UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NetApp, Inc., Rackspace US, Inc.
Petitioner

v.

Realtime Data LLC
Patent Owner

Patent No. 7,378,992

Inter Partes Review No. IPR2017-01663

PETITION FOR *INTER PARTES* REVIEW

UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100 *et seq.*

TABLE OF CONTENTS

Page

dn-191937

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF AUTHORITIES

Page(s)

Cases

Exhibit List for *Inter Partes* Review of U.S. Patent No. 7,378,992

Exhibit Description	Exhibit #
U.S. Patent No. 7,378,992 (the "'992 patent"), includes Reexamination Certificate No. 7,378,992 C1 and Reexamination Certificate No. 7,378,992 C2	1001
Hsu and Zwarico, "Automatic Synthesis of Compression Techniques for Heterogeneous Files," *Software-Practice and Experience*, Vol. 25(10), 1097-1116 (October 1995) ("Hsu")	1002
U.S. Patent No. 5,870,036 ("Franaszek")	1003
Deposition of Patent Owner's Expert, Dr. Kenneth Zeger in IPR2016-00373	1004
U.S. Patent No. 6,253,264 ("Sebastian")	1005
U.S. Patent No. 6,195,024 (the "'491 application")	1006
Declaration of Dr. Daniel Hirschberg ("Hirschberg Decl.")	1007
July 10, 2007 Non-Final Office Action, Application No. 11/400,533	1008
February 6, 2008 Notice of Allowability, Application No. 11/400,533	1009
December 15, 2009 Non-Final Office Action for Reexamination No. 95/000,478	1010
October 28, 2011 Rebuttal Brief for Reexamination No. 95/000,478	1011
March 2, 2012 *Inter Partes* Reexamination Petition for Reexamination No. 95/001,928	1012
August 16, 2013 Right of Appeal Notice for Reexamination No. 95/001,928	1013
Memorandum Opinion and Order Regarding Claim Construction ("Claim Construction Order")	1014
U.S. Patent No. 5,467,087 ("Chu")	1015

Exhibit Description	Exhibit #
Declaration of Dr. Scott Bennett ("Bennett Decl.")	1016
D.A. Lelewer and D.S. Hirschberg, "Data compression," *Computing Surveys* 19:3 (1987) 261-297	1017
International Patent Application Publication No. WO 2001/063772	1018
International Patent Application Publication No. WO 2001/050325	1019

Petitioner NetApp, Inc. and Petitioner Rackspace US, Inc. (collectively "Petitioner" herein) respectfully petition for *inter partes* review (IPR) of claim 48 of U.S. Patent No. 7,378,992 (the "'992 patent" (Ex. 1001)) in accordance with 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42.100 *et seq*. Claim 48 has been challenged in other IPR proceedings. This Petition relies on similar grounds and different grounds as presented in the other proceedings.

I. INTRODUCTION

The '992 patent relates generally to "[s]ystems and methods for providing fast and efficient data compression using a combination of content independent data compression and content dependent data compression." '992 patent at Abstract. For example, claim 48 generally recites a method for compressing a data block by associating coders to data types, analyzing the block to identify a data type by a means that is not based only on a descriptor indicative of the data type, and using a compression encoder associated with the identified data type if one is associated to that type. If no encoder is associated to that data type, then the block is compressed with a default encoder.

The challenged claim, however, does not recite new compression technology, such as a new compression algorithm. Instead, the challenged claim merely claims a well-known way to select a well-known compression algorithm to apply to a data block.

For example, "Automatic Synthesis of Compression Techniques for Heterogeneous Files," by Hsu and Zwarico and published in 1995 ("Hsu," Ex. 1002) discloses a "compression technique for heterogeneous files, those files which contain multiple types of data such as text, images, binary, audio, or animation." The system uses statistical methods to analyze data blocks to determine parameters of the data blocks that are used to "determine the best algorithm to use in compressing each block of data in a file (possibly a different algorithm for each block)." Hsu at Abstract. The Hsu system analyzes the data itself within a data block and compares the pattern of data found to known data patterns to determine which content dependent data compression encoder to apply. Accordingly, Hsu discloses analyzing data within a data block to determine a data type based on something other than a descriptor, and then applying content dependent data compression when the data type is recognized.

Similarly, Franaszek (U.S. Patent No. 5,870,036 (Ex. 1003)) filed February 24, 1995, discloses using selecting compression methods for data blocks based on a data type of the data block. Franaszek at 5:49-5:54. When the data type is not available, Franaszek teaches using one or more default compression methods. *Id*.

The combination of Hsu and Franaszek discloses all of the limitations of the challenged claim—indeed, Patent Owner has already conceded as much in IPR2016-00373. *See* Deposition of Patent Owner's Expert, Dr. Kenneth Zeger

(Ex. 1004) at 28:24-31:11 (acknowledging that each limitation of claim 48 is taught by the combinations of Hsu with Franaszek or Sebastian (U.S. Patent No. 6,253,264 (Ex. 1005)).

As explained below, a person of ordinary skill in the art would have found claim 48 obvious in view of Hsu's and Franaszek's teachings, as well as the other combinations presented below. Accordingly, the challenged claim is obvious and unpatentable.

II. NOTICES AND STATEMENTS

Pursuant to 37 C.F.R. § 42.8(b)(1), NetApp Inc., SolidFire LLC,[1] and Rackspace US, Inc. are each a real party-in-interest.

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner identifies the following related matters. In the Eastern District of Texas, Patent Owner has asserted the '992 patent in Case Nos.: 6-16-cv-01037, 6-16-cv-01035, 6-16-cv-00961, 6-15-cv-00463, 6-15-cv-00464, 6-15-cv-00465, 6-15-cv-00466, 6-15-cv-00467, 6-15-cv-00468, 6-15-cv-00469, 6-15-cv-00470, 6-10-cv-00493, and 6-08-cv-00144. In the Central District of California, Patent Owner has asserted the '992 patent in Case No. 2-16-cv-02743. In the Northern District of California, Patent Owner has asserted the '992 patent in Case Nos.: 3-17-cv-02109 and 3-16-cv-01836. In the Southern District of California, Patent Owner has asserted the '992 patent in Case

[1] SolidFire LLC is wholly owned by NetApp, Inc.

No. 3-12-cv-01048. Claim 48 of the '992 patent is challenged in IPR2016-00373 (terminated) and IPR2016-00980 (pending). The '992 patent was subject to *inter partes* reexamination in reexamination nos. 95/000,478 and 95/001,928.

Pursuant to 37 C.F.R. § 42.8(b)(3), Petitioner identifies the following counsel (and a power of attorney for each petitioner accompanies this Petition).

Lead Counsel for Petitioner	**Backup Counsel for Petitioner**
Diek O. Van Nort DVanNort@mofo.com Registration No.: 60,777 MORRISON & FOERSTER LLP 370 Seventeenth Street, Suite 4200 Denver, CO 80202 Tel: (650) 813-5696 Fax: (303) 592-1510	Jonathan Bockman JBockman@mofo.com Registration No.: 45,640 MORRISON & FOERSTER LLP 1650 Tysons Boulevard McLean, VA 22102 Tel: (703) 760-7769 Fax: (703) 760-7777
Backup Counsel for Petitioner Rackspace US, Inc.	**Backup Counsel for Petitioner Rackspace US, Inc.**
Paul V. Storm pvstorm@gardere.com Registration No.: 35,803 GARDERE 2021 McKinney Avenue Suite 1600 Dallas, TX 75201 Tel: (214) 999-4701 Fax: (214) 999-3701	Sarah M. Paxson spaxson@gardere.com Registration No.: 52,105 GARDERE 2021 McKinney Avenue Suite 1600 Dallas, TX 75201 Tel: (214) 999-4756 Fax: (214) 999-3756

Pursuant to 37 C.F.R. § 42.8(b)(4), service information for lead and backup counsel is provided above. Petitioner consents to electronic service by email to 35667-992-IPR@mofo.com, pvstorm@gardere.com, and spaxson@gardere.com.

Pursuant to 37 C.F.R. § 42.104(a), Petitioner certifies that the ’992 patent is available for *inter partes* review, and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claim on the grounds identified in this Petition.

