with the system and request content, a person having ordinary skill in the art further would have understood and found it obvious that the “media creation subsystem” generates, and then the “media caching subsystem” caches, this media

item once and then this same content item is “**returned to the browser**” on each of the two systems. This reuses the same generated media item for both requests concurrently to reduce the required overhead for the system. *See* [9.c](§IX.B.2(d)); Samaniego, [0151].

C. Ground 4: Samaniego in View of Tso Renders Obvious Claim 9

123. While Samaniego not only renders obvious, but actually discloses, Challenged Claim 9, including element [9.c]’s and [9.g]’s recitation of “concurrent[.]” user requests and delivery of content items, to the extent additional disclosure of these claim elements is desired, as I discussed above with respect to Ground 1 in Section IX.A.1, Tso teaches “concurrently” handling two client requests for the same content as is recited in these elements, by using multiple-thread processing. A person having ordinary skill in the art would have been motivated to apply these teachings to Samaniego.

124. Like Tso (as I discussed above in Section IX.A.1), Samaniego is also in the same field as the ’242 patent, networked media delivery, and indeed the ’242 patent claims priority to, and has the same specification as, Samaniego. Samaniego, [0002], [0064], [0090]; *see also* ’242 patent, Cover.

125. While Samaniego teaches that it is advantageous to handle “identical requests ...without regeneration of images,” Tso teaches how to avoid generating

two copies of an image for concurrent requests by using multiple-thread processing. Samaniego, [0054]; Tso, 6:45-50. A person having ordinary skill in the art would have been motivated to modify Samaniego with Tso's teachings of multiple-thread processing. As was well-understood with such processing, using multiple-thread processing allows a system to achieve the beneficial and predictable result of serving two concurrent requests for the same media content in parallel using different processing threads, which was advantageous for several reasons.

126. **First**, Tso teaches handling concurrent requests for the same media—*e.g.*, before the output from the first request is cached—by using multiple-thread processing to concurrently serve the requests in parallel using separate threads, while Samaniego teaches that it is advantageous to handle multiple requests without regenerating the same content. Tso, 6:51-63; Samaniego, [0110]. A person having ordinary skill in the art would have been motivated to modify Samaniego with Tso's teachings to reduce unnecessary processing that otherwise occurs if the same content is processed twice in the same way, as would happen without Tso's multiple-thread processing teachings when the output of the first request is not yet cached when the second request is received.

127. **Second**, Tso teaches improving the efficiency and speed of the system by using multiple-thread processing to eliminate the need to “wait[] for a hypertext

object” to be added to cache before reusing the object because the second request is handled using the content already in the process of being transcoded. Tso, 6:51-63. This improves speed in two ways: first, with respect to second requests, the request does not need to wait for a new transcoding operation begins for the media to be prepared in response to the request, and instead the second request is handled when the transcoding process in response to the first request is complete. Second, because the content is being transcoded only a single time for both requests, this frees up transcoding processing resources, allowing the system to handle other transcoding requests more quickly. *See id.* Samaniego recognizes these speed benefits, and discloses that it is advantageous to improve the “speed” to “add images, video, and sound to sites.” Samaniego, [0045]. A person having ordinary skill in the art would have been motivated to modify Samaniego with Tso’s teachings to improve the efficiency and speed of the system.

128. **Third**, Tso’s teaching of managing processing load complements Samaniego’s teaching of adjusting the quality of images based on client “connection speed” and “server traffic.” Samaniego, [0115], Table C. Tso discloses using the “condition of the device” to adjust content processing—which expands on Samaniego’s teaching by identifying similar parameters to “server traffic,” including “current processing load” at a transcoding server. Tso, 6:64-7:14. A person having

ordinary skill in the art would have found it obvious to further consider “current processing load” at a server when determining how to adjust the bandwidth of images to beneficially maximize the use of server processing resources.

129. In light of these teachings, a person having ordinary skill in the art would have had a reasonable expectation of success in modifying Samaniego with Tso’s teachings of multi-threaded processing. Indeed, Tso teaches that its disclosed multi-threaded access to cached objects uses a “standard” interface and does not “require any particular configuration” for the cache, while Samaniego teaches using a cache to avoid regenerating content for subsequent requests. Tso, 4:48-5:7, 5:2-7; Samaniego, [0135], [0138]. A person having ordinary skill in the art therefore would have found it straightforward and obvious to modify Samaniego with Tso’s teachings, yielding a system in which Samaniego’s teachings are used to concurrently serve content to multiple users using multi-threaded processing. A person having ordinary skill in the art likewise would have had a reasonable expectation of success because Tso merely discloses using factors to consider when adjusting the bandwidth of images that are similar to those already disclosed in Samaniego.

X. SECONDARY CONSIDERATIONS

130. I have not seen any evidence in the prosecution history of the '242 patent or any related application that I considered of any arguments regarding secondary considerations exist. Nor have I seen any evidence that any such evidence would have a nexus to any of the Challenged Claim 9. Moreover, as shown above with respect to the prior art referenced in Section IX, any alleged solutions to problems or unexpected results in the '242 patent were already well known.

XI. CONCLUSION

131. In summary, I have concluded that the Challenged Claim 9 is rendered obvious over the prior art references as discussed above.

132. Specifically, in my opinion, for at least the reasons above, the Challenged Claim 9 rendered obvious over Tso in view of Huang, anticipated and rendered obvious by Samaniego, and rendered obvious by Samaniego in view of Tso.

133. To the extent it is argued that any further disclosure is required for an element in the Challenged Claim 9 that I have identified as being disclosed, a person having ordinary skill in the art would certainly have found that element obvious to include based on the same explicit disclosure or inherent disclosures and analysis I have identified above.

134. I reserve the right to supplement my opinions in the future to respond to any arguments that Patent Owner or its expert(s) may raise and to take into account new information as it becomes available to me.

135. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

136. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this 10th day of January 2023, in Atlanta, Georgia

Respectfully submitted,



Prof. Vijay K. Madisetti, Ph.D.

Appendix A

Dr. Vijay K. Madiseti

Fellow, IEEE

vkm@madiseti.com

Cell: 770-527-0177

Address:

**56 Creekside Park Drive
Johns Creek, GA 30022**

Employment:

- 1989-present: Full Professor of Cybersecurity & Privacy, and Electrical & Computer Engineering (**Georgia Tech, Atlanta, GA 30332**).

Areas of Technical Interest – Software Engineering, Cloud Computing Signal Processing, Embedded Systems, Wireless & Mobile Communications, Computer Engineering, Circuit Design (Analog/Digital), Biomedical Instrumentation Hardware/Software, Software Engineering, Digital Signal Processing, Sensors, Wireline & Wireless Computer Networking, Internet of Things (IoT).

Education: PhD (EECS, University of California at Berkeley, 1989), **B.Tech** (Hons) in Electronics and Electrical Communications Engineering (Indian Institute of Technology, Kharagpur, 1984).

Startup Companies:

Director, **VP Technologies, Inc.** (1995-): A startup commercialized through Georgia Tech's Advanced Technology Development Corporation (ATDC) focusing on digital software and hardware design services for military market. <http://www.vptinc.net>

Director, **Soft Networks, LLC** (2001-2007): A startup commercialized through Georgia Tech support focusing on software development tools and compilers for Cellular/WiFi/VOIP/telecommunication products. <http://www.soft-networks.com>

Director, **Elastic Video Inc.** (2007- 2009): A startup commercialized through Georgia Tech's VentureLab (<http://venturelab.gatech.edu>) development image and video processing software for wireless & IP networking.

Litigation Experience (2016-2022) With Testimony

(Note: There may be multiple cases between the parties, e.g., District Court v. ITC, US versus Foreign Cases)

Chrimar v. Adtran, Alcatel, et al.

Technology: Power over Ethernet (2015-2017)

Submitted reports & Deposition & Trial

Chamberlain v. Ryobi/TTI

Case No: 1:16-cv-06097 (ND Illinois)

Expert for Ryobi

Technology: Wireless/IoT/Barrier Movement (2016 – 2019)

Submitted reports & deposition & trial testimony

IOEngine v. IMC/Imation

Case No: cv-14-1572-GMS (US Delaware)

Expert for IMC/Imation

Technology: Networked Storage Device (2016-2017)

Submitted reports & deposition & trial testimony

Huawei v. Samsung

Case No: 3:16-cv-2787-WHO (ND Cal)

Expert for Samsung

Technology: 4G/LTE Random Access Protocols (2016-2019)

Submitted reports & deposition

Hitachi Maxell v. ZTE/Huawei

Case No: 5:16-cv-00178-RWS (ED Texas)

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted reports & deposition (2017 – 2018)

Hitachi Maxell v. Apple

Case No: 5:19-cv-00036-RWS

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted Expert Reports (2020-)

Qualcomm v. Apple

Case No: 17-ccv-0108-GPC-MDD (SD Cal) Also, Related ITC/FTC Matters

Expert for Qualcomm

Technology: 4G/Wireless Communications/Smartphones (2017-2019)

Submitted Reports and Deposition

Qualcomm v. Apple

Case No: 3:17-cv-01375-DMS-MDD (SD Cal)

Expert for Qualcomm

Technology: 4G/Wireless Communications/Smartphones (2017-2019)

Submitted Reports and Deposition

Optis v. Huawei

Case No: 2:17-cv-123 (E.D. Texas)

Expert for Optis Wireless

Technology: 4G/Video (2017-2019)

Submitted reports & deposition

Beckman Coulter v. Sysmex

Case No: 1:17-cv-24049-DPG (ND Illinois)

Expert for Sysmex

Technology: Medical Instrumentation Automation (2017-present)

Testifying Expert

TQ Delta v. Xyxel/Adtran

Expert for TQ Delta

Technology: DSL Technologies (2018-present)

Testifying Expert

3GL/KPN v. LG, Blackberry, HTC

Expert for 3GL/KPN

Technology: 4G/LTE Protocols (2018-present)

Reports and Deposition / Testifying Expert

Rovi v. Comcast

ITC No. 337-TA-1103 (ITC)

Expert for Rovi

Technology: Digital Video & interactive GUI (Jan 2018-May 2018)

Reports and Deposition

Cirba/Densify v. VMWare

Case No: 1:19-cv-00742-LPS

Expert for Cirba/Densify

Technology: Virtualization (2019-present)

Reports Deposition/PI Hearing / Testifying Expert

Power Integrations v. On Semiconductor

Case No: 17-cv-03189-BLF

Expert for On Semiconductor

Technology: Power Electronics (2018-2019)

Reports & Deposition/Testifying Expert.

Pathway v. Lumens, IPEVO, Aver JDA

Case No: ITC 337-TA-1045

Expert for Lumens JDA

Technology: Document Cameras (2016-2019)

Reports & Deposition (for IPR/ITC, no trial)

Wilan v. Apple

Case No: 3:14-cv-2235

Expert for WiLan

Technology: Voice over LTE/LTE Protocols (2017-2020)

Reports and Testified through Deposition/Trials

St. Lawrence Comm. v. Amazon

Case No. 2:19-CV-00027-JRG

Expert for Amazon

Technology: Multimedia Codecs

Submitted Expert Reports/Deposition (2020-2020)

PanOptis v. Apple

Case No. 2:19-CV-00066-JRG

Expert for PanOptis

Technology: LTE

Submitted Expert Reports/Deposition/Trial (2019-2020)

GAS v. Sprint Communications

Case No. 2:20-cv-00007-RWS

Expert for General Access (GAS)

Technology: LTE

Submitted Expert Reports/Deposition (9/21-)

Causam v. Ecobee, Itron, et al

Case No. ITC 337-TA-1277

Expert for Causum

Technology: Load Management

Submitted Expert Reports and Deposition (6/22)

Masimo v. Apple

Case No. ITC 337-TA-1276 and related cases

Expert for Masimo

Technology: Pulse Oximetry

Submitted Expert Reports, Deposition & Trial

TQ Delta v. Nokia, Adtran, CommScope, 2Wire, et al

Cases in Delaware & Texas

Expert for TQ Delta

Technology: DSL and Cable Modems

Submitted Expert Reports, Depositions & Hearings.

Paltalk Holdings v. Cisco Systems Inc.

Western District of Texas (6:21-CV-00757-ADA)

Expert for Paltalk Holdings

Technology: VOIP Technologies

Submitted Expert Reports and Deposition

Ollnova Technologies v. Carrier Group

Southern District of Florida (Case 9:22-cv-80388-DMM)

Expert for Ollnova

Technology: IoT

Submitted Report and Deposition.