III. SUMMARY OF THE ’992 PATENT

A. Background of the ’992 Patent

The ’992 patent is directed to a method for analyzing a data block and selecting a compression method for that block. ’992 patent at Abstract. An embodiment of the alleged invention is depicted in, and is described with respect to, Figures 13A and 13B (which were not added to the applications in the ’992 patent’s priority chain until October 29, 2001):

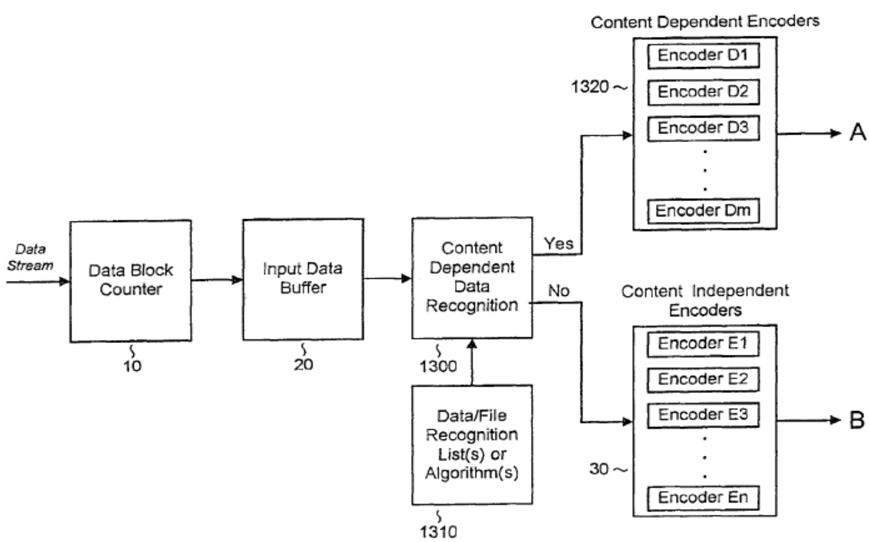

FIGURE 13A

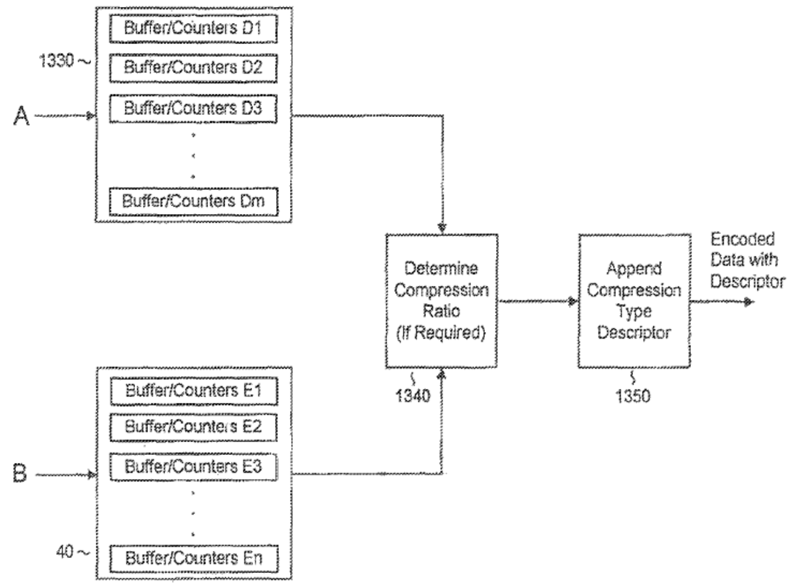

FIGURE 13B

The "content dependent data recognition module 1300 analyzes the incoming data stream to recognize . . . parameters that may be indicative of either the data type/content of a given data block or the appropriate data compression algorithm or algorithms (in serial or in parallel) to be applied." *Id.* at 16:27-33. "Each data block that is recognized by the content data compression module 1300 is routed to a content dependent encoder module 1320, if not the data is routed to the content independent encoder module 30." *Id.* at 16:36-40.

The content dependent encoder module 1320 includes "a set of encoders D1, D2, D3 . . . Dm [that provide] . . . lossless or lossy encoding techniques currently ***well known within the art*** such as MPEG4, various voice codecs, MPEG3, AC3, AAC, as well as lossless algorithms such as run length, Huffman, Lempel-Ziv

Dictionary Compression, arithmetic coding, data compaction, and data null suppression." *Id.* at 16:44-51 (emphasis added).

The content independent encoder module 30 includes "a set of encoders E1, E2, E3 . . . En [that provide] . . . lossless encoding techniques currently ***well known within the art*** such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression." *Id.* at 16:57-64 (emphasis added).

B. Priority Date

The '992 patent includes a priority claim to a long list of U.S. patent applications dating back to Application No. 09/210,491 (the "'491 application"), filed on December 11, 1998, and issued as U.S. Patent No. 6,195,024 (Ex. 1006).

The challenged claim is not supported in the '491 application and thus is not entitled to an effective filing date of December 11, 1998. The '491 application does not provide written description support for determining whether to compress a data block with a content dependent compression technique or a content independent compression technique. It provides no written description support for analyzing data blocks in order to determine a compression algorithm to apply, regardless of whether it is content dependent or content independent. Indeed, the portions of the '992 patent that describe analyzing a data block to determine a compression algorithm to apply (*see, e.g.*, Figs. 13A-13B and associated text) were not included in any of the earlier filed applications until October 29, 2001. Accordingly, to the extent that the challenged claim is supported at all in the '992 patent's specification, the earliest priority date that the challenged claim is entitled is October 29, 2001.[2] Declaration of Dr. Daniel Hirschberg ("Hirschberg Decl.") (Ex. 1007), ¶¶ 19-22.

C. Prosecution History of the '992 Patent

The '992 patent was filed on April 8, 2006 and granted May 27, 2008. It has undergone two *inter partes* reexaminations, during which twenty-three claims were cancelled and six were added.

1. Original Prosecution

During the original prosecution of the patent application that issued as the '992 patent, the Examiner only rejected then pending claim 6 (the claim most closely corresponding to challenged claim 48) based on non-statutory obviousness type double patenting. July 10, 2007 Non-Final Office Action, Application No. 11/400,533 (Ex. 1008) at 2. The Examiner allowed claim 6 to issue after a terminal disclaimer was filed. The Examiner indicated that the prior art failed to teach "compressing, if said first data type is not the same as one of several data types; said data block with a default to provide said compressed data block."

[2] The prior art references relied upon in this Petition are still prior art regardless of whether the '992 patent is entitled to the December 11, 1998 priority date.

February 6, 2008 Notice of Allowability, Application No. 11/400,533 (Ex. 1009) at 3.

Shown below are the differences between claim 6 in the prosecution history (which issued as claim 12 of the '992 patent) and challenged claim 48 (which was added in the second reexamination). The underlining indicates additions to claim 48 as compared to claim 6 and strikethroughs indicate deletions.

> A computer implemented method comprising:
>
> receiving a ~~first~~ data block;
>
> associating at least one encoder to each one of several data types;
>
> analyzing data within the data block to identify~~ing~~ a first data type of ~~said first~~ the data within data block;
>
> compressing, if said first data type is the same as one of said several data types, said data block with said at least one encoder associated ~~to~~ with said one of said several data types that is the same as said first data type to provide a compressed data block; and
>
> compressing, if said first data type is not the same as one of said several data types, said data block with a default encoder to provide said compressed data block;
>
> wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of the data within the data block.

2. First Reexamination

The first *inter partes* reexamination petition (Reexamination No. 95/000,478) relied on eight references to challenged claims of the '992 Patent, including claim 12 (which corresponds to claim 6 in the application that issued in the '992 patent), on various theories of anticipation and obviousness. December 15, 2009 Non-Final Office Action for Reexamination No. 95/000,478 (Ex. 1010) at 3, 9. The Patent Office granted reexamination of claim 12 (and other claims) based on anticipation by Franaszek and Sebastian. *Id.* Patent Owner conceded that Franaszek and Sebastian each anticipated claim 12 (and other claims) when Patent Owner withdrew all of its validity arguments for the challenged claims (including claim 12). October 28, 2011 Rebuttal Brief for Reexamination No. 95/000,478 (Ex. 1011) at 6-9. Accordingly, a reexamination certificate was issued that canceled claim 12 (and other claims) of the '992 patent. Reexamination Certificate No. 7,378,992 C1 (Ex. 1001).