Additional matters include declarations supporting IPRs at the PTAB for Google (US Patent 8,601,154), On Semiconductor (US Patent 6,212,079), Ubisoft (US Patent 5,490,216), Lumens JDA (2017-2019), Broadcom (WiFi), Sony (US Patent 6,101,534), Kia, (US Patent 5,530,431), Qualcomm, Ericsson, Daimler Benz, VW, Estech, Amazon, Ring, Digital Ally, On Semiconductor, LoganTree, Unified Patents, BMC Software, Masimo, Sentius, BMW, Facebook, PAWS, Google., Daimler and Dolby.

Earned Degrees

1. B. Tech (Hons), Electronics & Electrical Comm. Engineering

Indian Institute of Technology (IIT), Kharagpur, India

1984.

2. Ph.D., Electrical Engineering & Computer Sciences (EECS)

University of California (UC), Berkeley. CA

1989.

Books

1. VLSI Digital Signal Processors

Madiseti, V.K.;

Boston: MA, IEEE Press: Butterworth Heinemann, 1995, 525 pp.

2. Quick-Turnaround ASIC Design in VHDL

Romdhane, M., Madiseti, V.K., Hines, J.

Boston: MA, Kluwer Academic Press, 1996, 190 pp.

3. The Digital Signal Processing Handbook (First Edition)

Madiseti, V. K., Williams, D. (Editors)

CRC Press, Boca Raton, Fla, 1998, 2500 pp.

4. VHDL: Electronics Systems Design Methodologies.

Madiseti, V. K. (Editor)

Boston: MA, IEEE Standards Press, 2000, ISBN 0-7381-1878-8.

5. Platform-Centric Approach to System-on-Chip (SoC) Design.

Madiseti, V. K., Arpikanondt, A.

Springer, Boston: MA, Springer, 2004, 280 pp.

6. The Digital Signal Processing Handbook – Second Edition.

Madiseti, V. K. (2009)

CRC Press, Boca Raton, Fla.

7. Cloud Computing: A Hands-On Approach

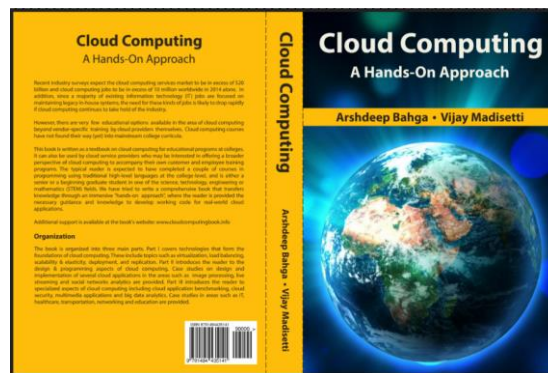
A Bahga, V. Madiseti (2013)

Amazon CreateSpace Publishing, 2013, 454 pp.

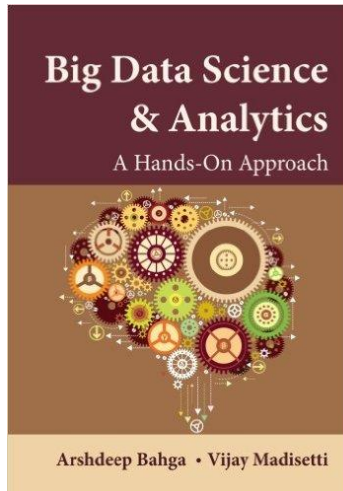
8. Internet of Things: A Hands-On Approach

A Bahga, V. Madiseti (2014)

Amazon CreateSpace Publishing, 2014, 450 pp.



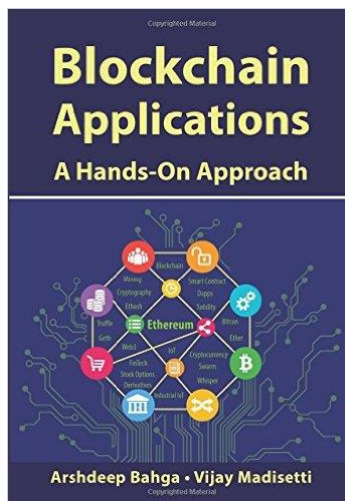
9. Big Data Science & Analytics: A Hands-On Approach
A Bahga, V. Madiseti (2016)
 Amazon CreateSpace Publishing, 2016, 542 pp.



10. Blockchain Applications: A Hands-On Approach

A Bahga, V. Madiseti (2017)

Amazon CreateSpace Publishing, 2017, 380 pp.



Edited Books & Collection of Papers

1. Advances in Parallel & Distributed Simulation

Madiseti, V.K.; Nicol, D., Fujimoto, R. (Editors)

San Diego, CA: SCS Press, 1991, 200 pp.

2. Modeling, Analysis, Simulation of Computer & Telecommunications Systems

Madiseti, V., Gelenbe, E., Walrand, J. W. (Editors)

Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. Modeling & Simulations on Microcomputers

Madiseti, V.K. (Editor)

San Diego, CA: SCS Press, 138 pp. 1990.

Editorship of Journals & Transactions

1. IEEE Design & Test of Computers

Special Issue: Reengineering Digital Systems

April – June 1999 (Vol 16, No 2)

Madiseti, V.K (Editor)

Los Alamitos: CA, IEEE Computer Society Press, 1999.

2. IEEE Design & Test of Computers

Special Issue: Rapid Prototyping of Digital Systems

Fall 1996 (Vol 13, No 3)

Madiseti, V., Richards, M. (Editors)

Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. IEEE Transactions on Circuits & Systems II

Associate Editor: 1993-1995.