3. Second Reexamination

The second *inter partes* reexamination petition (Reexamination No. 95/001,928) relied on three grounds to challenge three claims that survived the first reexamination, including obviousness in view of Sebastian combined with another reference. March 2, 2012 *Inter Partes* Reexamination Petition for Reexamination No. 95/001,928 (Ex. 1012) at 9. No other references relied upon in the present Petition were included in the second reexamination petition. In the resulting reexamination certificate, the Patent Office canceled two of three challenged claims and confirmed patentability of the third challenged claim after an amendment. Reexamination Certificate No. 7,378,992 C2 (Ex. 1001).

The Patent Office also confirmed several new claims that Patent Owner added during the reexamination, including claim 48, which is the challenged claim in this Petition. The Patent Office confirmed claim 48 over an anticipation rejection in view of Franaszek. August 16, 2013 Right of Appeal Notice for Reexamination No. 95/001,928 (Ex. 1013) at 5. Specifically, the Examiner found that Franaszek alone did not teach claim 48's limitation of "wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of data within the data block." *Id*. In contrast to this Petition, however, the Patent Office never considered whether claim 48 is patentable in view of Franaszek combined with other references that teach this limitation.

IV. LEVEL OF ORDINARY SKILL IN THE ART

As of the priority date of the '992 patent, a person having ordinary skill in the art relevant to the '992 patent would have had an undergraduate degree in computer science, computer engineering, electrical and computer engineering, or an equivalent field as well as one to three years of experience working with data compression or a graduate degree with coursework or research in the field of data compression. Individuals with additional education or additional industrial experience could still be of ordinary skill in the art if that additional education or experience compensated for a deficit in one of the other aspects of the requirements stated above. Hirschberg Decl., ¶¶ 32-33. In this Petition, reference to a person having ordinary skill in the art refers to a person with these qualifications.

V. CLAIM CONSTRUCTION

Pursuant to 37 C.F.R. § 42.100(b), a claim of an unexpired patent is given its broadest reasonable interpretation in light of the specification. *See Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131 (2016). If, however, a patent expires during an IPR proceeding, a claim is interpreted according "the *Phillips* standard for claim construction." *In re CSB-System Int'l, Inc.*, 832 F.3d 1335, 1341 (Fed. Cir. 2016), *cert. denied*, 137 S. Ct. 1384 (2017). Because the '992 patent will likely expire during the IPR, for the purposes of this proceeding, all terms have their meaning according to *Phillips*, "which emphasizes considering the plain meaning of the claim terms themselves in light of the intrinsic record." *Id.* at 1340 (citing *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-15 (Fed. Cir. 2005) (*en banc*)). This claim construction standard, however, does not necessarily result in a narrower construction. *Id.* at 1337-38 ("We conclude, however, that the Board's claim construction was correct even under the *Phillips* standard, and we affirm its rejection of all claims of the [challenged] patent as unpatentable over the prior art.").

Petitioner does not believe that any constructions are necessary to resolve this IPR. Below, however, Petitioner sets forth relevant constructions from the related district court case involving the '992 patent. For purposes of this Petition, the prior art is applied to the challenged claims consistent with these constructions.

A. "analyzing"/"analyze"

Patent Owner, in the district court case against Petitioner, has asserted that "analyzing"/"analyze" means "directly examining / directly examine." Claim Construction Order (Ex. 1014) at 32. Additionally, the court adopted this construction in the claim construction order. *Id.* at 34. Accordingly, for the purposes of this Petition this construction is applied

B. "default encoder"

In the district court case against Petitioner, Patent Owner and Petitioner agreed that this term means "an encoder used automatically in the absence of a designated alternative." *Id.* at 35. Accordingly, for the purposes of this Petition this construction is applied.

VI. GROUNDS OF REJECTION

Petitioner requests cancellation of claim 48 of the '992 patent in view of the following references and grounds:

- **Ground 1:** Claim 48 is obvious over Hsu[3] in view of Franaszek;[4]
- **Ground 2:** Claim 48 is obvious over Hsu in view of Sebastian;[5]
- **Ground 3:** Claim 48 is obvious over Franaszek in view of Hsu; and
- **Ground 4:** Claim 48 is obvious over Franaszek in view of Chu.[6]

All of the above references were cited during the original prosecution or the reexaminations of the '992 patent, along with hundreds of other references. While Franaszek has previously been found not to anticipate claim 48, the combination of Franaszek with Hsu or Chu was not applied to claim 48 during the original prosecution of the '992 patent or during either of the reexaminations. In IPR2017-00373 and IPR2017-00179, the PTAB has instituted *inter partes* review of claims of U.S. Patent Nos. 7,161,506 and 9,054,728, respectively, which are related to the '992 patent and include claims directed to the same subject matters as the '992 patent. Both of these IPRs were instituted based on the combination of Franaszek with other references, including Hsu.

---

[3] Exhibit 1002. Hsu qualifies as prior art under §§ 102(a) and (b).

[4] Exhibit 1003. Franaszek qualifies as prior art under §§ 102(a), (b), and (e).

[5] Exhibit 1005. Sebastian qualifies as prior art under §§ 102(a) and (e).

[6] U.S. Patent No. 5,467,087, Exhibit 1015. Chu qualifies as prior art under §§ 102(a), (b), and (e).

This Petition, supported by the Declaration of Dr. Daniel Hirschberg filed herewith (Ex. 1006), demonstrates that there is a reasonable likelihood that Petitioner will prevail in demonstrating that the challenged claim is not patentable. *See* 35 U.S.C. § 314(a).

VII. DETAILED EXPLANATION UNDER 37 C.F.R. § 42.104(B)

Pursuant to Rule 42.104(b)(4)-(5), specific grounds identified below and discussed in the Hirschberg Declaration show in detail the prior art disclosures that make the challenged claim obvious.

A. Ground 1: Claim 48 is Obvious in View of Hsu and Franaszek

Hsu was among hundreds of prior art references cited by the Patent Owner during the reexaminations but was never substantively applied to or discussed with respect to claim 48 of the '992 patent during the original prosecution or either reexamination proceeding. As noted above, however, by granting another IPR, IPR2016-00373, based on Hsu that challenged claim 48, the PTAB has previously found that Hsu combined with other references is reasonably likely to invalidate claim 48.

Indeed, in IPR2016-00373, Patent Owner's expert conceded that the combination of Hsu and Franaszek or Sebastian includes all limitations of claim 48. Deposition of Patent Owner's Expert in IPR2016-00373 at 28:24-31:11 (Patent Owner's expert confirming that each limitation of claim 48 is disclosed by

Hsu combined with Franaszek or Sebastian). As explained below, in addition to Hsu and Franaszek disclosing each limitation of claim 48, this combination of references also renders claim 48 obvious. *See* Hirschberg Decl., ¶¶ 56-63.

1. Summary of Hsu

Hsu was published in the journal *Software—Practice and Experience* (a Wiley-Interscience Publication) in October 1995. Hsu at Cover;[7] Declaration of Dr. Scott Bennett ("Bennett Decl.") (Ex. 1016), ¶ 22. Dr. Bennett confirms that a copy of Hsu was available in a library in 1995 and that Hsu was accessible to a person of ordinary skill in the art. Bennett Decl., ¶¶ 36-37. Specifically, in Dr. Bennett's opinion, Hsu was available in 1995, from numerous libraries, and an interested party would have been able to locate Hsu in those libraries without issue. *Id.*, ¶¶ 28-35. This is not surprising, as Hsu is a journal article that was intended for distribution among those interested in computer software, including compression software. Hsu at Editor's page 2. Accordingly, Hsu is prior art under 35 U.S.C. §§ 102(a) and (b).

Like the '992 patent, Hsu discloses a technique to select well-known compression techniques based on a data type of a data block. For example, Hsu discloses selecting among run length, Huffman, Lempel-Ziv, and arithmetic

---

[7] Under Rule 803(16) of the Federal Rules of Evidence, Hsu qualifies for the ancient-document exception to the rule against hearsay.

encoding, which are the same compression techniques as those identified in the ’992 patent. *Compare* Hsu at 1107 (Table I) *with* ’992 patent at 16:45-51. Hsu recognizes that using such a “system provides better space savings.” Hsu at Abstract.

Hsu’s compression system includes two phases. A first phase that analyzes data blocks to determine compression encoders[8] to use (the analysis phase). A second phase that compresses the data blocks and ensures that negative compression (i.e., expansion) does not occur (the compression phase).

In the first phase (the analysis phase), the data within each data block to be compressed is analyzed. Four routines are used to determine a type of the data.

---

[8] Hsu refers to a “compression algorithm” instead of a compression encoder. *See, e.g.*, Hsu at Abstract, 1106. The challenged claim uses the term compression “encoder.” *See, e.g.*, ’992 patent at Claim 48. The rest of the ’992 patent uses the terms “encoder” and “algorithm” interchangeably. *See, e.g.*, *id.* at 23:18-24 (“The system of FIGS. 17*a* and 17*b* additionally uses a priori estimation algorithms or look-up tables to estimate the desirability of using content independent data compression ***encoders*** and/or content dependent data compression ***encoders*** and selecting appropriate ***algorithms or subsets thereof*** based on such estimation.” (emphasis added)). For ease of discussion, the Petition uses “compression encoder” which is the same as a “compression algorithm” in the context of the ’992 patent.

One routine determines a "data classification" of a data block by analyzing 512-byte segments at the beginning, middle, and end of the data block. The other three routines each determine a "redundancy metric" that indicates the "compressibility of a block of data" by various types of compression encoders. Hsu at 1104.

Hsu's three redundancy metrics are based on statistical measures of the data blocks:

- "the degree of variation in character frequency or alphabetic distribution" (MAD) measures the relative frequencies of repeating units or characters;
- "the average run length of the block" (MRL) measures the average length of "long strings of identical units occurring next to one another"; and
- "the string repetition ratio of the block" (MSR) measures the amount of repetition of identical strings of units or characters that may or may not occur next to one another.

Hsu at 1104. After normalization, the largest metric that is also above a threshold (e.g., 2.5) is considered to be the largest redundancy metric. *Id.* at 1106.

If the first phase identifies a useful data type (i.e., a combination of data classification and largest redundancy metric with a corresponding associated encoder in Table I) present in the data to be compressed, the data type is used to identify a compression algorithm for the data block, as shown in Table I. This Petition focuses on the combination of the data classification and largest redundancy metric as the recited "data type" of claim 48. The data classification and largest redundancy metric individually, however, could each be a "data type" as well.

Table I. Database of compression algorithms†

	M_{AD}	M_{RL}	M_{SR}
ANSI	arithmetic coding *	run-length encoding byte-wise encoding	Lempel–Ziv freeze
hexadecimal	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
natural language	arithmetic coding *	* *	Lempel–Ziv freeze
source code	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
low redundancy binary	* *	run-length encoding n-bit run count	Lempel–Ziv *
audio	* *	run-length encoding byte-wise encoding	Lempel–Ziv freeze
low resolution graphic	* *	run-length encoding n-bit run count	Lempel–Ziv freeze
high resolution color graphic	JPEG improved Huffman	run-length encoding n-bit run count	JPEG improved Huffman
high redundancy binary	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
object	arithmetic coding *	run-length encoding byte-wise encoding	Lempel–Ziv freeze

† Note: the first line of each entry is the basic algorithm and the second line is the heuristic. An * as the heuristic indicates that no heuristic is used. Two * indicates no entry.

Id. at 1107 (annotated to show the ten rows for the ten categories of data). Note that four of the ten data classifications have a redundancy metric for which there is no assigned compression algorithm (those rows are marked with two asterisks (*)). These are non-useful data types that have no associated compression encoder. In Hsu, a data block that produces one of these four data types is not compressed. *Id.* at 1106-07. If a useful data type is identified, an identifying tag for the associated compression encoder from the database (represented by Table I in Hsu) is stored in a compression plan that identifies the compression encoder to be used with the particular data block.

Once all of the data blocks have been analyzed in the first phase, the second phase (the compression phase) applies the compression encoders identified in the compression plan to the data blocks. The second phase also ensures that compressing a data block does not result in negative compression. *Id.* at 1109. If negative compression is detected, the uncompressed data block is stored instead of the compressed data block. *Id.*

Hsu explicitly discloses every limitation of the challenged claim except for the limitations related to the "default encoder." Hsu in view of Sebastian or Franaszek, however, renders the challenged claim obvious because it would have been obvious to a person of ordinary skill in the art to implement Hsu's system with a default encoder when a useful data type is not identified and a corresponding compression encoder is not identified based on Hsu's analysis phase. As explained in Sections VII.A.2 and VII.B.1 below, Franaszek and Sebastian each expressly teach that a "default encoder," as recited in the challenged claims, can be used in compression systems similar to Hsu's. Furthermore, it would have been obvious to implement their default encoders in Hsu's system. Accordingly, the challenged claims are obvious in view of the combination of Hsu and Sebastian or Franaszek.

2. Summary of Franaszek

Franaszek discloses a compression system that "dynamically compress[es] data blocks by using the best of a given set of methods." Franaszek at 3:25-26; Hirschberg Decl., ¶¶ 45-53. When a data block is received for compression, Franaszek analyzes the data block to determine whether the contents of a "data type" entry in the type field is available to identify the type of data within the data block. Franaszek at 5:49-53, Fig. 4A. If the "data type" is determined from the type field, a "Compression Method List (CML)" is assigned to the data block based on the data type. *Id.* at 5:49-53, 6:1-7, Fig. 4A. If no "data type" can be identified from the contents of the type field, then a default CML is assigned to the data block. *Id.* at 5:53-54, 6:8-11, Fig. 4A. The data block is then compressed with the one or more methods identified in the CML. *Id.* at 6:21-32. The compressed data block with the best compression ratio is then output along with an indication of which compression encoder was used to produce it. *Id.* at 6:33-50. That is, Franaszek teaches automatically applying an encoder from the default CML list to the data block when a designated alternative encoder (i.e., a data type specific encoder) is not identified from a data type. Hirschberg Decl., ¶¶ 45.

3. Independent Claim 48 Is Obvious in View of Hsu and Franaszek

All of the limitations recited in claim 48 except for limitation E are expressly disclosed in Hsu. *See* Hirschberg Decl., ¶¶ 45-53. In view of the knowledge of a person of ordinary skill in the art and Franaszek, it would have been obvious to implement Hsu to include any missing limitations of claim 48 as explained in detail below.

Element	Claim 48
48[PR]	1. A computer implemented method comprising:
48[A]	receiving a data block;
48[B]	associating at least one encoder to each one of several data types;
48[C]	analyzing data within the data block to identify a first data type of the data within the data block;
48[D]	compressing, if said first data type is the same as one of said several data types, said data block with said at least one encoder associated with said one of said several data types that is the same as said first data type to provide a compressed data block; and
48[E]	compressing, if said first data type is not the same as one of said several data types, said data block with a default encoder to provide said compressed data block,
48[F]	wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of the data within the data block.

a. Hsu Discloses the Preamble of Claim 48

Hsu discloses "[a] computer implemented method ," as recited by the preamble of claim 48. For example, Hsu discloses a computer-implemented system and method of compressing "heterogeneous files," which are "those files which contain multiple types of data such as text, images, binary, audio, or animation." Hsu at Abstract. Hsu's "heterogeneous compressor" system includes "a driver module, four block analysis modules, and the synthesis module, which includes the database of compression algorithms." *Id.* at 1108. The software portion of the system is written in the C programming language and runs on a Sun workstation with a Unix operating system. *Id.* at 1108, 1110. Thus, Hsu discloses the preamble of claim 48.

b. Hsu Discloses Limitation A of Claim 48

Hsu discloses "receiving a data block," as recited by limitation A of claim 48. For example, Hsu describes that the test files that are compressed on a Sun workstation are transferred to the Sun workstation from various external sources, including an Apple Macintosh computer and an MS-DOS-based computer (e.g., an IBM PC). Hsu at 1110. For example, Hsu tests the system performance on a set of twenty test files. *Id.* at 1018-21. Hsu describes that the test files are transferred to a Sun workstation from various external sources, including an Apple Macintosh computer and an MS DOS-based computer (e.g., an IBM PC). *Id.* at 1110. This requires that the test files, which contain a "data block" as part of the data stream of one or more test files, to have been received on the Sun workstation. Similarly, Hsu's "compression technique for heterogeneous files" first requires that Hsu's compression system receive a file so that it may "determine the best algorithm to use in compressing *each block of data* in a file" or data stream. *Id.* at 1097 (emphasis added). Accordingly, Hsu discloses a system for compressing data that includes receiving a data block, as recited in limitation A of claim 48.

c. Hsu Discloses Limitation B of Claim 48

Hsu discloses "associating at least one encoder to each one of several data types," as recited by limitation B of claim 48.