4. International Journal in Computer Simulation

Associate Editor: 1990-1993

5. International Journal in VLSI Signal Processing

Editorial Board: 1995 - Present

Recent Issued US Patents

- [1] [10,503,927](#) [Method and system for securing cloud storage and databases from insider threats and optimizing performance](#), Issued Dec 10, 2019
- [2]. [10,460,283](#) [Smart contract optimization for multiparty service or product ordering system](#), Issued Oct 29, 2019
- [3]. [10,459,946](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued Oct 29, 2019

- [4]. [10,402,589](#) [Method and system for securing cloud storage and databases from insider threats and optimizing performance](#), Issued Sep 3, 2019
- [5]. [10,394,845](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), Issued Aug 27, 2019
- [6]. [10,289,631](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), Issued May 24, 2019
- [7]. [10,255,342](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued April 9, 2019
- [8]. [10,243,743](#) [Tokens or crypto currency using smart contracts and blockchains](#), Issued March 26, 2019
- [9]. [10,204,339](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), Issued February 12, 2019
- [10]. [10,204,148](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued Feb 12, 2019
- [11]. [10,121,143](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), Issued Nov 6, 2018
- [12]. [10,102,526](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), October 16, 2018
- [13]. [10,102,265](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), October 16, 2018
- [14]. [9,935,772](#) [Methods and systems for operating secure digital management aware applications](#), April 3, 2018
- [15]. [9,769,213](#) [Method and system for secure digital object management](#), Issued Sep 19, 2017

Refereed Journal Publications

1. Trends in the Electronic Control of Mine Hoists

*Madiseti, V. and Ramlu, M.,
IEEE Transactions on Industry Applications, Vol IA-22, No. 6,
November/December 1986. Pages 1105-1112*

2. Multilevel range/NEXT performance in digital subscriber loops

Brand, G.; Madiseti, V.; Messerschmitt, D.G.;
Communications, Speech and Vision, IEE Proceedings I [see also IEE

Proceedings-Communications] ,Volume: 136 , Issue: 2 , April 1989
Pages:169 – 174

3. Seismic migration algorithms on parallel computers

Madiseti, V.K.; Messerschmitt, D.G.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 39 , Issue: 7 , July 1991
Pages:1642 – 1654

4. Asynchronous algorithms for the parallel simulation of event-driven dynamical systems

Madiseti, V.K.; Walrand, J.C.; Messerschmitt, D.G.:
ACM Transactions on Modeling and Computer Simulation, v 1, n 3, July 1991,
Pages: 244-74

5. Synchronization mechanisms for distributed event-driven computation

Madiseti, V.K.; Hardaker, D.:
ACM Transactions on Modeling and Computer Simulation, v 2, n 1, Jan. 1992,
Pages: 12-51

6. Efficient VLSI Architectures for the Arithmetic Fourier Transform (AFT)

Kelley, B.T.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 1 , January
1993
Pages:365-378

7. The fast discrete Radon transform. I. Theory

Kelley, B.T.; Madiseti, V.K.;
Image Processing, IEEE Transactions on ,Volume: 2 , Issue: 3 , July 1993
Pages:382 – 400

8. The Georgia Tech digital signal multiprocessor

Barnwell, T.P., III; Madiseti, V.K.; McGrath, S.J.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 7 , July 1993
Pages:2471 – 2487

9. The MIMDIX Environment for Parallel Simulation

Madiseti, V.K.; Hardaker, D.; Fujimoto, R.M.:
Journal of Parallel and Distributed Computing, v18, no. 4, August 1993,
Pages: 473-83.

- 10. LMSGEN: a prototyping environment for programmable adaptive digital filters in VLSI**
Romdhane, M.S.B.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:33 – 42
- 11. Fixed-point co-design in DSP**
Egolf, T.W.; Famorzadeh, S.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:113 - 126
- 12. A fast spotlight-mode synthetic aperture radar imaging system**
Madiseti, V.K.;
Communications, IEEE Transactions on ,Volume: 42 , Issue: 234 , February-April 1994
Pages:873 – 876
- 13. Rapid prototyping on the Georgia Tech digital signal multiprocessor**
Curtis, B.A.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 3 , March 1994
Pages:649 – 662
- 14. Low-power signaling in asymmetric noisy channels via spectral shaping**
Sipitca, M.; Madiseti, V.K.;
Signal Processing Letters, IEEE, Volume: 1 , Issue: 8 , Aug 1994
Pages:117 – 118
- 15. A quantitative methodology for rapid prototyping and high-level synthesis of signal processing algorithms**
Madiseti, V.K.; Curtis, B.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 11 , Nov. 1994
Pages:3188 – 3208
- 16. Computer Simulation of Application-Specific Signal Processing Systems**
Casinovi, G.; Madiseti, V.K.;
International Journal in Computer Simulation, Vol. 4, No. 4, Nov 1994.
- 17. System partitioning of MCMs for low power**
Khan, S.A.; Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 12 , Issue: 1 , Spring 1995
Pages:41 – 52

18. Error correcting run-length limited codes for magnetic recording

Jaejin Lee; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 6 , Nov. 1995

Pages:3084 – 3086

19. Virtual prototyping of embedded microcontroller-based DSP systems

Madiseti, V.K.; Egolf, T.W.;

Micro, IEEE ,Volume: 15 , Issue: 5 , Oct. 1995

Pages:9 – 21

20. Constrained multitrack RLL codes for the storage channel

Lee, J.; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 3 , May 1995

Pages:2355 – 2364

21. Rapid digital system prototyping: current practice, future challenges

Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:12 – 22

22. Conceptual prototyping of scalable embedded DSP systems

Dung, L.-R.; Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:54 – 65

23. Advances in rapid prototyping of digital systems

Madiseti, V.K.; Richards, M.A.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:9

24. Combined modulation and error correction codes for storage channels

Jaejin Lee; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 32 , Issue: 2 , March 1996

Pages:509 – 514

25. Model-based architectural design and verification of scalable embedded DSP systems-a RASSP approach

Dung, L.-R.; Madiseti, V.K.; Hines, J.W.;

Chapter in VLSI Signal Processing, IX, 1996.

Pages:147 – 156

26. Low-power digital filter implementations using ternary coefficients

Hezar, R.; Madiseti, V.K.;

Chapter in VLSI Signal Processing, IX, 1996.,

Pages:179 – 188

27. All-digital oversampled front-end sensors

Romdhane, M.S.B.; Madiseti, V.K.;

Signal Processing Letters, IEEE, Volume: 3 , Issue: 2 , Feb. 1996

Pages:38 – 39

28. Modeling COTS components in VHDL

Calhoun, S., Reese, R; Egolf, T., Madiseti, V.K.;

Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996

Pages: 24 – 31

29. VHDL-Based Rapid Systems Prototyping

Egolf, T.; Madiseti, V.K.;

Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996

Pages: 40-52

30. Interface design for core-based systems

Madiseti, V.K.; Lan Shen;

Design & Test of Computers, IEEE ,Volume: 14 , Issue: 4 , Oct.-Dec. 1997

Pages:42 - 51

31. Incorporating cost modeling in embedded-system design

Debardelaben, J.A.; Madiseti, V.K.; Gadiant, A.J.;

Design & Test of Computers, IEEE ,Volume: 14 , Issue: 3 , July-Sept. 1997

Pages:24 – 35

32. On homomorphic deconvolution of bandpass signals

Marenco, A.L.; Madiseti, V.K.;

Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and

Signal Processing, IEEE Transactions on] ,Volume: 45 , Issue: 10 , Oct. 1997

Pages:2499 – 2514

33. A case study in the development of multi-media educational material: the VHDL interactive tutorial

Gadiant, A.J.; Stinson, J.A., Jr.; Taylor, T.C.; Aylor, J.H.; Klenke, R.H.;

Salinas, M.H.; Madiseti, V.K.; Egolf, T.; Famorzadeh, S.; Karns, L.N.; Carter,

H.W.;

Education, IEEE Transactions on ,Volume: 40 , Issue: 4 , Nov. 1997

Pages:17 pp.

34. Adaptive mobility management in wireless networks

Jeongwook Kim; Madiseti, V.K.;

Electronics Letters ,Volume: 34 , Issue: 15 , 23 July 1998

Pages:1453 – 1455

35. Efficient implementation of two-band PR-QMF filterbanks

Hezar, R.; Madiseti, V.K.;

Signal Processing Letters, IEEE ,Volume: 5 , Issue: 4 , April 1998

Pages:92 – 94

36. On fast algorithms for computing the inverse modified discrete cosine transform

Yun-Hui Fan; Madiseti, V.K.; Mersereau, R.M.;

Signal Processing Letters, IEEE ,Volume: 6 , Issue: 3 , March 1999

Pages:61 – 64

37. System on chip or system on package?

Tummala, R.R.; Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:48 – 56

38. Reengineering legacy embedded systems

Madiseti, V.K.; Jung, Y.-K.; Khan, M.H.; Kim, J.; Finnessy, T.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:38 – 47

39. Reengineering digital systems

Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:15 – 16

40. Parameter optimization of robust low-bit-rate video coders

Sangyoun Lee; Madiseti, V.K.;

Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 9
Issue: 6 , Sept. 1999

Pages:849 – 855

41. Closed-form for infinite sum in bandlimited CDMA

Jatunov, L.A.; Madiseti, V.K.;

Communications Letters, IEEE ,Volume: 8 , Issue: 3 , March 2004

Pages:138 – 140

42. A new protocol to enhance path reliability and load balancing in mobile ad hoc networks

Argyriou, A.; Madiseti, V.K.;

Journal of Ad Hoc Networks, Elsevier Press, 2004

43. Closed-form analysis of CDMA systems using Nyquist pulse

Jatunov, L.A.; Madiseti, V. K.;

Communications Letters, IEEE (Under Revision), 2005.

44. Systematic Design of End-to-End Wireless Mobility Management Protocols,

Argyriou, A.; Madiseti, V. K.;

ACM/Springer Wireless Networks (WINET), Accepted 2005.

45. A Novel End-to-End Approach for Video Streaming Over the Internet,

Argyriou, A.; Madiseti, V. K.;

Kluwer Telecommunications Systems, Vol. 28, No. 2, Pages 133-150, Jan 2005. *Special Issue on Multimedia Streaming.*

46. An Analytical Framework of RD Optimized Video Streaming with TCP,

Argyriou, A.; Madiseti, V. K.;

IEEE Transactions on Multimedia, Submitted for review in March 2005.

47. Modeling the Effect of Handoffs on Transport Protocol Performance,

Argyriou, A.; Madiseti, V. K.;

IEEE Transactions on Mobile Computing, Submitted for review in March 2005

48. Throughput Models for Transport Protocols with CBR and VBR Traffic Workloads",

Argyriou, A.; Madiseti, V. K.;

ACM Transactions on Multimedia Computing, Communications & Applications, Submitted for review in April 2005.

49. "Electronic System, Platform & Package Codesign",

Madiseti, V. K.

IEEE Design & Test of Computers, Volume 23, Issue 3, June 2006. pages 220-233.

50. "The Design of an End-to-End Handoff Management Protocol",

A. Argyriou, Madiseti, V. K.

Wireless Networks, Springer, May 2006.

51. "A Soft-Handoff Transport Protocol for Media Flows in Heterogeneous Mobile Networks ",

A. Argyriou, Madiseti, V. K.

Computer Networks, Vol 50, Issue 11, Pages 1860-1871, August 2006.

- 52. "Computationally Efficient SNR Estimation for Bandlimited WCDMA Systems"**
L. Jatunov, Madiseti, V. K.
IEEE Transactions on Wireless Communications, Volume 5, Issue 13,
December 2006, Pages 3480-3491.
- 53. "Space-Time Codes for Wireless & Mobile Applications",**
M. Sinnokrot, Madiseti, V.K.
DSP Handbook, Second Edition, 2009 (to be published)
- 54.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems",** *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013
- 55.A. Bahga, V. Madiseti, "Cloud-Based Information Integration & Informatics Framework for Healthcare Applications",** *IEEE Computer*, February 2015.
- 56.A. Bahga, V. Madiseti, "A Cloud-based Approach for Interoperable EHRs",** *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 57.A. Bahga, V. Madiseti, "Cloud-Based Information Technology Framework for Data Driven Intelligent Transportation Systems",** *Journal of Transportation Technologies*, vol.3 no.2, April 2013
- 58.A. Bahga, V. Madiseti, "Performance Evaluation Approach for Multi-tier Cloud Applications",** *Journal of Software Engineering and Applications*, vol. 6, no. 2, pp. 74-83, Mar 2013.
- 59.Yusuf, A., V. Madiseti, "Configuration for Predicting Travel Time Using Wavelet Packets and Support Vector Regression",** *Journal of Transportation Technologies*, vol 3, no. 3, June 2013.
- 60.A. Bahga, V. Madiseti, "Analyzing Massive Machine Maintenance Data in a Computing Cloud",** *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1831 - 1843, 2012.
- 61.N. Radia. Y. Zhang, M. Tatimapula, V. Madiseti, "Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions,"** *Proceedings of the IEEE*, Vol 100, Issue 4, pp. 841-854, 2012.
- 62.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems",** *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013

- 63.**A. Bahga, V. Madiseti, "**A Cloud-based Approach for Interoperable EHRs**", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 64.**A. Bahga, V. Madiseti, "**Cloud-Based Information Integration & Informatics Framework for Healthcare Applications**", *IEEE Computer*, February 2015.