Hsu's "heterogeneous compressor" system decides among various compression encoders when determining how to compress a given data block. "The compressibility of a block of data and the appropriate algorithm to do so are determined by the type of data contained in a block and the type of redundancy (if any) in the data." Hsu at 1103. Table I discloses these various compression encoders. *Id*. at 1107. Specifically, the encoders associated with useful combinations of data classifications and largest redundancy metrics included in Table I show that Hsu discloses "associating at least one encoder to each one of several data types," as recited in claim 48.

Additionally, the compression encoders listed in Hsu's Table I are many of the same "well known" compression encoders listed in the '992 patent as examples of encoders. For example, Hsu's Table I includes arithmetic coding, run-length coding, Lempel-Ziv, and improved Huffman compression encoders.

Table I. Database of compression algorithms†

	M_{AD}	M_{RL}	M_{SR}
ANSI	arithmetic coding *	run-length encoding byte-wise encoding	Lempel–Ziv freeze
hexadecimal	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
natural language	arithmetic coding *	* *	Lempel–Ziv freeze
source code	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
low redundancy binary	* *	run-length encoding n-bit run count	Lempel–Ziv *
audio	* *	run-length encoding byte-wise encoding	Lempel–Ziv freeze
low resolution graphic	* *	run-length encoding n-bit run count	Lempel–Ziv freeze
high resolution color graphic	JPEG improved Huffman	run-length encoding n-bit run count	JPEG improved Huffman
high redundancy binary	arithmetic coding *	run-length encoding n-bit run count	Lempel–Ziv freeze
object	arithmetic coding *	run-length encoding byte-wise encoding	Lempel–Ziv freeze

† Note: the first line of each entry is the basic algorithm and the second line is the heuristic. An * as the heuristic indicates that no heuristic is used. Two * indicates no entry.

Hsu at 1107 (annotated to show rows). Similarly, the ’992 patent includes “run length, Huffman, Lempel-Ziv Dictionary Compression, [and] arithmetic coding” as content dependent data compression encoders. ’992 patent at 16:49-56.

d. Hsu Discloses Limitation C of Claim 48

Hsu discloses “analyzing data within the data block to identify a first data type of the data within the data block,” as recited by limitation C of claim 48.

Hsu’s “heterogeneous compressor” system runs on a Sun workstation that has a processor that executes the software part of the system. As explained above, the software includes two phases: an analysis phase and a compression phase. In the analysis phase, Hsu’s software attempts to identify a useful data type based on a data classification and largest redundancy metric. Hsu at 1103. If a specific useful data type is identified, the data type is used to determine an associated compression encoder from the compression encoder database, the contents of which are shown in Table I. Hsu at 1107.

The analysis proceeds by running a program called "new-file" on three 512-byte segments of data from the data block that is being analyzed (specifically, the segments are from the beginning, middle, and end of the data block). Hsu at 1104. The "new-file" program returns a data type for the data block. In short, *the data within the data block* is analyzed by specific means in an attempt to determine a specific data type. No "descriptor that is indicative of the data type of the data within the data block" is used by Hsu when attempting to determine the data block's specific data type.

e. Hsu Discloses Limitation D of Claim 48

Hsu discloses "compressing, if said first data type is the same as one of said several data types, said data block with said at least one encoder associated with said one of said several data types that is the same as said first data type to provide a compressed data block," as recited by limitation D of claim 48. For example, when Hsu's system identifies a useful data type (e.g., combination of data classification and redundancy metric with an entry in Table I), the data block is compressed using the associated compression encoder from the database (represented in Hsu's Table I). Hsu at 1102 ("[U]sing the block type and largest metric, the appropriate compression algorithm (and possible heuristic) are chosen from the compression algorithm database. The compression method for the current block is then recorded in a small array-based map of the file[.]"), 1109. The entries in Hsu's database are selected to be optimized for the corresponding data type. *Id.* at 1106-1108. Accordingly, Hsu's system compresses data blocks that are identified to be of a useful data type with a content dependent compression encoder selected from Table I that is associated with the useful data type.

f. Implementing Limitation E of Claim 48 in View of Hsu and Franaszek Would Have Been Obvious

A person of ordinary skill in the art would have found it obvious to use the processor in Hsu's "heterogeneous compressor" system to compress, "if said first data type is not the same as one of said several data types, said data block with a default encoder to provide said compressed data block," as recited by limitation E of claim 48. *See* Hirschberg Decl., ¶¶ 63-68. This is especially true in view of Franaszek's express teaching of using a default data compression encoder when a data type with one or more associated encoders is not identified. More specifically, as discussed above, Franaszek teaches automatically applying an encoder from the default CML list to a data block when a designated alternative encoder (i.e., a data type specific encoder) cannot be identified based on a data type. Hirschberg Decl., ¶ 45.

Indeed, Hsu discloses many of the same compression encoders that Patent Owner cited as examples of "a default encoder." Specifically, the '992 patent provides "content independent lossless compression" encoders as an example of "a default encoder." '992 patent at 20:50-51. The cited content independent encoders are well-known lossless encoders, such as "run length, Huffman, Lempel-Ziv Dictionary Compression, [and] arithmetic coding." '992 patent at 16:57-64. Hsu discloses using the same types of well-known lossless compression encoders, including arithmetic coding, run-length encoding, Lempel-Ziv, and improved Huffman compression encoders. Hsu at 1107. Accordingly, Hsu discloses the same types of compression encoders that the '992 patent discloses as examples of "a default encoder."

As explained below, there are several reasons why the combination of Hsu and Franaszek render claim 48 obvious. Specifically, a person of ordinary skill in the art would have been motivated to implement Hsu's system with a default encoder as taught in Franaszek to increase the compression ratio of the system; and would have found implementing Hsu's system with a default encoder as taught in Franaszek to be nothing more than using existing technologies to produce predictable results. These reasons individually or in combination show that Hsu and Franaszek render claim 48 obvious.

(i) A Person of Ordinary Skill in the Art Would Have Been Motivated to Implement Hsu's System With "a Default Encoder" Such As Taught in Franaszek to Provide for Maximum Compression

It would have been obvious to a person of ordinary skill in the art to combine the teachings of Hsu and Franaszek. *See* Hirschberg Decl., ¶¶ 56-63. Both references are in the same technical field of data compression. *See, e.g.*, Hsu at 1097; Franaszek at Abstract, 1:6-9. Both teach that some compression methods (e.g., Lempel-Ziv) effectively compress many but not all types of data. Both also are directed to compressing a collection of data blocks containing different data parameters or attributes—such as different data types. *See, e.g.*, Hsu at 1097, 1098 (discussing effectiveness in terms of a "per cent savings for a compressed file" in terms of file "length"), 1099-1100, 1106-08; Franaszek at 3:25-28 (describing "an object of this invention [is] to dynamically compress data blocks by using the best of a given set of methods."), 6:59-67 (describing "increase[ing] the total number of bytes of uncompressed data blocks that can be stored before the memory becomes full."); *see also* '992 patent at 16:29-31 ("data types, data structures, data block formats, file substructures, file types, and/or any other parameters"). Indeed, it was well known that different types of data would be compressed to different degrees by different compression algorithms. *Id.*, ¶¶ 28-30 (discussing Ex. 1017).

Franaszek expressly teaches using a default data compression encoder when a data type with an associated encoder is not identified in a data block.

> In step 401, if a data type (e.g. text, image, etc.) for a given uncompressed block B is available, in step 404 the Compression Method List (CML) is set to a list of compression methods that have been preselected for that data type. ***Otherwise, if no data type is available, in step 407 the CML is set to a default list of compression methods***.

Franaszek at 5:49-54 (emphasis added); Hirschberg Decl., ¶ 45. That is, Franaszek teaches automatically applying an encoder from the default CML list to the data block when a designated alternative encoder (i.e., a data type specific encoder) is not identified from a data type. Hirschberg Decl., ¶¶ 45. It would have been obvious to implement Hsu's compression system to use the default encoder as taught in Franaszek.

Petitioner is not asserting that the exact system in Franaszek in its entirety should be combined with Hsu's "heterogeneous compressor" system. Rather, Hsu teaches a skilled artisan the benefits of a compression system that chooses a compression encoder for each data block based on its suitability to compress that data block. Franaszek teaches that it is beneficial and reasonable in a similar system to use a default data compression encoder when a type of data is not identifiable as being assigned to existing compression encoders. A person of ordinary skill in the art would have been motivated to use a "heterogeneous compressor" system, as taught in Hsu, with a default encoder when a data type associated with an encoder is not identified, as expressly taught in Franaszek. The above teachings and combination are independent of whether the physical systems in Hsu and Franaszek are physically combinable. *Allied Erecting & Dismantling Co. v. Genesis Attachments, LLC*, 825 F.3d 1373, 1381 (Fed. Cir. 2016) ("The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference." (*quoting In re Keller*, 642 F.2d 413, 425 (C.C.P.A. 1981))). The same is true of the other combination presented in this Petition.

A person of ordinary skill in the art would have been motivated to modify Hsu's "heterogeneous compressor" system to use a default data compression encoder as taught in Franaszek. While Hsu teaches that, in response to failing to identify a useful data type (i.e., a combination of data classification and largest redundancy metric that has an associated encoder in Table I), no compression is to be performed, Hsu's suggestion to not perform compression is an effort to conserve computing resources. Hsu at 1106; Hirschberg Decl. ¶ 61. A person of ordinary skill in the art implementing Hsu's "heterogeneous compressor" system under different circumstances, such as where computing resources are less limited or where the compression ratio needs to be as high as possible, would have been motivated to use a default data compression encoder when a useful data type is not identified. *Id.*, ¶¶ 61-63.

A person of ordinary skill in the art would have recognized Hsu's teaching of foregoing compression when a useful combination of specific data type and largest redundancy metric is not identified leaves open the possibility that the data block is, in fact, compressible, but for some reason, the data was misclassified. For example, Hsu recognizes that its system would need to be expanded and modified to account for new data types. Hsu at 1114. Additionally, a skilled artisan would have recognized that the continually increasing number of data types in the late 1990s and early 2000s need to be handled in Hsu's compression system. Hirschberg Decl., ¶ 62 (citing International Patent Application Publication Nos. WO 2001/063772 (Ex. 1018) at 5 and WO 2001/050325 (Ex. 1019) at 10-11 that describe the need for systems to handle unknown data types). If Hsu's system was used on data blocks that included new data formats before the system was updated to handle the new data formats, the data blocks with the new data formats might be misidentified so that a useful combination of data classification and largest redundancy metric (e.g., any of the entries Table I that specify a compression encoder) is not identified. Hirschberg Decl., ¶¶ 62-63. In this case, based on Hsu's teachings, a data block having the new data format would be marked for no compression even though the data block could be compressible (perhaps even highly compressible). *Id*.

One obvious solution to this problem of incorrect tagging of data blocks as uncompressible and losing compression performance is to use a default data compression encoder to attempt to obtain some compression before marking it for no compression. *Id.*, ¶ 61. While this modification would increase utilization of computing resources by attempting to compress all data blocks, this modification also would result in a higher overall compression ratio (assuming at least one data block having a non-useful data type was actually compressible). *Id.*, ¶¶ 60-63. Specifically, it would sometimes produce a compressed data block that is smaller than the original data block even though the data type indicated that the data block has poor potential for compression and had an empty field for the compression encoder in the compression database (e.g., Table I of Hsu). A person of ordinary skill in the art would have recognized that a higher compression ratio is beneficial in many common scenarios, such as archival systems and those system with limited storage, and that modifying Hsu's system as taught in Franaszek is one way to achieve that beneficial result. *Id.*, ¶ 61.

(ii) The Combination of Hsu and Franaszek Does Nothing More Than Implementing One Known Technology with Another Known Technology to Produce Predictable Results

Claim 48 is nothing more than the predictable result of combining Hsu's known compression system and Franaszek's known default data compression encoder. Making such a combination would have been well within the skill of a person of ordinary skill in the art. *Id.*, ¶ 57. A person of ordinary skill in the art could have implemented functionality to use a default data compression encoder using the most basic of programming techniques. Using the C programming language, which Hsu disclosed as being used to implement the "heterogeneous compressor" system, standard features such as "if . . . else" statements could be used. *Id.* , ¶¶ 57-59. Specifically, using an "if" statement, the system would select an associated compression encoder to compress the data block if the system identified a useful data type (e.g., when a combination of data classification and redundancy metric have an associated compression encoder in the database of Table I). *Id.* In the case of a non-useful data type, using an "else" statement, the system would use the default data compression encoder for use with the data block because the combination of data classification and largest redundancy metric did not have an associated encoder (i.e., an empty field in Table I). *Id.* The straightforward nature of the modification above and its predictable result show that claim 48 is nothing but an obvious combination of Hsu and Franaszek. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417-18 (2007) (predictable use of one technology for another likely to be obvious if the elements are used for their established functions).

Claim 48 is also obvious because it requires nothing more than substituting one known solution (i.e., no compression of data at all if a non-useful data type is identified, as taught in Hsu) with another known solution (i.e., compression of data using a default compression encoder if a non-useful data type is identified, as taught in Franaszek) to yield predictable results. Such a combination is obvious. *KSR,* 550 U.S. at 416 ("[W]hen a patent claims a structure already known in the prior art that is altered by the mere substitution of one element for another known in the field, the combination must do more than yield a predictable result.").

Furthermore, there are a finite number of options for handling the failure to identify a useful data type for a data block in Hsu's system. One option disclosed in Hsu is to not compress the data block at all because of the potential for wasting computer resources.[9] Hsu at 1106. While Hsu discloses this (no compression) as

[9] To the extent that Patent Owner argues this portion of Hsu teaches away from using a default encoder when the system identifies a non-useful combination of data classification and largest redundancy metric, Patent Owner is incorrect. Teaching away occurs when "references taken in combination would produce a

one option, another option that would have been obvious to try to a person of ordinary skill in the art was to use a default data compression encoder, which was a well-known option as taught in Franaszek. Hirschberg Decl., ¶¶ 59-60; *KSR*, 550 U.S. at 421 ("[A] person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely that product [was] not of innovation but of ordinary skill and common sense. In that instance the fact that a combination was obvious to try might show that it was obvious under § 103."). As explained in the previous paragraph, a

---

'seemingly inoperative device.'" *McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1354 (Fed. Cir. 2001) (citing *In re Sponnoble*, 405 F.2d 578, 587 (C.C.P.A. 1969)). Altering Hsu as described in this section would not render Hsu inoperable for its intended purpose. In fact, altering Hsu as described in this section could make Hsu's system more appropriate for its intended purpose by increasing the achieved "space savings." *See* Hsu at Summary (stating that a goal of the described system is to provide for "better space savings"). Accordingly, Hsu's statement regarding one option for handling a non-useful combination of data type and largest redundancy metric does not "teach away" from all other options. Rather, Hsu's statement merely provides an option that is suitable when "overhead" is an issue.

person of ordinary skill in the art would have recognized this modification as a simple, straightforward solution to the above problem.

g. Hsu Discloses Limitation F of Claim 48

Hsu discloses "wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of the data within the data block," as recited by limitation F of claim 48.