Peer Reviewed Conference Publications

- 1. Dynamically-reduced complexity implementation of echo cancelers**
Madiseti, V.; Messerschmitt, D.; Nordstrom, N.;
Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. ,Volume: 11 , Apr 1986
- 2. Seismic migration algorithms using the FFT approach on the NCUBE multiprocessor**
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988
International Conference on , 11-14 April 1988
- 3. Seismic migration algorithms on multiprocessors**
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988
International Conference on , 11-14 April 1988
Pages:2124 - 2127 vol.4
- 4. WOLF: A rollback algorithm for optimistic distributed simulation systems**
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Conference Proceedings, 1988 Winter , December 12-14, 1988
Pages:296 – 305
- 5. Efficient distributed simulation**
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Symposium, 1989. The 22nd Annual , March 28-31, 1989
Pages:5 - 6
- 6. High speed migration of multidimensional seismic data**
Kelley, B.; Madiseti, V.;

Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991
International Conference on , 14-17 April 1991
Pages:1117 - 1120 vol.2

7. Performance of a fast analog VLSI implementation of the DFT

Buchanan, B.; Madiseti, V.; Brooke, M.;
Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on
, 9-12 Aug. 1992
Pages:1353 - 1356 vol.2

8. Task scheduling in the Georgia Tech digital signal multiprocessor

Curtis, B.A.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on ,Volume: 5 , 23-26 March 1992
Pages:589 - 592 vol.5

9. The fast discrete Radon transform

Kelley, B.T.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on ,Volume: 3 , 23-26 March 1992
Pages:409 - 412 vol.3

10.Yield-based system partitioning strategies for MCM and ASEM design

Khan, S.; Madiseti, V.;
Multi-Chip Module Conference, 1994. MCMC-94, Proceedings., 1994 IEEE,15-
17 March 1994
Pages:144 - 149

11. Multitrack RLL codes for the storage channel with immunity to intertrack interference

Lee, J.; Madiseti, V.K.;
Global Telecommunications Conference, 1994. GLOBECOM '94.
'Communications: The Global Bridge', IEEE,Volume: 3 , 28 Nov.-2 Dec. 1994
Pages:1477 - 1481 vol.3

12. A parallel mapping of backpropagation algorithm for mesh signal processor

Khan, S.A.; Madiseti, V.K.;
Neural Networks for Signal Processing [1995] V. Proceedings of the 1995
IEEE Workshop , 31 Aug.-2 Sept. 1995
Pages:561 - 570

13. Virtual prototyping of embedded DSP systems

Madiseti, V.K.; Egolf, T.; Famorzadeh, S.; Dung, L.-R.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995

International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2711 - 2714 vol.4

14. Assessing and improving current practice in the design of application-specific signal processors

Shaw, G.A.; Anderson, J.C.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2707 - 2710 vol.4

15. Introduction to ARPA's RASSP initiative and education/facilitation program

Corley, J.H.; Madiseti, V.K.; Richards, M.A.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2695 - 2698 vol.4

16. DSP design education at Georgia Tech

Madiseti, V.K.; McClellan, J.H.; Barnwell, T.P., III;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 5 , 9-12 May 1995
Pages:2869 - 2872 vol.5

17. Rapid prototyping of DSP systems via system interface module generation

Famorzadeh, S.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1256 - 1259 vol. 2

18. Rapid prototyping of DSP chip-sets via functional reuse

Romdhane, M.S.B.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1236 - 1239 vol. 2

19. A constructive deconvolution procedure of bandpass signals by homomorphic analysis

Marenco, A.L.; Madiseti, V.K.;
Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International ,Volume: 3 , 27-31 May 1996
Pages:1592 - 1596 vol.3

20. BEEHIVE: an adaptive, distributed, embedded signal processing environment

Famorzadeh, S.; Madiseti, V.; Egolf, T.; Nguyen, T.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 1 , 21-24 April 1997
Pages:663 - 666 vol.1

21. Target detection from coregistered visual-thermal-range images

Perez-Jacome, J.E.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 4 , 21-24 April 1997
Pages:2741 - 2744 vol.4

22. Variable block size adaptive lapped transform-based image coding

Klausutis, T.J.; Madiseti, V.K.;
Image Processing, 1997. Proceedings., International Conference on ,Volume: 3 , 26-29 Oct. 1997
Pages:686 - 689 vol.3

23. A Rate 8/10 (0, 6) MTR Code And Its Encoder/decoder

Jaejin Lee; Madiseti, V.K.;
Magnetics Conference, 1997. Digests of INTERMAG '97., 1997 IEEE International , 1-4 April 1997
Pages:BS-15 - BS-15

24. VHDL models supporting a system-level design process: a RASSP approach

DeBardelaben, J.A.; Madiseti, V.K.; Gadiant, A.J.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:183 - 188

25. A performance modeling framework applied to real time infrared search and track processing

Pauer, E.K.; Pettigrew, M.N.; Myers, C.S.; Madiseti, V.K.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:33 - 42

26. System design and re-engineering through virtual prototyping: a temporal model-based approach

Khan, M.H.; Madiseti, V.K.;
Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on ,Volume: 2 , 1-4 Nov. 1998
Pages:1720 - 1724 vol.2

27. A debugger RTOS for embedded systems

Akgul, T.; Kuacharoen, P.; Mooney, V.J.; Madiseti, V.K.;
Euromicro Conference, 2001. Proceedings. 27th , 4-6 Sept. 2001
Pages:264 - 269

28. Adaptability, extensibility and flexibility in real-time operating systems

Kuacharoen, P.; Akgul, T.; Mooney, V.J.; Madiseti, V.K.;
Digital Systems, Design, 2001. Proceedings. Euromicro Symposium on , 4-6 Sept. 2001
Pages:400 - 405

29. Effect of handoff delay on the system performance of TDMA cellular systems

Turkboylari, M.; Madiseti, V.K.;
Mobile and Wireless Communications Network, 2002. 4th International Workshop on , 9-11 Sept. 2002
Pages:411 - 415

30. Enforcing interdependencies and executing transactions atomically over autonomous mobile data stores using SyD link technology

Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on , 19-22 May 2003
Pages:803 - 809

31. Performance evaluation and optimization of SCTP in wireless ad-hoc networks

Argyriou, A.; Madiseti, V.;
Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , 20-24 Oct. 2003
Pages:317 - 318

32. Implementation of a calendar application based on SyD coordination links

Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Parallel and Distributed Processing Symposium, 2003. Proceedings. International , 22-26 April 2003
Pages:8 pp.