As explained above regarding Limitation C, the Hsu system analyzes data within the data block to identify a data type of the data within the data block. Hsu's system uses the "new-file" program on three 512-byte segments of data from the data block that is being analyzed (specifically, the segments are from the beginning, middle, and end of the data block). Hsu at 1104. The "new-file" program returns a data classification for the data block. In other words, *the data within the data block* is analyzed in an attempt to determine a specific data type.

Hsu's analysis phase also includes analyzing data within the data block to calculate redundancy metrics, which are character frequencies or alphabetic distribution (the relative frequencies of repeating units or characters in the block), average run length (the average length of identical strings of units or characters that occur next to one another), and string repetition (the amount of repetition of identical strings of units or characters that may or may not occur next to one another). Hsu at 1104. Like the determination of the specific data type described in the previous paragraph, the redundancy metric analysis is also based on analysis of the data within the data block. "[A] descriptor that is indicative of the data type of the data within the data block" is not used in calculating the redundancy metrics. '992 patent at Claim 48. For example, the average run-length metric is calculated by dividing the number of bits in a block by the number of runs (repetition of symbols either in bits or bytes). *Id.* at 1105.

After normalization, the redundancy metric that yields the highest number is considered to be the largest redundancy metric. *Id.* at 1106. If a useful data type is identified from the combination of the specific data classification and the largest redundancy metric, Hsu's "heterogeneous compressor" system selects the optimal compression encoder for the data block corresponding to that combination from the database. *Id.* at 1102.

From this disclosure of Hsu, it is clear that Hsu's system does not use "only [] a descriptor that is indicative of the data type of the data within the data block" when attempting to determine the data block's data type. Thus, Hsu discloses limitation F of claim 48.

Accordingly, the combination of Hsu and Franaszek disclose and teach all limitations of claim 48. Therefore, a person of ordinary skill in the art would have found claim 48 obvious in view of Hsu and Franaszek for the reasons explained above.

B. Ground 2: Claim 48 is Obvious in View of Hsu and Sebastian

1. Summary of Sebastian

Similar to the ’992 patent and Hsu, Sebastian describes a compression system and method to select a compression encoder based on the data type of a data block. *See* Sebastian at Abstract. Sebastian’s “Base-Filter-Resource” (BFR) system applies data-type-specific compression to a variety of data types. *Id.* at 1:45-50.

Sebastian’s system includes different compression encoders (alternatively referred to as “filters” in Sebastian) that each support a specific data type (alternatively referred to as the “data format” of a “source data” in Sebastian). For example, Sebastian’s system includes compression encoders for data types such as Excel XLS worksheets or Word DOC files. *Id.* at 1:50-52. If an available compression encoder “matches the format of the data to be encoded, the advantages of format-specific compression can be realized for that data.” *Id.* at 1:55-57. If no “installed” compression encoder matches with the data type to be compressed, a “generic” compression encoder is used. *Id.* at 1:55-60; *see also id.* at 4:9-23 (suitable generic filters include Lempel-Ziv variants). Sebastian teaches that the use of a “generic” compression encoder is beneficial because it allows the compression system to handle a "wide range of data formats," including formats unknown by the compression system. *See id.* at 1:45-60. Sebastian also recognizes that the choice of the best compression algorithm may be based on system resources and other factors, including "the current availability of memory and the desired trade-off between speed and compression performance." *Id.* at 17:49-52. Accordingly, Sebastian teaches automatically applying an encoder to a data block when a designated alternative encoder (i.e., a data type specific encoder) cannot be identified due to an unsupported data type. Hirschberg Decl., ¶ 44.

2. Independent Claim 48 Is Obvious in View of Hsu and Franaszek

a. Hsu Discloses the Preamble and Limitations A-D, and F of Claim 48

As explained above in Sections VII.A.3.a-e and VII.A.3.g, Hsu discloses the Preamble and limitations A-D and F of claim 48.

b. Implementing Hsu in View of Sebastian to Include Limitation E Claim 48 Would Have Been Obvious

As explained above, Hsu and Sebastian are both directed to compression systems that determine what compression encoder will be used based on the type of data that is to be compressed. Sebastian provides an additional express teaching to use a default data compression encoder when analysis of the data within the data block does not identify a data type corresponding to an appropriate compression encoder. Hirschberg Decl., ¶ 44. Sebastian, summarized in detail in Section

VII.B.1, describes a compression system similar to Hsu's system. Sebastian's system analyzes the content of a data source (e.g., a file) to identify whether its data format is supported by one of the preexisting compression encoders in the system. Sebastian at 4:47-50, 5:24-26. Importantly, Sebastian expressly teaches using a default data compression encoder, "such as Lempel-Ziv (LZ)" encoder, when a data source is unsupported by the existing compression encoders. *Id.* at 4:9-15. That is, as explained above, Sebastian teaches automatically applying an encoder to a data block when a designated alternative encoder (i.e., a data type specific encoder) cannot be identified due to an unsupported data type. Hirschberg Decl., ¶ 44.

This teaching provides a further reason why a person of ordinary skill in the art would have found any differences between Hsu and claim 48 obvious. For the same reasons as those explained above with respect to Franaszek in Section VII.A.3.f, implementing limitation E in Hsu in view of Sebastian would have been obvious and claim 48 is thus obvious in view of Hsu and Sebastian. Hirschberg Decl., ¶¶ 56-83. Specifically, a person of ordinary skill in the art would have been motivated to use Sebastian's generic (i.e., default) compression encoder with Hsu's system in order to improve the compression ratio of the system by compressing data blocks with a default encoder when a data block is identified as a non-useful data type, one that is not associated with an encoder. *Id.*, ¶ 61. Sebastian provides an express teaching of this feature. Furthermore, implementing Hsu's system with Sebastian's teaching of a default encoder, as described above, would have required nothing more than using two known technologies for their intended purposes to produce predictable results. *Id.*, ¶ 57.

C. Ground 3: Claim 48 is Obvious in View of Franaszek and Hsu

1. Independent Claim 48 Is Obvious in View of Franaszek and Hsu

a. Franaszek Discloses the Preamble of Claim 48

Franaszek discloses "[a] computer implemented method" as recited by the preamble of claim 48. For example, Franaszek discloses a computer system that "dynamically compress[es] data blocks by using the best of a given set of methods." Franaszek at 3:25-28, 7:37-55. Franaszek's compression system is implemented using a processor and memory. *Id.* at 4:3-13.

b. Franaszek Discloses Limitation A of Claim 48

Franaszek discloses "receiving a data block," as recited by limitation A of claim 48. For example, Franaszek's method discloses that data blocks are first received from a first memory by a CPU. Franaszek at 4:3-13; Fig. 1. Fig. 2 also depicts data blocks 210 being received by data compression 220. *Id.* at 4:25-35, Fig. 2.

c. Franaszek Discloses Limitation B of Claim 48

Franaszek discloses "associating at least one encoder to each one of several data types," as recited by limitation B of claim 48. For example, Franaszek's system discloses that "if a data type (e.g. text, image, etc.) for a given uncompressed block B is available, in step 404 the Compression Method List (CML) is set to a list of compression methods that have been preselected for that data type." Franaszek at 5:49-53.

d. Franaszek Discloses Limitation C of Claim 48

Franaszek discloses "analyzing data within the data block to identify a first data type of the data within the data block," as recited by limitation C of claim 48. For example, Franaszek's method discloses analyzing a "data type" field of a data block to determine a data type of the block. Franaszek also discloses that "[i]n step 401, if a data type (e.g. text, image, etc.) for a given uncompressed block B is available, in step 404 the Compression Method List (CML) is set to a list of compression methods that have been preselected for that data type." *Id.* at 5:49-54. Further, as explained below with respect to limitation F, it would have been obvious to modify Franaszek to perform further analysis of data within the data block to determine data type information for the data block.