33. Bandwidth aggregation with SCTP

Argyriou, A.; Madiseti, V.;
Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE Volume: 7 , 1-5 Dec. 2003
Pages:3716 - 3721 vol.7

34. Software streaming via block streaming

Kuacharoen, P.; Mooney, V.J.; Madisetti, V.K.;
Design, Automation and Test in Europe Conference and Exhibition, 2003
, 2003
Pages:912 – 917

35. Frequency-dependent space-interleaving for MIMO OFDM systems

Mohajerani, P.; Madisetti, V.K.;
Radio and Wireless Conference, 2003. RAWCON '03. Proceedings , Aug. 10-13, 2003
Pages:79 - 82

36. A media streaming protocol for heterogeneous wireless networks

Argyriou, A.; Madisetti, V.;
Computer Communications, 2003. CCW 2003. Proceedings. 2003 IEEE 18th Annual Workshop on , 20-21 Oct. 2003
Pages:30 – 33

37. Realizing load-balancing in ad-hoc networks with a transport layer protocol

Argyriou, A.; Madisetti, V.;
Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE ,Volume: 3 , 21-25 March 2004
Pages:1897 - 1902 Vol.3

38.Streaming H.264/AVC video over the Internet

Argyriou, A.; Madisetti, V.;
Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE , 5-8 Jan. 2004
Pages:169 – 174

Other Publications

1. **A Transport Layer Technology for Improving the QoS of Networked Multimedia Applications <draft-madisetti-arguriou-qos-sctp-00.txt>.**
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Jul 25, 2002.
2. **Voice & Video over Mobile IP Networks <draft-madisetti-argyriou-voice-video-mip-00.txt>**
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Nov 20, 2002.
3. **Enhancements to ECRTTP with Applications to Robust Header Compression for Wireless Applications. <draft-madisetti-rao-suresh-rohc-00.txt>**
Madisetti, V.; Rao, S., Suresh, N.
IETF Internet-Draft, June 30, 2003.

Ph.D. Students Graduated

1. **Brian T. Kelley, 1992**
VLSI Computing Architectures for High Speed Signal Processing
Member of Technical Staff, Motorola.

Winner of Dr. Thurgood Marshall Dissertation Fellowship Award
2. **Bryce A. Curtis, 1992**
Special Instruction Set Multiple Chip Computer for DSP
Member of Technical Staff, IBM
3. **Jaejin Lee, 1994**
Robust Multitrack Codes for the Magnetic Channel
Professor, Yonsei University, Korea
4. **Mohamed S. Ben Romdhane, 1995**
Design Synthesis of Application-Specific IC for DSP
Director of IP, Rockwell.
5. **Shoab A. Khan, 1995**
Logic and Algorithm Partitioning on MCMs
Professor, National University of Science & Technology, Pakistan
6. **Lan-Rong Dung, 1997**
VHDL-based Conceptual Prototyping of Embedded DSP Architectures
Professor, National Chaio Tung University, Taiwan.

Winner of VHDL International Best PhD Thesis Award, 1997

7. Thomas W. Egolf, 1997

Virtual Prototyping of Embedded DSP Systems

Distinguished Member of Technical Staff, Agere

8. Alvaro Marenco, 1997

On Homomorphic Deconvolution of Bandpass Signals

Professor, Texas A&M University.

Winner of GIT ECE Outstanding Teaching Assistant Award

9. Shahram Famorzadeh, 1997

BEEHIVE: A Distributed Environment for Adaptive Signal Processing

Member of Technical Staff, Rockwell.

10. Timothy J. Klausutis, 1997

Adaptive Lapped Transforms with Applications to Image Coding.

US Air Force/Univ. of Florida.

11. Lan Shen, 1998

Temporal Design of Core-Based Systems

Member of Technical Staff, IBM

12. James DeBardelaben, 1998

Optimization Based Approach to Cost Effective DSP Design

Research Scientist, Johns Hopkins University

Georgia Tech ECE Faculty Award

13. Sangyoun Lee, 1999

Design of Robust Video Signal Processors

Professor, Yonsei University

US Army Sensors Lab Research Excellence Award, 1999

14. Rahmi Hezar, 2000

Oversampled Digital Filters

Member of Technical Staff, Texas Instruments

15. Yong-kyu Jung, 2001

Model-Based Processor Synthesis

Professor, Texas A&M University

16. Mustafa Turkboylari, 2002

Handoff Algorithms for Wireless Applications

Member of Technical Staff, Texas Instruments

17. Yun-Hui Fan, 2002

A Stereo Audio Coder with Nearly Constant Signal to Noise Ratio

Post-Doctoral Research Associate, Northeastern University

18. Subrato K. De, 2002

Design of a Retargetable Compiler for DSP
Member of Technical Staff, Qualcomm

US Army Sensors Lab Research Excellence Award, 1999

19. Chonlameth Aripnikanondt, 2004

System-on-Chip Design with UML
Professor, King Mongkut's University, Thailand.

US Army Sensors Lab Research Excellence Award, 1999

20. Loran Jatunov, 2004

Performance Analysis of 3G CDMA Systems
Senior Research Scientist, Soft Networks, LLC.