e. Franaszek Discloses Limitation D of Claim 48

Franaszek discloses "compressing, if said first data type is the same as one of said several data types, said data block with said at least one encoder associated with said one of said several data types that is the same as said first data type to provide a compressed data block," as recited by limitation D of claim 48. For example, Franaszek discloses compressing a sample of the data block with each compression encoder that is associated with the identified data type of a data block if the "data type" is identified. Franaszek at 5:49-53, 6:22-32, 6:43-50. The associated compression encoders are "the best of a given set of [compression] methods" for a given data type of the data block. *Id.* at 3:26. "Representative samples of each block of data are tested to select an appropriate one of the data compression mechanisms to apply to the block." Franaszek at 3:31-34. In other words, when the data type of the data block is identified, Franaszek's system compresses a data block with a compression encoder selected based on the data type. If the data type is not identified, a default list of compression encoders is used to compress the data block, regardless of the data block's actual data type. *Id.* at 5:49-54.

f. Franaszek Discloses Limitation E of Claim 48

Franaszek discloses "compressing, if said first data type is not the same as one of said several data types, said data block with a default encoder to provide said compressed data block," as recited by limitation E of claim 48. For example, Franaszek discloses using a default CML when a data type is not identified for a data block. Franaszek at 5:53-54. In other words, when the data type of the data block is not identified, a default list of compression encoders, which includes one or more default encoders, is used to compress the data block, regardless of the data block's actual data type. Accordingly, Franaszek teaches automatically applying an encoder from the default CML list to the data block when a designated alternative encoder (i.e., a data type specific encoder) is not identified from a data type. Hirschberg Decl., ¶¶ 45.

g. Implementing Limitation F of Claim 48 in View of Franaszek and Hsu Would Have Been Obvious

Modifying Franaszek in view of Hsu such that "wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of the data within the data block," as recited by limitation F of claim 48, would have been obvious to a person of ordinary skill in the art. Hirschberg Decl., ¶¶ 64-65.

Franaszek discloses that data blocks to be compressed may have a "data type" entry in a "type field" that may include content that identifies the type of data within the block. Franaszek at 6:1-2. When the content of the "data type" field identifies a data type, an associated predefined list of compression encoders is assigned to compress the data block. Franaszek, however, is silent on how to generate the contents of its "data type" field. In some cases, this field would already show a data type entry when the block is first received by Franaszek's system. But if no data type entry is available in the particular data block that is to be used with Franaszek's compression system, a person of ordinary skill in the art implementing Franaszek's compression system would look to others' teachings to determine how to identify the data type in order to generate the content for the "type field." In that case, if the data within the block does not include "a descriptor that is indicative of the data type of the data within the data block," a person of ordinary skill in the art would have had motivation to employ Hsu's teachings that include identifying a data type without relying "only on a descriptor that is indicative of the data type."

As explained above in Section VII.A.3.e, Hsu discloses analyzing the input data itself to identify a data classification based on three 512-byte segments of data at the beginning, middle, and end of the data block. Hsu also discloses performing statistical analysis of data blocks to determine redundancy metrics. A person of ordinary skill in the art would have been motivated to use Hsu's method of identifying data types (e.g., a data classification, a redundancy metric, or a type based on a combination of these types) so that Franaszek's compression system could be used to compress data blocks that do not already include the content for the "data type" field. Hirschberg Decl., ¶¶ 64-65. Combining the teachings of Franaszek and Hsu would have required nothing more than using each technology for its intended purpose to produce predictable and beneficial results. *Id.*

Accordingly, implementing limitation F of claim 48 would have been obvious in view of Franaszek and Hsu.

D. Ground 4: Claim 48 is Obvious in View of Franaszek and Chu

1. Summary of Chu

Chu discloses a data compression system that identifies the specific data type of an input data stream and then selects a compression encoder that provides for "an optimal compression ratio" for the data stream. Chu at Abstract, 8:28-43. Chu states that the data bytes of the data stream can be analyzed to determine the specific data type. *Id.* at 8:56-66 (comparing data bytes to each other and predetermined values to determine a specific data type for the input data); Hirschberg Decl., ¶¶ 54-55.

2. Independent Claim 48 Is Obvious in View of Franaszek and Chu

a. Franaszek Discloses the Preamble and Limitations A-E of Claim 48

As explained above in Sections VII.C.1.a-f, Franaszek discloses the Preamble and limitations A-E of claim 48.

b. Implementing Franaszek in View of Chu to Include Limitation F of Claim 48 Would Have Been Obvious

As explained above, Franaszek and Chu are both directed to compression systems that determine which compression encoder to use based on the type of data that is to be compressed. Chu's disclosure of analyzing the input data itself to determine a specific data type so that an optimized compression method can be selected (Chu at Abstract, 8:28-43, 8:56-66; Hirschberg Decl., ¶¶ 54-55) provides specific means to accomplish "wherein the analyzing of the data within the data block to identify one or more data types excludes analyzing based only on a descriptor that is indicative of the data type of the data within the data block," as recited by limitation F of claim 48. Specifically, Chu discloses that "the step of identifying the specific data type of the set of input data includes the step of determining whether each byte in said series of bytes represents a value greater than a predetermined value," and also that "the step of identifying the specific data type of the set of input data includes the step of determining whether selected bytes in said series of bytes are identical." (Chu at 8:58-61, 8:64-66.)

Chu's methods of identifying data types by analyzing the data when no descriptor is available provide further reasons why a person of ordinary skill in the art would have found any differences between Franaszek and claim 48 obvious. For the same reasons as those explained above with respect to Hsu in Section VII.C.1.g, claim 48 is obvious in view of Franaszek and Chu. Hirschberg Decl., ¶¶ 64-65. Specifically, a person of ordinary skill in the art implementing Franaszek's system to work with data streams that do not already include Franaszek's "data type" field (or do include a "data type" field but that includes insufficient information to determine an actual data type) would have been motivated to use other known techniques to determine the necessary data for the "data type" field. *Id.* Chu discloses one such well-known method that relies on the analysis of the data itself. *Id.*, ¶¶ 54-55. Furthermore, implementing Franaszek's system to include Chu's teachings, as described above, would have required nothing more than using two well-known technologies for their intended purposes to produce beneficial and predictable results. *Id.*, ¶¶ 64-65.

Accordingly, Franaszek in view of Chu renders obvious this claim for the same reasons as those explained in Section VII.C.1.g.

VIII. CONCLUSION

Based on the foregoing, it is clear that challenged claim 48 of the ’992 patent is obvious in view of the prior art references cited in this Petition. Accordingly, the Petitioner requests institution of an *inter partes* review to cancel those claims.

The USPTO is authorized to charge any required fees, including the fee as set forth in 37 C.F.R. § 42.15(a) and any excess claim fees, to Deposit Account No. **03-1952** referencing Docket No. **356670000015**.

Dated: June 22, 2017

Respectfully submitted,

By /Diek Van Nort/

Diek O. Van Nort
dvannort@mofo.com
Registration No.: 60,777
MORRISON & FOERSTER LLP
370 Seventeenth Street, Suite 4200
Denver, CO 80202
Tel: (650) 813-5696

Certification of Word Count (37 C.F.R. § 42.24)

I hereby certify that this Petition for *Inter Partes* Review has 10,529 words (as counted by the "Word Count" feature of the Microsoft Word™ word-processing system used to create this Petition), exclusive of "a table of contents, a table of authorities, mandatory notices under § 42.8, a certificate of service or word count, or appendix of exhibits or claim listing."

Dated: June 22, 2017

By /Diek Van Nort/
Diek O. Van Nort

Certificate of Service (37 C.F.R. § 42.6(e)(4))

I hereby certify that the attached Petition for *Inter Partes* Review and supporting materials were served as of the below date by courier, which is a means at least as fast and reliable as U.S. Express Mail, on the Patent Owner at the correspondence address indicated for U.S. Patent No. 7,378,992.

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1100 New York Avenue, N.W.
Washington DC 20005

Dated: June 22, 2017

By /Diek Van Nort/

Diek O. Van Nort
dvannort@mofo.com
Registration No.: 60,777
MORRISON & FOERSTER LLP
370 Seventeenth Street, Suite 4200
Denver, CO 80202
Tel: (650) 813-5696