21. Antonios Argyriou, 2005, Serving in Hellenic Army.

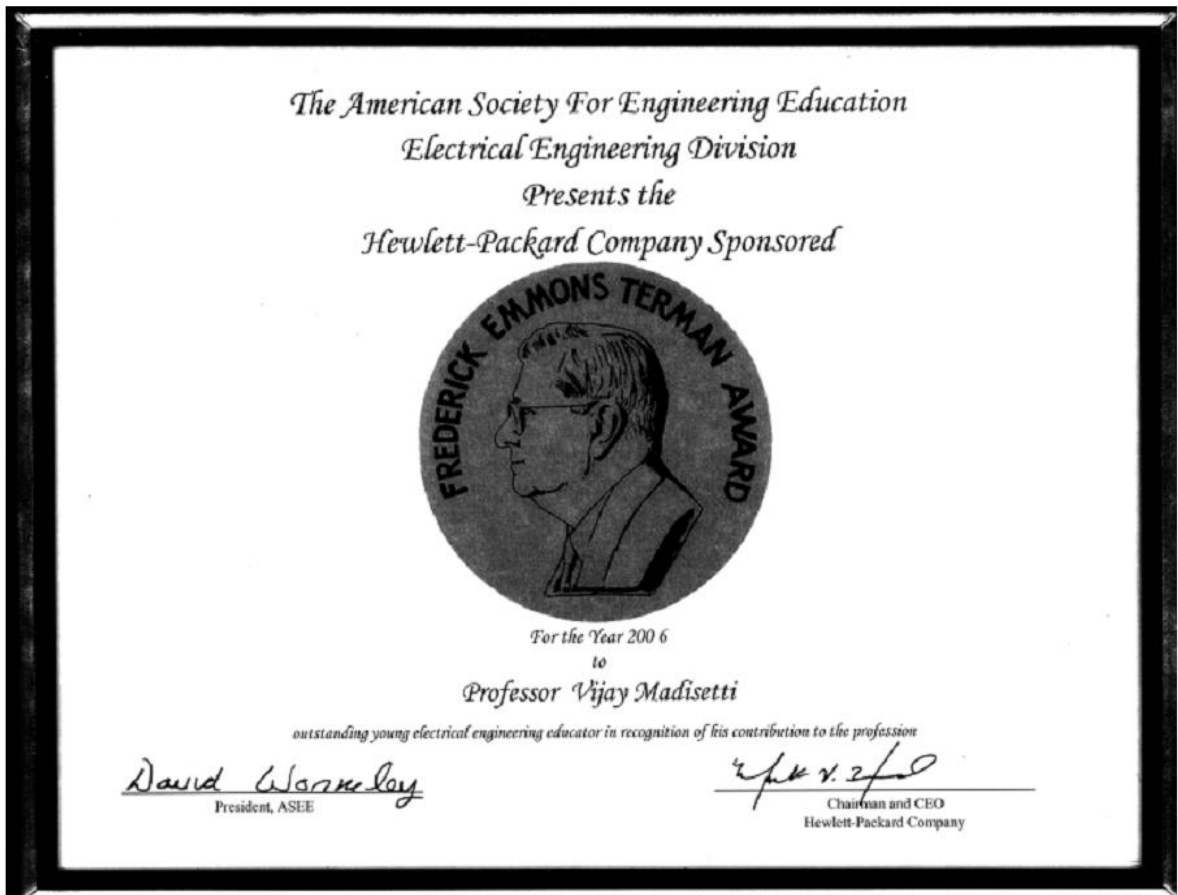
22. Pilho Kim, 2009, Scientist, VP Technologies, Inc.

23. M. Sinnokrot, 2009, Staff Engineer, Qualcomm.

Awards & Honors

1. **Jagadis Bose National Science Talent Fellowship**, Indian Institute of Technology, Kharagpur, 1980-1984.
2. **General Proficiency Prize**, Indian Institute of Technology, Kharagpur, 1984.
3. **Demetri Angelakos Outstanding Graduate Student Award**, Univ. of California, Berkeley, 1989
4. **Ira Kay IEEE/ACM Best Paper Award** for Best Paper presented at IEEE Annual Simulation Symposium, 1989
5. **IBM Faculty Development Award** 1990
6. **Technical Program Chair**, IEEE Workshop on Parallel and Distributed Simulation. 1990.
7. **Technical Program Chair**, IEEE MASCOTS'94
8. **NSF RI Award**, 1990
9. **VHDL International Best PhD Dissertation Advisor Award**, 1997

10. Georgia Tech Outstanding Doctoral Dissertation Advisor Award, 2001.
11. ASEE 2006 Frederick Emmons Terman Medal, 2006.
12. Fellow of IEEE



Intellectual Property Disclosures (Georgia Tech)

<u>Patent</u>	<u>Date</u>	<u>Description</u>
2843	2004	Method and Apparatus for Improving the Performance of Wireless LANs
2825	2003	Method and Apparatus for Optimal Partitioning and Ordering of Antennas for Layered Space-Time Block Codes in MIMO Communications Systems
2815	2003	How to Rapidly Develop a SyD Application
GSU-023	2003	Rapid Development of SyD Applications

2810	2003	System on Mobile Devices Middleware Design
2718	2003	A Transport Layer Algorithm for Improved Anycast Communication
2717	2003	A Novel Transport Layer Load-Balancing Algorithm
2716	2003	A Transport Layer QoS Algorithm
2715	2003	A Novel Transport Layer Algorithm for MPLS Performance
2659	2002	A New Algorithm and Technology for Implementing Mobile IP with Applications to Voice and Video over Mobile IP
2656	2002	Debugging with Instruction-Level Reverse Execution
2655	2002	Embedded Software Streaming
2539		System of Databases: An Enabling Technology for Programming
2517	2002	A Dynamic Instantiated Real-Time Operating System Debugger
2516	2002	A Dynamic Real-Time Operating System
GSU-009	2001	System of Databases: Architecture,, Global Queries, Triggers and Constraints
2480	2001	Mobile Fleet Application based on SyD Technology
2479	2001	System of Databases: A model with coordination links and a calendar application
1893	1999	Beehive
1726	1995	Very High Scale Integrated Circuit Hardware Description Language Models (VHDL Models)
1401	1995	Self-Compensation Receiver (SCR)

Issued Patents: 9,935,772, 9,769,213

Appendix B

APPENDIX B: LIST OF MATERIALS CONSIDERED

Ex. 1001	U.S. Patent No. 8,495,242 (“’242”)
Ex. 1002	File History of U.S. Application No. 12/713,637 (“’242 FH”)
Ex. 1004	U.S. Patent No. 6,421,733 (“Tso”)
Ex. 1005	U.S. Patent No. 6,438,576 (“Huang”)
Ex. 1007	U.S. Patent Application Publication No. US 2002/0078093 (“Samaniego”)
Ex. 1008	U.S. Patent No. 5,420,967 (“Delp”)
Ex. 1009	U.S. Patent No. 5,969,716 (“Davis”)
Ex. 1013	Redline Comparison of specifications from Samaniego and the ’242 patent